Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Informationstechnische Regelung

Constrained Nonlinear Control for Safe Human-Robot Interaction

Melanie Kimmel

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzende:		Prof. Dr. Majid Zamani
Prüfende der Dissertation:		
	1.	Prof. Dr.-Ing. Sandra Hirche
	2.	Prof. Maria Prandini, Ph.D.

Die Dissertation wurde am 05.04.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 11.06.2018 angenommen.

In memoriam of Cobot4.

Preamble

The present thesis summarizes the research conducted during my four years at the Chair of Information-oriented Control (ITR) of the Technical University of Munich. Finishing this work would not have been possible without the people, who supported me and gave me advice in different ways and to whom I feel deeply indebted.

First and foremost, I would like to sincerely thank my doctoral advisor and head of the chair Prof. Sandra Hirche for allowing me the freedom to conduct research and follow my interests, for the stimulating discussions, the continued encouragement and for challenging me scientifically.

I am deeply grateful for the support and the inspiring discussions with Prof. Maria Prandini from the Dipartimento di Elettronica at the Politecnico di Milano, who insprired the union of scenario optimization with invariance control.

A special thank you goes out to the three Martins without whom I might have never tried my hand at research: to my uncle Martin Heigl for inspiring me to study electrical engineering and for frequently encouraging me to pursue a doctorate from the start, to Martin Lawitzky who introduced me to the chair and who supported my first scientific publications, and to Prof. Martin Buss, the head of the Chair of Automatic Control Engineering (LSR) at the Technical University of Munich, for finally convincing me to sign up for the challenge.

I would like to thank my colleagues for the productive and enjoyable work environment, especially Christoph Jähne for the cooperation producing augmented invariance control and my collaborators within the project con-humo Thomas Beckers, Hendrik Börner, and Jonas Umlauft, who were always willing to discuss pressing matters. I would further like to thank the administrative team of the chair for their kind and professional support in all nonscientific matters.

During my time at ITR, I had the honor of working with many excellent students. I would especially like to thank Manuel Rauscher, Jannick Pfort, and Jan Wöhlke for their contributions to control using control barrier functions and to the control of multi-agent systems.

Last but not least, I would like to express my gratitude to my parents for the guidance and for always encouraging me to pursue my goals and to my partner Lukas for the love, the support and for the enduring patience in all phases of this journey.

Acknowledgments

This work was supported by the EU Seventh Framework Programme FP7/2007-2013 within the ERC Starting Grant Control based on Human Models (con-humo), grant agreement no. 337654.

Abstract

Robotic systems sharing their workspace with humans must be able to interact and cooperate with human partners. As such a cooperation often involves physical contact or at least close interaction, it is essential to ensure that the robot poses no threat to the human. It does, however, not suffice to avoid potentially harmful collisions. In addition, the human needs to feel comfortable and relaxed during the cooperation. Otherwise the readiness to use assistive robots such as mobility aids, household assistants or rehabilitation devices will be low. A robot that meets both the physical and psychological demands of human-robot collaboration is considered safe. The desired safety defines several requirements a control scheme must fulfill to qualify for implementation on robots engaging in human-robot interaction. On the one hand, it needs to be capable of generating motions that satisfy any constraints on position or velocity imposed by interacting humans or the environment. These motions have to be generated under real-time conditions to enable a reaction to dynamic changes and they need to account for potential uncertainties in the constraint description. On the other hand, the control scheme should generate a behavior that feel natural to human partners and that may be extended to multiple interacting robots.

This thesis investigates some of the major challenges originating from close and physical human-robot interaction. It addresses the question how the satisfaction of dynamic constraints may be guaranteed under real-time conditions as well as how the least possible interference with the unconstrained task objective may be achieved and how the resulting dynamic behavior may be adjusted to feel comfortable for interacting humans. Furthermore, we aim at finding solutions to the challenges introduced by uncertain constraint parameters and by multiple interacting agents.

The main contribution of this thesis is a novel control framework combining a task-oriented control law with a constraint-enforcing control input, which is designed to satisfy dynamic and uncertain constraints under real-time conditions, while distributing the necessary actions between involved agents, if possible. For this purpose, we present two interchangeable feedback-linearization-based control approaches, namely invariance control and control barrier function-based control, that are implemented as an add-on to existing control loops. Both approaches formally guarantee constraint satisfaction while allowing to follow the task objective at a distance from the constraints. By introducing an augmentation to the system dynamics, the interaction behavior may be shaped to fulfill human expectations. Furthermore, robust and probabilistic satisfaction guarantees for constraints with uncertain parameters are presented and a novel prioritization scheme is established, which is the basis for distributing necessary actions for constraint satisfaction between multiple agents. Deriving the control action from a convex optimization allows finding a solution in real-time by applying efficient solvers. All presented approaches are designed to yield a similar structure, which enables arbitrary combinations allowing for more flexibility in the implementation.

Zusammenfassung

Robotersysteme, die sich ihren Arbeitsraum mit Menschen teilen, benötigen die Fähigkeit mit ihren menschlichen Partnern zu interagieren und kooperieren. Da solche Kooperationen oft in enger Zusammenarbeit und in direktem Kontakt stattfinden, muss unbedingt sichergestellt werden, dass der Roboter keine Bedrohung darstellt. Allerdings reicht es nicht aus nur potenziell gefährliche Kollisionen zu vermeiden. Der Mensch muss sich währenddessen auch wohl fühlen. Ansonsten wäre wohl kaum jemand bereit Roboter als Fortbewegungshilfen, Haushaltshilfen oder Trainer in der Rehabilitation zu akzeptieren. Ein Roboter, der sowohl die physischen als auch die psychologischen Anforderungen der Mensch-Roboter Interaktion erfüllt, wird als sicher bezeichnet. Die gewünschten Sicherheitsaspekte definieren mehrere Kriterien, die erfüllt sein müssen, damit sich ein Regelkonzept für die Mensch-Roboter Interaktion eignet. Einerseits muss es Bewegungen generieren, die, durch den Menschen oder durch die Umgebung bedingte, Positions- und Geschwindigkeitsbeschränkungen einhalten. Diese Bewegungen müssen in Echtzeit berechenbar sein, um eine Reaktion auf dynamische Änderungen der Beschränkungen zu erlauben und sollten eventuelle Unsicherheiten in der Beschränkungsdefinition berücksichtigen. Andererseits sollte sich das resultierende Verhalten für den interagierenden Menschen natürlich anfühlen und auf mehrere Roboter erweiterbar sein.

Diese Doktorarbeit untersucht einige der größten Herausforderungen, die durch die Zusammenarbeit von Menschen und Robotern entstehen. Wir widmen uns den Fragen, wie garantiert werden kann, dass dynamische Beschränkungen unter Echtzeitbedingungen eingehalten werden, wie das zielorientierte Sollverhalten so wenig wie möglich beeinflusst wird und wie die resultierende Dynamik an die Erwartungen des Interaktionspartners angepasst werden kann. Außerdem sollen Lösungsansätze für die Handhabung von Beschränkungen mit unsicheren Parametern und für die Regelung von multiplen Agenten entwickelt werden.

Der wichtigste Beitrag dieser Arbeit ist ein neuartiges Regelkonzept, das ein zielorientiertes mit einem beschränkungsorientierten Regelgesetz kombiniert, das die Einhaltung dynamischer und unsicherer Beschränkungen unter realen Bedingungen garantiert, während es, falls möglich, den nötigen Aufwand auf mehrere Agenten verteilt. Zu diesem Zweck stellen wir mit Invarianzregelung und Regelung mit Barrierefunktionen zwei Regelansätze vor, die auf Eingangs-Ausgangs-Linearisierung basieren und in bestehende Regelstrukturen eingefügt werden. Beide Methoden geben eine formale Garantie für die Einhaltung der Beschränkungen und erlauben das Verfolgen des Regelziels in gewisser Entfernung zu den Grenzen. Eine Erweiterung der Systemdynamik erlaubt das Verhalten während der Interaktion an die menschlichen Erwartungen anzupassen. Außerdem werden robuste und wahrscheinlichkeitstheoretische Garantien für Beschränkungen mit unsicheren Parametern entwickelt und ein neuartiges Priorisierungskonzept vorgestellt, worauf basierend die notwendigen Aktionen für die Einhaltung der Grenzen auf mehrere Agenten verteilt werden können. Da der Systemeingang über eine konvexe Optimierung bestimmt wird, ist es möglich effiziente Algorithmen zu benutzen, die das Problem in Echtzeit lösen. Alle vorgestellten Ansätze führen zu einer ähnlichen Struktur, wodurch beliebige Kombinationen möglich sind und das Konzept flexibel einsetzbar ist.

Contents

Notation

In this work, explicit function dependencies are omitted whenever clear from the context in order to improve readability.

Acronyms

I/O	input/output
PD	proportional-derivative
PID	proportional-integral-derivative
ISS	input-to-state stable
SISO	single-input single-output
MIMO	multi-input multi-output
MPC	model-predictive control
CBF	control barrier function
LP	linear program
QP	quadratic program
CARE	collision avoidance in real-time environments

Mathematical Conventions

Sets

$\mathcal{A}$, $\mathcal{H}$, $\mathcal{X}$	sets
$\partial\mathcal{A}$	boundary of the set $\mathcal{A}$
$\mathrm{Int}(\mathcal{A})$	interior of the set $\mathcal{A}$
$\neg\mathcal{A}$	inverse set of $\mathcal{A}$

$\mathcal{A} \cup \mathcal{B}$	union of sets $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} \cap \mathcal{B}$	intersection of sets $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{C}^k$	set of k times continuously differentiable functions $h : \mathbb{R}^n \to \mathbb{R}$
$\mathbb{N} = \{1, 2, 3, \ldots\}$	set of natural numbers
$\mathbb{R}$	set of real numbers
$\mathbb{R}^+$	set of positive real numbers

Scalars, Vectors, Matrices and Functions

a, h, x	scalars (small letters)
$\boldsymbol{a}$, $\boldsymbol{h}$, $\boldsymbol{x}$	column vectors (bold small letters)
$\boldsymbol{A}$, $\boldsymbol{H}$, $\boldsymbol{X}$	matrices (bold capital letters)
$\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$	identity matrix
$\mathbf{0}_n \in \mathbb{R}^{n \times n}$, $\mathbf{0}_{m \times n} \in \mathbb{R}^{m \times n}$	zero matrix
$\boldsymbol{a}^\mathsf{T} \in \mathbb{R}^{1 \times n}$, $\boldsymbol{A}^\mathsf{T} \in \mathbb{R}^{n \times m}$	transpose of $\boldsymbol{a} \in \mathbb{R}^n$ or $\boldsymbol{A} \in \mathbb{R}^{m \times n}$
$\boldsymbol{A}^{-1} \in \mathbb{R}^{n \times n}$	inverse of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}_n$
$\boldsymbol{A}^+ = \boldsymbol{A}^\mathsf{T}(\boldsymbol{A}\boldsymbol{A}^\mathsf{T})^{-1} \in \mathbb{R}^{n \times m}$	Moore-Penrose pseudo inverse of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, where $m < n$, $\operatorname{rank}(\boldsymbol{A}) = m$, $\boldsymbol{A}\boldsymbol{A}^+ = \boldsymbol{I}_m$ (right inverse)
$\boldsymbol{a}_1 \preccurlyeq \boldsymbol{a}_2$	element-wise inquality of $\boldsymbol{a}_1, \boldsymbol{a}_2 \in \mathbb{R}^n$
$\boldsymbol{A}_\mathcal{B} = [\boldsymbol{a}_{\boldsymbol{i}}{}^\mathsf{T}]_{i \in \mathcal{B}} = [\boldsymbol{a}_{\boldsymbol{i_1}} \ldots \boldsymbol{a}_{\boldsymbol{i}_{\lvert \boldsymbol{\mathcal{B}} \rvert}}]^\mathsf{T} \in \mathbb{R}^{\lvert \mathcal{B} \rvert \times n}$	concatenation of indexed vectors $\boldsymbol{a}_{\boldsymbol{i}} \in \mathbb{R}^n$
$\boldsymbol{b}_\mathcal{B} = [b_i]_{i \in \mathcal{B}} = [b_{i_1} \ldots b_{i_{\lvert \mathcal{B} \rvert}}]^\mathsf{T} \in \mathbb{R}^{\lvert \mathcal{B} \rvert}$	concatenation of indexed scalars $b_i \in \mathbb{R}^n$
$\lVert \boldsymbol{a} \rVert_2 = \sqrt{\boldsymbol{a}^\mathsf{T}\boldsymbol{a}}$	Euclidean vector norm (2-norm) of $\boldsymbol{a} \in \mathbb{R}^n$
$\operatorname{rank}(\boldsymbol{A})$	rank of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\operatorname{rank}(\boldsymbol{A}) \leq \min(m, n)$
$\ker(\boldsymbol{A})$	kernel of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{A}\boldsymbol{x} = \mathbf{0}_{m \times 1}$ if $\boldsymbol{x} \in \ker(\boldsymbol{A})$
$\operatorname{diag}(\boldsymbol{a}) = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & a_n \end{bmatrix}$	diagonal matrix with entries defined by the vector $\boldsymbol{a} = [a_1 \ldots a_n]^\mathsf{T} \in \mathbb{R}^n$
$\ln(a)$	natural logarithm of a positive scalar $a \in \mathbb{R}^+$
$\operatorname{sign}(a) = \begin{cases} 1 & a > 0 \\ 0 & a = 0 \\ -1 & a < 0 \end{cases}$	signum of a

$\lceil a \rceil$	smallest integer greater than or equal to $a \in \mathbb{R}$

Derivatives

$\frac{\mathrm{d}f(x)}{\mathrm{d}x}$	total derivative
$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \left[\frac{\partial f(\boldsymbol{x})}{\partial x_n} \; \dots \; \frac{\partial f(\boldsymbol{x})}{\partial x_n}\right]$	partial derivative
$\dot{x} = \frac{\mathrm{d}x}{\mathrm{d}t}$, $\ddot{x} = \frac{\mathrm{d}^2 x}{\mathrm{d}t^2}$	low order time derivatives
$x^{(r)} = \frac{\mathrm{d}^r x}{\mathrm{d}t^r}$	time derivative of order r
$\mathscr{L}_{\boldsymbol{x}\boldsymbol{f}} h(\boldsymbol{x}) = \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}$	first order Lie derivative of $h(\boldsymbol{x})$ w.r.t. $\boldsymbol{f} \in \mathbb{R}^n$ with $\boldsymbol{x} \in \mathbb{R}^n$ (directional derivative)
$\mathscr{L}^r_{\boldsymbol{x}\boldsymbol{f}} h(\boldsymbol{x}) = \frac{\partial \mathscr{L}^{r-1}_{\boldsymbol{x}\boldsymbol{f}} h(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}$	r-th order Lie derivative of $h(\boldsymbol{x})$ w.r.t. $\boldsymbol{f}$
$\mathscr{L}_{\boldsymbol{x}\boldsymbol{G}} h(\boldsymbol{x}) = \left[\mathscr{L}_{\boldsymbol{x}\boldsymbol{g}_1} \; \dots \; \mathscr{L}_{\boldsymbol{x}\boldsymbol{g}_m}\right] h(\boldsymbol{x})$	first order Lie derivative of $h(\boldsymbol{x})$ w.r.t. the matrix $\boldsymbol{G} = [\boldsymbol{g_1} \dots \boldsymbol{g_m}] \in \mathbb{R}^{n \times m}$ with $\boldsymbol{x} \in \mathbb{R}^n$

Probabilities

$\mathscr{P}(\mathcal{A})$	probability of $\mathcal{A}$
$\mathscr{P}(\neg\mathcal{A}) = 1 - \mathscr{P}(\mathcal{A})$	probability of the inverse set of $\mathcal{A}$
$\mathscr{P}(\mathcal{A}\vert\mathcal{B}) = \dfrac{\mathscr{P}(\mathcal{A} \cap \mathcal{B})}{\mathscr{P}(\mathcal{B})}$	conditional probability of $\mathcal{A}$ given $\mathcal{B}$

Subscripts

$(\cdot)_0$	initial value
$(\cdot)_f$	final value
$(\cdot)_{\mathrm{des}}$	desired value considering constraints
$(\cdot)_{\mathrm{m}}$	measured value
$(\cdot)_{\mathrm{ref}}$	reference value without consideration of constraints
$(\cdot)_{\mathrm{a}}$	associated with augmented control
$(\cdot)_{\mathrm{B}}$	associated with control barrier functions
$(\cdot)_{\mathrm{c}}$	associated with invariance control
$(\cdot)_{\mathrm{no}}$	associated with nominal control

Variables

$\boldsymbol{a}_i^\intercal \in \mathbb{R}^m$	input/output-linearizing input transformation, multiplicative component
$\boldsymbol{A}_\mathrm{a} \in \mathbb{R}^{m \times m}$	input/output-linearizing input transformation of the augmentation, multiplicative component
$\mathcal{A}_i$	set of active agents for constraint i
b_i	input/output linearizing input transformation, additive component
$\boldsymbol{b}_\mathrm{a} \in \mathbb{R}^m$	input/output-linearizing input transformation of the augmentation, additive component
$B_i : \mathbb{R}^n \times \mathbb{R}^{(r_\mathrm{max}+1)n_\eta} \to \mathbb{R}^+$	control barrier function for constraint i
$\mathcal{B}$	set of constraints
$\mathcal{B}_\mathrm{act}$	set of active constraints
$\mathcal{B}_\mathrm{pr}$	set of chance constraints
$\mathcal{B}_\mathrm{pr,act}$	set of active chance constraints
$\mathcal{B}_\mathrm{sc}$	set of scenario constraints
$\mathcal{B}_\mathrm{sc,act}$	set of active scenario constraints
$\mathcal{B}_\mathrm{rob}$	set of robust constraints
$\mathcal{B}_\mathrm{rob,act}$	set of active robust constraints
$\tilde{\mathcal{B}}$	set of constraints included in the optimization for control actions
$c_j \in \mathbb{R}^+$	priority of agent j
$c_{j,i} \in \mathbb{R}^+$	criterion coefficient for priority assignment of agent j and criterion i
$c_\mathrm{max} \in \mathbb{R}^+$	maximum priority value
$\boldsymbol{C_q}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^{n_q}$	generalized Coriolis and centripetal forces
d_i	right hand side of the optimization condition derived from constraint i
$\mathcal{D}_\eta \subseteq \mathbb{R}^{n_\eta}$	set of all possible values of the parameters $\boldsymbol{\eta}$
$\mathcal{D} \subseteq \mathbb{R}^{n_\eta(r_\mathrm{max}+1)}$	set of all possible instances of the uncertainties in the parameter state $\boldsymbol{x}_\eta$
$\boldsymbol{e}_\mathrm{u} \in \mathbb{R}^m$	input error

$\boldsymbol{e}_{\boldsymbol{x}} \in \mathbb{R}^n$	tracking error of state $\boldsymbol{x}$
$\mathcal{E}$	emergency community
$\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$	system vector field, control affine form
$\boldsymbol{f}_{\boldsymbol{R}} : \mathbb{R}^{2n_q} \to \mathbb{R}^{2n_q}$	robot vector field, control affine form
$\boldsymbol{f}_{\boldsymbol{p}} : \mathbb{R}^{n_q} \to \mathbb{R}^{n_p}$	forward kinematics, robotic system
$\boldsymbol{f}_{\mathrm{ext}} \in \mathbb{R}^{n_p}$	external forces acting on a robotic system
$\boldsymbol{g}_{\boldsymbol{q}} : \mathbb{R}^{n_q} \to \mathbb{R}^{n_q}$	generalized gravitational torques
$\boldsymbol{g}_{\boldsymbol{i}} : \mathbb{R}^n \to \mathbb{R}^n$	columns of system matrix field $\boldsymbol{G}$
$\boldsymbol{G} : \mathbb{R}^n \to \mathbb{R}^{n \times m}$	system matrix field, control affine form
$\boldsymbol{G}_{\boldsymbol{R}} : \mathbb{R}^{2n_q} \to \mathbb{R}^{2n_q \times n_q}$	robot matrix field, control affine form
$\mathcal{G}$	invariant set
$\boldsymbol{h}_{\mathrm{c}} : \mathbb{R}^n \times \mathbb{R}^{n_\eta} \to \mathbb{R}^l$	artificial constraint output function
$\boldsymbol{h}_{\mathrm{B}} : \mathbb{R}^n \times \mathbb{R}^{n_\eta} \to \mathbb{R}^l$	artificial constraint output function for CBF-based control, $\boldsymbol{h}_{\mathrm{B}} = -\boldsymbol{h}_{\mathrm{c}}$
$\boldsymbol{h}_{\mathrm{nat}} : \mathbb{R}^n \to \mathbb{R}^q$	natural system output function
$\mathcal{H}$	admissible set
$\boldsymbol{J} : \mathbb{R}^{n_q} \Rightarrow \mathbb{R}^{n_p \times n_q}$	Jacobian matrix
$\mathcal{J}_i$	set of inactive agents for constraint i
l	number of constraints
m	dimension of system input
m_i	input dimension of agent i
$\boldsymbol{M}_{\boldsymbol{q}} : \mathbb{R}^{n_q} \to \mathbb{R}^{n_q \times n_q}$	generalized mass matrix
$\mathcal{M}$	set of admissible control inputs for invariance control
n	dimension of the system state
n_{ag}	number of agents
n_{pc}	number of priority criteria
n_i	state dimension of agent i
n_η	dimension of the uncertain constraint parameters
n_{nat}	dimension of the system output

$n_{\mathrm{nat},i}$	output dimension of agent i
n_q	dimension of the generalized coordinates of the robotic system
n_p	dimension of the task space coordinates
$N_i \in \mathbb{N}$	number or scenarios required for scenario-based satisfaction of constraint i
$\mathcal{N}_{\mathrm{ag}}$	set of controlled agents
$\boldsymbol{p} \in \mathbb{R}^{n_p}$	task space coordinates of the robotic system
$\mathcal{P}_j$	j-th agent priority community
$\mathcal{P}_{\mathrm{min},i}$	minimal agent priority community of constraint i
$\boldsymbol{q} \in \mathbb{R}^{n_q}$	generalized coordinates of the robotic system
r	relative degree in input/output (I/O)-linearization
r_{max}	maximum relative degree for multiple outputs
T_A	sampling time
$\boldsymbol{u} \in \mathbb{R}^m$	system input
$\boldsymbol{u}_{\mathrm{no}} \in \mathbb{R}^m$	nominal control input
$\boldsymbol{u}_{\mathrm{c}} \in \mathbb{R}^m$	corrective control input
$\boldsymbol{u}_{\mathrm{ext}} \in \mathbb{R}^m$	external measurable input disturbance
$\boldsymbol{u}_{\mathrm{e}} \in \mathbb{R}^m$	external unmeasurable input disturbance
v	degree of augmentation
$w_{i,j} : (\mathbb{R}^+)^{n_{\mathrm{ag}}} \to \mathbb{R}^+$	weight for distributing the evasive control action for constraint i to agent j
$\omega_{j,i} \in \mathbb{R}^+$	weight corresponding to the criterion coefficient $c_{j,i}$
$\boldsymbol{x} \in \mathbb{R}^n$	system state
$\boldsymbol{x}_{\boldsymbol{R}} = [\boldsymbol{q}^\mathsf{T},\ \dot{\boldsymbol{q}}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{2n_q}$	state of the robotic system
$\boldsymbol{x}_\eta \in \mathbb{R}^{(r_{\mathrm{max}}+1)n_\eta}$	state of the constraint dynamics
$\boldsymbol{y}_{\mathrm{c}} \in \mathbb{R}^l$	artificial constraint output
$\boldsymbol{y}_{\mathrm{B}} \in \mathbb{R}^l$	barrier constraint output, negative of $\boldsymbol{y}_{\mathrm{c}}$
$\boldsymbol{y}_{\mathrm{nat}} \in \mathbb{R}^q$	natural system output
$\boldsymbol{\eta} \in \mathbb{R}^{n_\eta}$	constraint parameter vector

$\gamma \in \mathbb{R}^-$	control parameter for invariance control
$\mu \in \mathbb{R}^+$	control parameter for CBF-based control
$\Phi_i : \mathbb{R}^n \times \mathbb{R}^{n_\eta} \times \mathbb{R}^- \to \mathbb{R}$	invariance function corresponding to constraint i
$\Phi_{\text{th}} \in \mathbb{R}^-$	threshold determining nearly active constraints
$\boldsymbol{\tau} \in \mathbb{R}^{n_q}$	input torque, robotic system
$\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^{n_q}$	generalized torques generated by external forces acting on the robotic system
$\zeta_a,\ \zeta_b \in \mathbb{R}^+$	Control Barrier Function parameters for relative degree two constraints

1 Introduction

Recent advances in robotics research and development have opened up a variety of application fields. Increased computational capabilities, the development of dexterous grippers, mobile platforms and strong manipulators enable the execution of complex tasks, the employment in highly dynamic environments and allow intelligent robots to enter human-centered environments.

While the use of robotic manipulators is already common in industrial manufacturing, these systems are still strictly separated from the humans' workspace for safety reasons. There is, however, an effort to join the workspaces of humans and robots and to exploit the advantages of physical human-robot interaction, which aims at employing strong manipulators to take over the main load and strenuous tasks while the human partner adopts a guiding role [MLK+12]. This close interaction of humans and robots operating potentially dangerous tools create a need to identify safety hazards [VB13] and to specify safety requirements [ISO14; ISO16] for such robotic systems.

An even closer interaction is found in the health care sector, where there is a growing interest in assistive robotic systems. While tele-operated manipulators are instrumental in minimal invasive surgery [MLO+03], wearable devices such as exoskeletons are designed to help patients with motor disorders [Pon10], intelligent systems are used in rehabilitation for training the patients through kinesthetic interaction [RPH+05] and mobile robots serve as mobility assistant for elderly and physically impaired people [GP14].

Robots have entered our homes to clean and efforts are made to introduce robotic assistants helping with everyday chores [Kha99; JG12]. These robotic helpers may even be trained to assist in the transport and manipulation of bulky object such as, for example, furniture [SKK03; MLM+11].

Meanwhile, even cars show similarities to intelligent robots as they park and by now even drive autonomously. The control design of such vehicles faces similar challenges to human-robot interaction as it requires reliable path planning, an estimation of traffic [ASB09] and especially of the behavior and intention of human road users [SCC+13].

1.1 Challenges in Close Human-Robot Interaction

The presented exemplary applications involve a close or even physical interaction of humans and robots. The success of physical human-robot interaction, which is characterized by a physical coupling of human and robot either directly or indirectly through an object, depends on compliance, performance, dependability under real-time conditions and safety of the developed control scheme [DSD+08]. Safe interaction is characterized by physical and psychological factors. It does not suffice to ensure the integrity of the human partner but the interaction also needs to be comfortable and stress-free [LFS17]. Therefore, it is desirable for the robot to carry out predictable motions which allow human co-workers to assess the consequences of actions [SR11].

Challenge 1. *Is it possible to design the control input such that the resulting interaction feels natural to human partners?*

Nevertheless, guaranteeing the integrity of any humans in the vicinity, the environment and the robotic system itself is essential. This imposes restrictions on the robot as it introduces bounds on velocities, interaction forces and configurations as the avoidance of potentially harmful collisions has to be avoided [ISO14; ISO16].

Challenge 2. *How may the satisfaction of arbitrarily high numbers of constraints be guaranteed in real-time applications even in the presence of interaction forces?*

Naturally, there is generally the potential for undesired collisions due to modeling or measurement errors, which may be resolved by designing the compliance such that forces occurring due to undesired collisions are minimized [HAD+08]. Ideally, however, the goal is to avoid collisions altogether by designing a control approach that anticipates imminent collisions and avoids them by constraining the robotic system to a virtual safe region using constraints. If it is possible to combine such a method with active compliance control [Hog85; AOF+03] or PD tracking control [MLS94] in a single framework, this allows to design the behavior to facilitate successful human-robot interaction. As there are applications which require the use of learning-based approaches such as reinforcement learning [WSB90; BBS06], the control framework should be able to handle implicit goal descriptions.

Challenge 3. *How may little interference with arbitrary goals and tasks be achieved in the presence of constraints?*

As humans move and everyday environments may change dynamically, the imposed constraints vary over time. Being able to satisfy the imposed dynamic constraints creates the need for accurate methods to estimate human motion [ĆJP+16], for example by exploiting knowledge about the underlying dynamics such as the minimum jerk property of human motion [FH85]. Such a motion model is also helpful in creating a behavior that improves the perceived safety [LFS17]. However, estimations are generally afflicted with errors, which means that a constraint enforcing control scheme has to handle the resulting uncertainties, for example by using knowledge about the related distribution or by using samples.

Challenge 4. *How may uncertainties in measurements and estimations of constraint definitions be included in the constraint description?*

If these uncertainties are widespread, the admissible space available may be highly restricted. This is also true for narrow environments or applications including multiple closely interacting robots and/or humans. In this case, each agent should be able to use as much of the unconstrained space as possible for achieving the task objective. Furthermore, if the robots are able to share the effort necessary for constraint satisfaction, this has two effects:

individual agents experience less limitations and prioritization of tasks is enabled. As, for example, the rescue of an injured person is probably more important than scouting the environment and reaching the docking station for low energy takes priority over a cleaning task, the possibility of task prioritization is useful for applications including multiple agents carrying out tasks of different importance.

Challenge 5. *Is there a way to share the effort for constraint satisfaction between multiple agents?*

Addressing these challenges is key to a successful interaction. The present thesis studies the design of control schemes for safe human-robot interaction, concentrating on the presented challenges.

1.2 Control for Constraint Satisfaction

There is a variety of control approaches to choose from to achieve the satisfaction of constraints required for safe human-robot interaction.

Potential fields, for example, are applied in collision-free navigation and obstacle avoidance [Kha85; RK92]. Using repulsive potentials allows to artificially limit the workspace of a robotic system to a safe region [DKW+14]. The functions may even be designed to handle geometric uncertainties [NRL94], which addresses Challenge 4. As the control input is determined analytically, the use in fast real-time systems is straightforward. However, it is not straightforward to explicitly take the system dynamics into account. As these are not negligible especially for high inertias or high accelerations, potential fields are not able to provide the satisfaction guarantee requested in Challenge 2.

In the area of virtual reality, haptic feedback is provided by virtual wall rendering, where the stiffness switches from low to high values at a wall [GC96]. Similarly, virtual fixtures, which originate from tele-operation [Ros93], provide an overlaid haptic feedback, for example for improved performance in robot-aided surgery but also for guiding robots [MLO+03] and robot-assisted manipulation [AMO07]. The approach is designed to reject any motion into forbidden regions and aims at guidance for improved performance. However, as both virtual walls and virtual fixtures do not include the system dynamics, they are not able to address Challenge 2 and do not guarantee the constraint satisfaction required for safe human-robot interaction.

The dynamic window approach [FBT97; LAW+11] is used to generate safe platform trajectories. It restricts the available velocity inputs to safe values by predicting the system trajectory thus achieving obstacle avoidance. The approach is designed for velocity inputs and hence lacks flexibility as it may not be applied to the torque controlled manipulators that are required to generate the desired compliance in physical human-robot interaction.

Shared control provides a trade-off between a human-generated input and a stabilizing input [GG04; KBS+06; SCC+13] or an input designed for constraint satisfaction [JA14]. The result may however be suboptimal in a way that the system does not necessarily use the safe space to all extent, which is disadvantageous in narrow environments.

The probably most well-known approach, model-predictive control (MPC) [MRR+00], originates from constrained control and allows for input, output and state constraints. A control input is derived by joining task objectives and constraints in a constrained optimization problem. MPC may be combined with barrier certificates for the strict satisfaction of constraints [WH04]. Furthermore, there are robust MPC [BM99] and scenario MPC [SM15; GZM+16], which enable handling uncertain constraints. Note that while robust approaches guarantee constraint satisfaction, scenario-based MPC provides a satisfaction probability [CC05]. However, even though there exist implementations designed for fast real-time applications [ITO+11], in general, the cost of solving the optimization problem may prevent finding a solution in real-time especially for high-dimensional and nonlinear systems subject to a high number of constraints.

Set-based control approaches with examples including reference governors [GK02], command governors [AM99], invariant and reachable sets [Bla99] rely on deriving a set solely containing admissible actions or states. If the control input keeps the state within such a set, constraint satisfaction is guaranteed. Deriving robust sets [FAC10] allows accounting for uncertainties in the constraint description. Furthermore, reachable sets are also demonstrated to be suitable for safe learning [AFG+14]. Even though there exist the so-called fast reference governors [VKS07], the computation of these sets is costly especially for complex system dynamics, which renders them unsuitable for applications with high sampling times.

Barrier certificates [PR07] are another approach that allows including constraints in control loops. They are used as control barrier functions (CBFs) in combination with feedback-linearization [AGT14; RKH16; Xu18] similar to invariance control [MBS00; WB04; SWB08]. Both methods combine characteristics from optimization-based and set-based approaches. They apply feedback linearization to simplify the derivation of an invariant safe set of states and control inputs. They choose a suitable control input by solving a convex optimization problem, which allows for the use of efficient solvers and enables the application in real-time addressing Challenge 2. However, only preliminary results are available for invariance control in the presence of uncertainties [BG10; MB99], an issue that has not been addressed to date for CBFs to the best of our knowledge. Furthermore, an approach for distributing the control action in multi-agent systems using the linearization-based approaches is missing to date.

Addressing Challenge 5, there exist some approaches especially designed to satisfy constraints in multi-agent systems. In [Ros96; FSR01; FR03], collision avoidance in real-time environments (CARE) is introduced for multi-agent systems. CARE is, however, not able to provide guarantees for constraint satisfaction as the approach only achieves an approximate reaction especially for complex shapes or structures. Another approach handles multi-agent collision avoidance by stopping the lower priority agents [CYZ+07], which may result in agents blocking the path. Alternatively, task allocation models [LJX15; MBH+15] distribute evasive actions and schedule tasks on the basis of existing priorities, but do not allow re-scheduling in case of changing priorities.

1.3 Main Contributions and Outline

This thesis aims at developing a control framework, which efficiently determines a control input for constraint satisfaction as well as task execution in real time, which is safe in human-robot interaction and applicable in case of uncertain constraints and for multiple agents, hence addressing the challenges discussed in Sec. 1.1.

Chapter 2 introduces the general setting consisting of system, task-oriented control and constraints. Based on the system characteristics, Chapter 3 introduces two control approaches for guaranteed constraint satisfaction suitable for close and physical human-robot interaction. Chapter 4 examines the effect of uncertainties on the constraints and provides extensions to the previously introduced approaches. Chapter 5 develops a strategy allowing the introduced control methods to share the control effort between multiple agents. All derivations are supported by experiments and examples. Conclusions and potential future directions are presented in Chapter 6 and the proofs supporting the formal results are provided in the appendix.

At the beginning of each chapter, a brief review of the relevant related work and the open problems is provided. The major contributions of this thesis addressing the presented challenges are outlined in the remainder of this section.

Chapter 3: Control Methods for Guaranteed Constraint Satisfaction

This chapter addresses Challenge 1–3. We show that the input/output (I/O)-linearization-based control approaches, invariance control and CBF-based control, may be added to arbitrary task-oriented control laws while guaranteeing the satisfaction of dynamic constraints. We address the implementation in sampled real-time systems and present augmented invariance control as an approach allowing for more adaptability in the dynamics of the controlled system, thus being able to design a behavior suitable for close and physical human-robot interaction. The results presented in this chapter have partially been published in [KH14; KH15; KH16; RKH16; KJH16; KH17].

Chapter 4: Satisfaction of Constraints with Uncertain Parameters

This chapter addresses Challenge 4. Extending the control approaches presented in Chapter 3 to allow for the satisfaction of constraints with uncertain parameters, we present control for robust, probabilistic and scenario-based constraint satisfaction. Preserving the characteristics of the original control methods, the control types guarantee robust or probabilistic constraint satisfaction and are interchangeable to allow for a combination of the different constraint types. The results presented in this chapter have been partially submitted for publication in [KPHed].

Chapter 5: Sharing Control in Multi-Agent Systems

This chapter addresses Challenge 5. Extending the control approaches presented in Chapter 3, we introduce shared constraint enforcement. Using weights, which are determined from agent priorities, the control effort is divided between the agents to allow for improved task execution and a distributed implementation. We further introduce a two-stage prioritization scheme that avoids blocking situations and distributes priorities according to a variety of criteria. The results presented in this chapter have been partially submitted for publication in [KPW+ed].

2

Problem Setting

The choice of control scheme for achieving safe and task-oriented control of a dynamical system depends on various factors: number and shape of constraints, desired behavior in the admissible space, behavior in case of constraint violations, size and characteristics of the system such as linear or nonlinear, control-affine, discrete, sampled or continuous. These factors strongly influence whether a certain constraint-enforcing control approach may be considered or not and thus need to be examined carefully. In addition, to allow for wide applicability, the constraint enforcement should be compatible with any type of desired behavior. This is the case if the constraint-enforcing control scheme (corrective control) may be added to any existing control loop of system and task-oriented control (nominal control) as depicted in Fig. 2.1. This structure allows nominal control to be designed according to task specifications such as stability, tracking behavior and compliance to forces without any considerations of the constraints. The corrective control part should then combine the nominal control signal with knowledge of constraints and system and adapt the control input, if necessary, to ensure adherence to the constraints.

This chapter establishes the prerequisites for developing such control schemes. First, the requirements for general stand-alone systems based on [WB04; AGT14] and multi agent systems according to [KPW+ed] are introduced. The subsequent definition of constraints with static [MBS00] and dynamic [KH15; KH17] parameters includes both constraints with exactly known and constraints with uncertain parameters [KPHed]. Finally, the problem setting is completed by discussing the requirements that a task-oriented control scheme has to meet and by introducing well-known control laws for robotic applications.

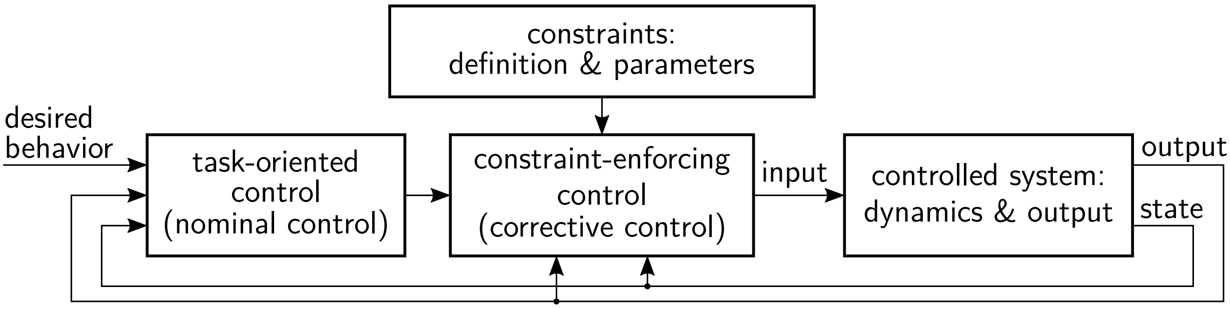


Figure 2.1.: Desired control structure with the constraint-enforcing controller added to the control loop in between the system and the task-oriented controller.

2.1 System

Let a ***stand-alone control affine nonlinear system*** be given in a time-continuous control affine form

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{y}_{\text{nat}} &= \boldsymbol{h}_{\text{nat}}(\boldsymbol{x}) \\ \boldsymbol{x}(t_0) &= \boldsymbol{x}_0 \end{aligned} \tag{2.1}$$

with state $\boldsymbol{x} \in \mathbb{R}^n$, input $\boldsymbol{u} \in \mathbb{R}^m$ and output $\boldsymbol{y}_{\text{nat}} \in \mathbb{R}^{n_{\text{nat}}}$. The dynamics are determined by the locally Lipschitz vector fields $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\boldsymbol{g}_i : \mathbb{R}^n \to \mathbb{R}^n$ with $i \in \{1, \dots, m\}$, which make up the matrix vector field $\boldsymbol{G} = [\boldsymbol{g}_1 \cdots \boldsymbol{g}_m] : \mathbb{R}^n \to \mathbb{R}^{n \times m}$.

In a setup of ***multiple control affine nonlinear systems***, the dynamics of each agent $i \in \mathcal{N}_{\text{ag}}$, where

$$\mathcal{N}_{\text{ag}} = \{i \in \mathbb{N} \mid 1 \leq i \leq n_{\text{ag}}\} \tag{2.2}$$

is the set of agents, are given by

$$\begin{aligned} \dot{\boldsymbol{x}}_i &= \boldsymbol{f}_i(\boldsymbol{x}_i) + \boldsymbol{G}_i(\boldsymbol{x}_i)\boldsymbol{u}_i \\ \boldsymbol{y}_{\text{nat},i} &= \boldsymbol{h}_{\text{nat},i}(\boldsymbol{x}_i) \\ \boldsymbol{x}_i(t_0) &= \boldsymbol{x}_{i,0} \end{aligned} \tag{2.3}$$

with state $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$, input $\boldsymbol{u} \in \mathbb{R}^{m_i}$ and output $\boldsymbol{y}_{\text{nat},i} \in \mathbb{R}^{n_{\text{nat},i}}$. Hence, the concatenated state $\boldsymbol{x} = [\boldsymbol{x}_1^\intercal \dots \boldsymbol{x}_{n_{\text{ag}}}^\intercal]^\intercal \in \mathbb{R}^n$, input $\boldsymbol{u} = [\boldsymbol{u}_1^\intercal \dots \boldsymbol{u}_{n_{\text{ag}}}^\intercal]^\intercal \in \mathbb{R}^m$ and natural system output $\boldsymbol{y}_{\text{nat}} = [\boldsymbol{y}_{\text{nat},1}^\intercal \cdots \boldsymbol{y}_{\text{nat},n_{\text{ag}}}^\intercal]^\intercal \in \mathbb{R}^{n_{\text{nat}}}$ of the multi-agent system have the dimensions $n = \sum_{i \in \mathcal{N}_{\text{ag}}} n_i$, $m = \sum_{i \in \mathcal{N}_{\text{ag}}} m_i$ and $n_{\text{nat}} = \sum_{i \in \mathcal{N}_{\text{ag}}} n_{\text{nat},i}$. Locally Lipschitz vector fields $\boldsymbol{f}_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$ and $\boldsymbol{g}_{i,j} : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$, $j \in \{1, \dots, m_i\}$ determine the dynamics and the matrix vector field $\boldsymbol{G}_i = [\boldsymbol{g}_{i,1} \cdots \boldsymbol{g}_{i,m_i}] : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i \times m_i}$.

The control approaches proposed in this work rely on input/output (I/O)-linearization of the system using an artificial output representing the constraint. In order to allow for the linearization, the following assumption concerning the system properties is necessary.

Assumption 2.1. *The (matrix) vector fields $\boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{G}(\boldsymbol{x})$ for stand-alone systems* (2.1) *and $\boldsymbol{f}_i(\boldsymbol{x}_i)$, $\boldsymbol{G}_i(\boldsymbol{x}_i)$ for each agent i in a multi-agent system* (2.3) *are sufficiently smooth to allow for I/O-linearization.*

As I/O-linearization involves partial derivation of the vector fields, sufficiently smooth means that these derivatives are defined and continuous [Kha96, p.508]. The system properties are illustrated in the following example.

Example 2.1 (Generalized dynamics of robotic systems)**.** As an example, we consider a robotic system with the dynamics

$$\boldsymbol{M_q}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C_q}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_q(\boldsymbol{q}) = \boldsymbol{\tau} \tag{2.4}$$

and the generalized coordinates $\boldsymbol{q} \in \mathbb{R}^{n_q}$. The dynamic behavior is influenced by the mass matrix $\boldsymbol{M_q}(\boldsymbol{q}) \in \mathbb{R}^{n_q \times n_q}$, the Coriolis and centripetal forces $\boldsymbol{C_q}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^{n_q}$, the gravitational torques $\boldsymbol{g}_q(\boldsymbol{q}) \in \mathbb{R}^{n_q}$ and the torque input $\boldsymbol{\tau} \in \mathbb{R}^{n_q}$. By defining $\boldsymbol{x}_R = [\boldsymbol{q}^\intercal,\ \dot{\boldsymbol{q}}^\intercal]^\intercal \in \mathbb{R}^{2n_q}$ as the state, the robot dynamics (2.4) are transformed into the control affine form

$$\dot{\boldsymbol{x}}_R = \underbrace{\begin{bmatrix} \dot{\boldsymbol{q}} \\ -\boldsymbol{M_q}^{-1}(\boldsymbol{q})(\boldsymbol{C_q}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_q(\boldsymbol{q})) \end{bmatrix}}_{\boldsymbol{f}_R(\boldsymbol{x}_R)} + \underbrace{\begin{bmatrix} 0 \\ \boldsymbol{M_q}^{-1}(\boldsymbol{q}) \end{bmatrix}}_{\boldsymbol{G}_R(\boldsymbol{x}_R)} \boldsymbol{\tau} \tag{2.5}$$

This system may be I/O-linearized with respect to any output function depending on $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ or the entire state $\boldsymbol{x}_R$ as shown in [KH17] and thus fulfills Assumption 2.1.

2.2 Constraints

Wide applicability of a constraint enforcing control scheme is given if it allows for different types of constraints: convex and non-convex, static and dynamic, exactly known and with uncertain parameters. Each constraint on the system state $\boldsymbol{x}$ is defined as artificial output function which fulfills the following conditions.

Definition 2.1. A ***constraint (output) function*** y_c is an analytic function depending on the state $\boldsymbol{x} \in \mathbb{R}^n$ and parameters $\boldsymbol{\eta} \in \mathcal{D}_\eta \subseteq \mathbb{R}^{n_\eta}$ and fulfills

$$y_\mathrm{c} = h_\mathrm{c}(\boldsymbol{x}, \boldsymbol{\eta}) \begin{cases} < 0 & \text{if } \boldsymbol{x} \text{ adheres to the constraint for } \boldsymbol{\eta}, \\ = 0 & \text{if } \boldsymbol{x} \text{ lies on the constraint for } \boldsymbol{\eta}, \\ > 0 & \text{if } \boldsymbol{x} \text{ violates the constraint for } \boldsymbol{\eta}. \end{cases}$$

Each such function represents a single ***state constraint***.

Note that even though Def. 2.1 addresses state constraints, the concept may straightforwardly extended to output constraints.

Remark 2.1 (Output constraints)**.** A constraint on the output $\boldsymbol{y}_\mathrm{nat}$ is represented by

$$y_\mathrm{c} = h_\mathrm{c}(\boldsymbol{y}_\mathrm{nat}, \boldsymbol{\eta}) \begin{cases} < 0 & \text{if } \boldsymbol{y}_\mathrm{nat} \text{ adheres to the constraint for } \boldsymbol{\eta}, \\ = 0 & \text{if } \boldsymbol{y}_\mathrm{nat} \text{ lies on the constraint for } \boldsymbol{\eta}, \\ > 0 & \text{if } \boldsymbol{y}_\mathrm{nat} \text{ violates the constraint for } \boldsymbol{\eta}. \end{cases}$$

Using the dependence of the output $\boldsymbol{y}_\mathrm{nat} = \boldsymbol{h}_\mathrm{nat}(\boldsymbol{x})$ on the state $\boldsymbol{x}$, the constraint function may be expressed as a state constraint similar to Def. 2.1. Hence, output constraints are merely a special case of state constraints. This type of constraints is therefore not addressed explicitly in the remainder of this work as the introduced concepts for state constraints hold, by extension, also for output constraints.

Two types of state constraints, namely linear and spherical constraints, are often used in control applications. Upper or lower limits are best described by linear functions, whereas more complex objects or humans may be encased in multiple spherical constraints [DAO+07]. Linear and spherical constraints are discussed in the following two examples.

Example 2.2 (Linear constraints)**.** The most straightforward type of constraints are linear constraints. In their most general form

$$h_\mathrm{c}(\boldsymbol{x}, \boldsymbol{\eta}) = \boldsymbol{\eta}_a^\intercal \boldsymbol{x} - \eta_b \ , \qquad \text{with } \boldsymbol{\eta} = [\boldsymbol{\eta}_a^\intercal, \eta_b]^\intercal \tag{2.6}$$

they are, for example, useful to model environmental constraints such as walls or limit the state space of the controlled systems. Two special cases of these constraints provide the means to model lower limits

$$h_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{\eta}) = \eta_{\mathrm{low}} - x_i \ , \qquad \text{with } \boldsymbol{\eta} = \eta_{\mathrm{low}} \tag{2.7}$$

and upper limits on any state x_i

$$h_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{\eta}) = x_i - \eta_{\mathrm{up}} \ , \qquad \text{with } \boldsymbol{\eta} = \eta_{\mathrm{up}} \ . \tag{2.8}$$

An illustration of how upper and lower limits define constraint admissible state values is provided in Fig. 2.2a. Each constraint cuts the state space in an admissible ($h_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{\eta}) \leq 0$) and an inadmissible ($h_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{\eta}) > 0$) half-plane. If an upper and a lower limit are combined, the admissible half-planes of the two constraints need to overlap, i.e. $\eta_{\mathrm{low}} < \eta_{\mathrm{up}}$, as depicted in the figure for admissible states to exist.

Example 2.3 (Spherical constraints)**.** Another useful constraint type are spherical limits. They either render all states outside of the spherical constraint admissible

$$h_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{\eta}) = \eta_r - \|\boldsymbol{x} - \boldsymbol{\eta}_m\|_2 \ , \qquad \boldsymbol{\eta} = [\boldsymbol{\eta}_m^{\mathsf{T}}, \eta_r]^{\mathsf{T}} \tag{2.9}$$

or all states inside the spherical constraint admissible

$$h_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{\eta}) = \|\boldsymbol{x} - \boldsymbol{\eta}_m\|_2 - \eta_r \ , \qquad \boldsymbol{\eta} = [\boldsymbol{\eta}_m^{\mathsf{T}}, \eta_r]^{\mathsf{T}} \ . \tag{2.10}$$

The first type allows the definition of restricted areas in the state space as required for obstacles. Different shapes are achieved by a combination of multiple spherical constraints. For the second type, the restriction of the state space is more severe as this constraint renders only the states inside the spherical constraint admissible. Both types of spherical constraints in two-dimensional space are illustrated in Fig. 2.2b and 2.2c. The admissible states ($h_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{\eta}) \leq 0$) of the first type in Fig. 2.2b lie outside the sphere, while states within the sphere are inadmissible ($h_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{\eta}) > 0$). Contrary to that, the admissible states ($h_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{\eta}) \leq 0$) of the second type in Fig. 2.2c lie within the sphere, while inadmissible states ($h_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{\eta}) > 0$) are on the outside.

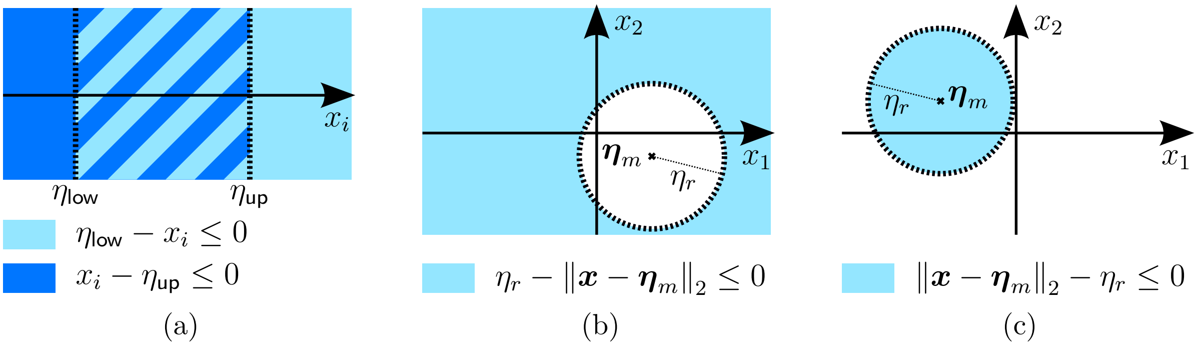
Naturally, many applications require the definition of more than one constraint. In order to define arbitrarily many constraints, the individual constraint functions are concatenated in a vector

$$\boldsymbol{y}_{\mathrm{c}} = \boldsymbol{h}_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{\eta}) = \begin{bmatrix} h_{\mathrm{c},1}(\boldsymbol{x},\boldsymbol{\eta}) \\ \vdots \\ h_{\mathrm{c},l}(\boldsymbol{x},\boldsymbol{\eta}) \end{bmatrix} \ , \tag{2.11}$$

where the indices of the constraints are taken from the ***set of constraints***

$$\mathcal{B} = \{ i \in \mathbb{N} \mid 1 \leq i \leq l \} \ . \tag{2.12}$$

Based on the constraint functions, it is possible to identify a constraint admissible subset of the state space.


x_i
η_{low}
η_{up}
$\eta_{\text{low}} - x_i \leq 0$
$x_i - \eta_{\text{up}} \leq 0$
x_2
x_1
$\boldsymbol{\eta}_m$
η_r
$\eta_r - \|\boldsymbol{x} - \boldsymbol{\eta}_m\|_2 \leq 0$
x_2
x_1
η_r
$\boldsymbol{\eta}_m$
$\|\boldsymbol{x} - \boldsymbol{\eta}_m\|_2 - \eta_r \leq 0$

(a) (b) (c)

Figure 2.2.: Illustration of admissible state values, i.e. $h_{\text{c}}(\boldsymbol{x}, \boldsymbol{\eta}) \leq 0$, for (a) upper and lower bounds on a state x_i, (b) a spherical constraint, where the outside of the sphere is admissible, for $\boldsymbol{x} \in \mathbb{R}^2$ and (c) a spherical constraint, where the inside of the sphere is admissible, for $\boldsymbol{x} \in \mathbb{R}^2$.

Definition 2.2. The ***admissible set*** of states contains the states which adhere to all constraints and is defined as

$$\mathcal{H}(\boldsymbol{\eta}) = \{\boldsymbol{x} \mid h_{\text{c},i}(\boldsymbol{x}, \boldsymbol{\eta}) \leq 0 \quad \forall\, i \in \mathcal{B}\} \ .$$

In order to guarantee adherence to all limits, the used control law should keep the states of the system (2.1) or (2.3) within the admissible set for all times. The control approaches proposed in this work rely on I/O-linearization to find an appropriate control action, generating a necessity for the following assumption.

Assumption 2.2. *For all $i \in \mathcal{B}$, there exist constants $r_i \in \mathbb{N}$ such that each of the following conditions is fulfilled:*

(i) $h_{c,i}(\boldsymbol{x}, \boldsymbol{\eta})$ is r_i times differentiable with respect to time and for a continuous input $h_{c,i}(\boldsymbol{x}, \boldsymbol{\eta}) \in \mathcal{C}^{r_i}$ holds $\forall\ i \in \mathcal{B}$,

(ii) each $h_{c,i}(\boldsymbol{x}, \boldsymbol{\eta})$ fulfills $\forall\ \boldsymbol{\eta} \in \mathcal{D}_\eta$

$$\begin{aligned} \mathscr{L}_{\boldsymbol{x}\boldsymbol{G}} \mathscr{L}_{\boldsymbol{x}\boldsymbol{f}}^{k} h_{c,i}(\boldsymbol{x}, \boldsymbol{\eta}) &= \boldsymbol{0}_{1\times m} \quad \forall\, 0 \leq k < r_i - 1 \ , \\ \mathscr{L}_{\boldsymbol{x}\boldsymbol{G}} \mathscr{L}_{\boldsymbol{x}\boldsymbol{f}}^{r_i - 1} h_{c,i}(\boldsymbol{x}, \boldsymbol{\eta}) &\neq \boldsymbol{0}_{1\times m} \end{aligned}$$

(iii) each η_j, $1 \leq j \leq n_\eta$ is $r_{\max} + 1$ times differentiable with respect to time and $\eta_j \in \mathcal{C}^{r_{\max}}$ with

$$r_{\max} = \max_{i \in \mathcal{B}}(r_i) \ ,$$

(iv) each η_j, $1 \leq j \leq n_\eta$ and its $r_{\max}$ derivatives are measurable and bounded,

(v) $\mathcal{H}(\boldsymbol{\eta})$ is non-empty, i.e. $\mathcal{H}(\boldsymbol{\eta}) \neq \emptyset$.

The necessity for conditions (i)–(iii) follows directly from the requirements of I/O-linearization. By designing the constraint output accordingly, it is ensured that the condition holds. In view of the application, the constraints have to be designed such that the admissible set does not include any singularities as these lead to a loss of manipulability and unsafe behavior. Without any singularities in the admissible set, controllability is achieved and intuitively, this means that it is possible to apply inputs which result in a motion away from the constraint. In addition, condition (v) is natural, since it would be impossible for the controlled system to adhere to all constraints if no admissible states exist.

Remark 2.2. The admissible set may also consist of multiple disjoint admissible sets

$$\mathcal{H} = \mathcal{H}_1 \cup \ldots \cup \mathcal{H}_{l_H} \ ,$$

where each set $\mathcal{H}_i$, $i \in \{1, \ldots, l_H\}$ is defined by a set of constraints according to Definition 2.2. In order to achieve constraint adherence in this case, the set $\mathcal{H}_i$, which contains the initial state, has to fulfill condition (v).

The restrictions, which (iii) and (iv) impose on the parameter dynamics, are owed to the requirements of I/O-linearization used for the control derivation. Parameter dynamics generated by jerk controlled systems, such as human motion [FH85], fulfill these requirements for $r_{\max} \leq 2$. If the conditions on the parameters do not hold naturally, the parameter variation has to be represented by a sufficiently smooth approximation. If it is not possible to measure $\boldsymbol{\eta}$, an estimation, e.g. using an observer, of the parameters has to be included.

In this work, the dynamics generating the parameters $\boldsymbol{\eta} \in \mathbb{R}^{n_\eta}$ are modeled by

$$\underbrace{\begin{bmatrix} \dot{\boldsymbol{\eta}} \\ \vdots \\ \boldsymbol{\eta}^{(r_{\max})} \\ \boldsymbol{\eta}^{(r_{\max}+1)} \end{bmatrix}}_{\dot{\boldsymbol{x}}_\eta} = \underbrace{\begin{bmatrix} \mathbf{0}_{n_\eta} & \boldsymbol{I}_{n_\eta} & \ldots & \mathbf{0}_{n_\eta} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0}_{n_\eta} & \ldots & \mathbf{0}_{n_\eta} & \boldsymbol{I}_{n_\eta} \\ \mathbf{0}_{n_\eta} & \ldots & \mathbf{0}_{n_\eta} & \mathbf{0}_{n_\eta} \end{bmatrix} \underbrace{\begin{bmatrix} \boldsymbol{\eta} \\ \vdots \\ \boldsymbol{\eta}^{(r_{\max}-1)} \\ \boldsymbol{\eta}^{(r_{\max})} \end{bmatrix}}_{\boldsymbol{x}_\eta}}_{\boldsymbol{f}_\eta(\boldsymbol{x}_\eta)} + \begin{bmatrix} \mathbf{0}_{n_\eta} \\ \vdots \\ \mathbf{0}_{n_\eta} \\ \boldsymbol{I}_{n_\eta} \end{bmatrix} \boldsymbol{u}_{\boldsymbol{\eta}} \ , \quad \boldsymbol{x}_\eta(t_0) = \boldsymbol{x}_{\eta,0} \tag{2.13}$$

with the state $\boldsymbol{x}_\eta \in \mathbb{R}^{(r_{\max}+1)n_\eta}$, the input $\boldsymbol{u}_{\boldsymbol{\eta}} \in \mathbb{R}^{n_\eta}$ and with $r_{\max}$ according to Assumption 2.2. This may seem rather restrictive at first, however, any parameter dynamics fulfilling Assumption 2.2 may be transformed into the above form using I/O-linearization with the output $\boldsymbol{\eta}$ and the input $\boldsymbol{u}_{\boldsymbol{\eta}} = \boldsymbol{\eta}^{(r_{\max}+1)}$.

Remark 2.3. Static constraints are modeled using constant parameters $\boldsymbol{\eta}$, which is achieved by choosing $\boldsymbol{x}_{\eta,0} = [\boldsymbol{\eta}^\intercal, \mathbf{0}_{1\times n_\eta}, \ldots, \mathbf{0}_{1\times n_\eta}]^\intercal$ and $\boldsymbol{u}_{\boldsymbol{\eta}} = \mathbf{0}_{n_\eta \times 1}$ in (2.13).

2.2.1 Input/Output-linearization

For constraints fulfilling Assumption 2.2, it is possible to perform an I/O-linearization of system (2.1) with respect to the outputs given by the constraints (2.11). This allows the determination of the influence of the system input $\boldsymbol{u}$ on the change of the constraint functions. I/O-linearization is achieved by deriving each element of (2.11) with respect to time until $\boldsymbol{u}$

appears explicitly. The linearizing input transformation is then given by

$$\begin{aligned} z_i = y_{\mathrm{c},i}^{(r_i)} &= \boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}(\boldsymbol{x},\boldsymbol{\eta})\boldsymbol{u} + b_{\mathrm{c},i}(\boldsymbol{x},\boldsymbol{x}_\eta) \ , \\ \text{with } \boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}(\boldsymbol{x},\boldsymbol{\eta}) &= \mathscr{L}_{\boldsymbol{x}\boldsymbol{G}}\mathscr{L}_{\boldsymbol{x}\boldsymbol{f}}^{r_i-1} y_{\mathrm{c},i} \ , \\ b_{\mathrm{c},i}(\boldsymbol{x},\boldsymbol{x}_\eta) &= \left(\mathscr{L}_{\boldsymbol{x}\boldsymbol{f}} + \mathscr{L}_{\boldsymbol{x}_\eta \boldsymbol{f}_\eta}\right)^{r_i} y_{\mathrm{c},i} \ , \end{aligned} \tag{2.14}$$

which is determined by deriving each constraint function r_i times, where r_i is the so-called ***relative degree***. Note that with Assumption 2.2, $\boldsymbol{a}_i(\boldsymbol{x},\boldsymbol{\eta}) \neq \boldsymbol{0}_{m\times 1}$ holds, which is required for a well-defined relative degree r_i and hence a valid I/O-linearization [Kha96; SS12, p.508ff.].

For the vectorial output (2.11), the individual transformations (3.1) are concatenated into matrix-vector notation

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{A}_{\mathrm{c},\mathcal{B}}(\boldsymbol{x},\boldsymbol{\eta})\boldsymbol{u} + \boldsymbol{b}_{\mathrm{c},\mathcal{B}}(\boldsymbol{x},\boldsymbol{x}_\eta) \ , \\ \text{with } \boldsymbol{A}_{\mathrm{c},\mathcal{B}}(\boldsymbol{x},\boldsymbol{\eta}) &= [\boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}(\boldsymbol{x},\boldsymbol{\eta})]_{i\in\mathcal{B}} = \begin{bmatrix} \boldsymbol{a}_{\mathrm{c},1}^{\mathsf{T}}(\boldsymbol{x},\boldsymbol{\eta}) \\ \vdots \\ \boldsymbol{a}_{\mathrm{c},l}^{\mathsf{T}}(\boldsymbol{x},\boldsymbol{\eta}) \end{bmatrix} \ , \\ \boldsymbol{b}_{\mathrm{c},\mathcal{B}}(\boldsymbol{x},\boldsymbol{x}_\eta) &= [b_{\mathrm{c},i}(\boldsymbol{x},\boldsymbol{x}_\eta)]_{i\in\mathcal{B}} = \begin{bmatrix} b_{\mathrm{c},1}(\boldsymbol{x},\boldsymbol{x}_\eta) & \dots & b_{\mathrm{c},l}(\boldsymbol{x},\boldsymbol{x}_\eta) \end{bmatrix}^{\mathsf{T}} \ . \end{aligned} \tag{2.15}$$

Such a vectorial output is associated with a vector relative degree $\boldsymbol{r} = [r_1 \ \dots \ r_l]^{\mathsf{T}}$. A necessary and sufficient condition for the existence of a I/O-linearizing transformation for a vectorial output is a well-defined vector relative degree [Isi95].

Definition 2.3. The control affine system (2.1) has a ***well-defined vector relative degree*** given by $\boldsymbol{r} = [r_1 \ \dots \ r_l]^{\mathsf{T}}$ on a subset $\mathcal{X}$ of the state space if the following conditions are fulfilled for all $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$, $\boldsymbol{x}_\eta \in \mathbb{R}^{(r_{\max}+1)n_\eta}$:

1. $\mathscr{L}_{\boldsymbol{x}\boldsymbol{G}}\mathscr{L}_{\boldsymbol{x}\boldsymbol{f}}^{k} y_{\mathrm{c},i} = \boldsymbol{0}_{1\times m} \ \forall \ 0 \leq k < r_i - 1, \ 1 \leq i \leq l,$
2. the decoupling matrix $\boldsymbol{A}_{\mathrm{c},\mathcal{B}} \in \mathbb{R}^{l\times m}$ has full row rank.

$$\operatorname{rank}(\boldsymbol{A}_{\mathcal{B}}(\boldsymbol{x},\boldsymbol{\eta})) = \operatorname{rank}\left(\begin{bmatrix} \mathscr{L}_{\boldsymbol{x}\boldsymbol{G}}\mathscr{L}_{\boldsymbol{x}\boldsymbol{f}}^{r_1-1} y_{\mathrm{c},1} \\ \vdots \\ \mathscr{L}_{\boldsymbol{x}\boldsymbol{G}}\mathscr{L}_{\boldsymbol{x}\boldsymbol{f}}^{r_l-1} y_{\mathrm{c},l} \end{bmatrix}\right) = l$$

The first condition is straightforwardly fulfilled by Assumption 2.2. However, if the number of constraints is larger than the input dimension $l > m$, the second condition is not fulfilled. In addition, I/O-linearization may result in potentially unstable internal dynamics if the system is not fully linearized. The existence of internal dynamics is indicted by the total relative degree.

Definition 2.4. The ***total relative degree*** r corresponding to an output with a well-defined vector relative degree $\boldsymbol{r} = [r_1 \ \dots \ r_l]$ is given by the sum of the elements of the vector relative degree.

$$r = \sum_{i=1}^{l} r_i \tag{2.16}$$

Note that for scalar outputs, the total relative degree is equal to the relative degree. Then, if the total relative is smaller than the number of states in the system, internal dynamics are present [Kha96, p.508ff.].

The control schemes derived in this thesis, do, however, not rely on an exact I/O-linearization of the system and stability properties are analyzed separately. As the linearizing transformation (2.15) is solely a means to determine the influence of the input on the output, it is no problem if internal dynamics occur or if the number of constraints is larger than the input dimension even though the vector relative degree is not well-defined in this case.

Up to this point, we implicitly assume that the constraint parameters are exactly known. In some applications, however, this may not be the case.

2.2.2 Uncertain Parameters

If the constraint parameters are subject to uncertainties, either due to a smooth estimation of the parameter dynamics, due to stochastic uncertainties in measurements or because the model is not exactly known, this introduces the need to model constraints with uncertain parameters. These uncertainties are represented in the following way.

Assumption 2.3. *The uncertain parameters $\boldsymbol{x}_\eta$ are given by*

$$\boldsymbol{x}_\eta = \overline{\boldsymbol{x}_\eta} + \boldsymbol{\Delta}_{\boldsymbol{\eta}} \ , \qquad \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} = \begin{bmatrix} \overline{\boldsymbol{\eta}} \\ \dot{\overline{\boldsymbol{\eta}}} \\ \vdots \\ \overline{\boldsymbol{\eta}}^{(\boldsymbol{r}_{\mathbf{max}})} \end{bmatrix} \ , \qquad \boldsymbol{\Delta}_{\boldsymbol{\eta}} = \begin{bmatrix} \boldsymbol{\delta}_\eta \\ \boldsymbol{\delta}_{\dot{\eta}} \\ \vdots \\ \boldsymbol{\delta}_{\eta^{(r_{\max})}} \end{bmatrix} \tag{2.17}$$

where the elements of $\overline{\boldsymbol{x}_{\boldsymbol{\eta}}}$ are smooth trajectories and $\boldsymbol{\Delta}_{\boldsymbol{\eta}}$ represents the stochastically distributed variations of the parameter values from the trajectories.

This parameter model splits the parameters into a continuous part $\overline{\boldsymbol{x}_{\boldsymbol{\eta}}}$, which models the general trend of the parameter variation, and a stochastic part, which differs for each instance of the uncertainty according to the underlying uncertainty distribution. A fitting representation for $\overline{\boldsymbol{x}_{\boldsymbol{\eta}}}$ is, for example, the expected value of the parameter distribution, but other definitions may be considered as well. Furthermore, we introduce the set of uncertainties $\mathcal{D} \subseteq \mathbb{R}^{n_\eta(r_{\max}+1)}$, which contains all possible instances of $\boldsymbol{\Delta}_{\boldsymbol{\eta}}$, i.e.

$$\mathscr{P}(\boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}) = \mathscr{P}(\mathcal{D}) = 1 \ . \tag{2.18}$$

For now, we do not make any assumptions on this set or the underlying probabilistic distribution.

2.3 Task-oriented Control

In addition to the system dynamics, the constraint definition and a constraint enforcing control law, the envisioned control loop in Fig. 2.1 includes a task-oriented control law also called nominal control. Using a desired behavior, for example a desired state trajectory, the state and/or output of the system, this control law yields an input $\boldsymbol{u}_{\text{no}}$, which is designed without taking any constraints into account, solely according to task specifications and the performance goal. In this work, we assume that nominal control exhibits the following stability property.

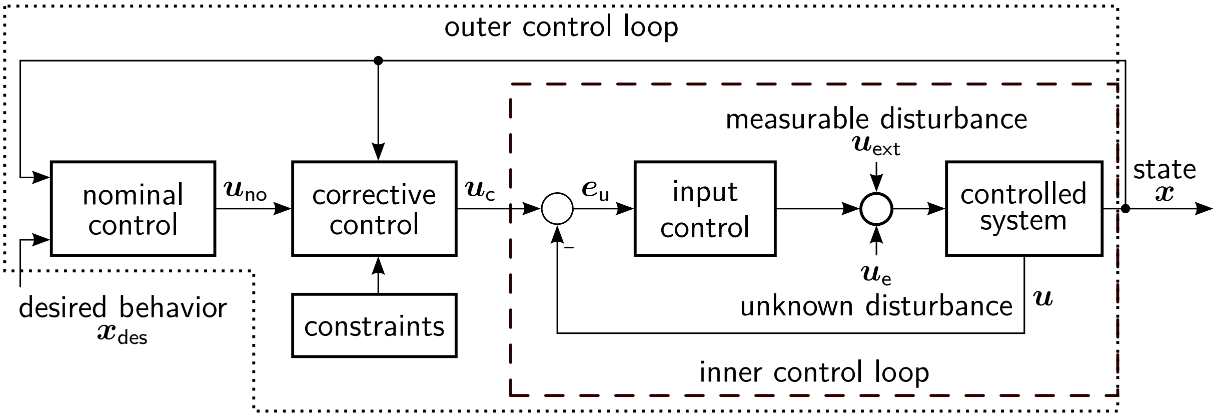


Figure 2.3.: Structure of a system with external disturbances (measurable and unmeasurable) controlled by two cascaded control loops and add-on constraint-enforcing corrective control.

Assumption 2.4. *Nominal control globally stabilizes the tracking error* $\boldsymbol{e}_{\boldsymbol{x}} = \boldsymbol{x}_{des} - \boldsymbol{x}$ *in the sense of Lyapunov, where* $\boldsymbol{x}_{des}$ *is a sufficiently smooth desired trajectory, if no external disturbances occur.*

With this assumption, stable tracking is guaranteed if nominal control is applied and no disturbances are present. As stability is the most basic design goal, the assumption is generally fulfilled for control schemes derived from explicit goal descriptions. If goal descriptions are given implicitly as for example in learning-based approaches, stability properties are not immediately clear. This issue may be resolved by restricting the set of control laws, which makes up the basis for the learning process, to solely stabilizing control laws.

Remark 2.4. Assumption 2.4 is required for the stability investigation of the controlled system in the following chapters. It may, however, be disregarded if stability is a subordinate goal as constraint adherence will be independent from any stability properties. This allows the use of other stability notions.

Extending the framework of Fig. 2.1, some applications involve cascaded control schemes and external disturbances as depicted in Fig. 2.3. Here, direct application of the control input generated by the serial connection of nominal control $\boldsymbol{u}_{\text{no}} \in \mathbb{R}^m$ and corrective control $\boldsymbol{u}_{\text{c}} \in \mathbb{R}^m$ to the system does not result in the desired behavior. Both the measurable disturbances $\boldsymbol{u}_{\text{ext}}$, such as for example forces and torques occurring in physical interaction of robots with humans and the environment, and the unmeasurable disturbances $\boldsymbol{u}_{\text{e}}$, resulting from measurement errors and unmodeled dynamics, lead to a distortion of the input. Therefore, an inner control loop is included which stabilizes the error between the control input from the outer loop $\boldsymbol{u}_{\text{c}}$ and the system input $\boldsymbol{u}$

$$\boldsymbol{e}_{\text{u}} = \boldsymbol{u}_{\text{c}} - \boldsymbol{u} \ . \tag{2.19}$$

As the goal is to achieve $\boldsymbol{u} = \boldsymbol{u}_{\text{c}} \in \mathbb{R}^m$, we make the following assumption on the control law in the inner control loop.

Assumption 2.5. *The inner loop is input-to-state stable (ISS) with respect to bounded inputs $\boldsymbol{u}_{ext}$ and $\boldsymbol{u}_e$, i.e.*

$$\|\boldsymbol{e}_u(t)\| \leq \alpha(\|\boldsymbol{e}_u(t_0)\|, t - t_0) + \beta \left(\sup_{t_0 \leq s \leq t} \left\| \begin{bmatrix} \boldsymbol{u}_{ext}(s) \\ \boldsymbol{u}_e(s) \end{bmatrix} \right\| \right)$$

with a class $\mathcal{KL}$ function α and a class $\mathcal{K}$ function β holds for any initial error $\boldsymbol{e}_u(t_0)$ and all $t \geq t_0$.

The condition imposed by the assumption is met by choosing an appropriate control law for input control. However, after the addition of the inner control loop in between corrective control and the system, it is not immediately clear, whether the requirements introduced in Section 2.1 are still fulfilled. Regarding the inner control loop as the new system, the structure reminds of Fig. 2.1. The system represented by the inner control loop has the dynamics

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})(\boldsymbol{u}_\mathrm{c} - \boldsymbol{e}_\mathrm{u}) \tag{2.20}$$

and is therefore control affine with respect to the new input $\boldsymbol{u}_\mathrm{c}$, thus fulfilling the system requirements. From this formulation, it is clear that knowledge of the input error $\boldsymbol{e}_\mathrm{u} \in \mathbb{R}^m$ is required for a full system description.

Assumption 2.6. *The input error $\boldsymbol{e}_\mathrm{u}$ from* (2.19) *is known and bounded.*

As $\boldsymbol{e}_\mathrm{u}$ is used for input control, it is safe to assume that it is generally available. This also explains the positioning of corrective control in between nominal control and input control instead of between input control and system. The second option would require exact knowledge of all disturbances for a full system model, which is not available as some disturbances are assumed to be unmeasurable and therefore unavailable in contrast to $\boldsymbol{e}_\mathrm{u}$, which is available. In view of the application to robotic systems, consider the following remark.

Remark 2.5. For torque-controlled robotic systems, the previous considerations are directly applicable [KH17] and by adapting the control law accordingly, it is applicable to position-controlled robots as well. In that case, the corrective control law needs to be included in the generation of the desired position.

In the following, we will give some examples for well-known control laws that are commonly used in the control of robotic systems.

Example 2.4 (Example 2.1 continued)**.** In torque controlled robotic systems, the control structure is often similar to Fig. 2.3, where the inner control loop is the torque control loop for the input $\boldsymbol{u} = \boldsymbol{\tau}$ of the robotic system (2.5). Apart from measurement and modeling errors, these systems are subject to external forces and torques caused by physical interaction or collision with the environment or with humans. In order to eliminate the effects of the disturbances on the task execution, the inner torque control loop may be designed using the concept of passivity, which leads to a proportional-derivative (PD) control law [AEG+08].

In general, the control goal of the outer loop is given in task coordinates $\boldsymbol{p} \in \mathbb{R}^{n_p}$, which are a function $\boldsymbol{f}_p : \mathbb{R}^{n_q} \to \mathbb{R}^{n_p}$ of the generalized coordinates $\boldsymbol{q} \in \mathbb{R}^{n_q}$

$$\boldsymbol{p} = \boldsymbol{f}_p(\boldsymbol{q}) \; . \tag{2.21}$$

The Jacobian matrix $\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}} \in \mathbb{R}^{n_p \times n_q}$ relates the task and generalized velocities as well as the forces/torques $\boldsymbol{f}_{\text{ext}} \in \mathbb{R}^{n_p}$ in task space and the generalized torques $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^{n_q}$ to each other

$$\dot{\boldsymbol{p}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.22}$$

$$\boldsymbol{\tau}_{\text{ext}} = \boldsymbol{J}(\boldsymbol{q})^{\intercal}\boldsymbol{f}_{\text{ext}} \; . \tag{2.23}$$

Examples for the control law of the outer control loop generating the nominal input $\boldsymbol{\tau}_{\text{no}} \in \mathbb{R}^{n_p}$ range from motion control such as proportional-integral-derivative (PID), tracking or computed torque control [SK08, Ch. 6] to force control with stiffness and impedance control [SK08, Ch. 7]. Stiffness and impedance control are especially helpful, if compliance to external forces and torques is required. By imitating spring damper behavior, these approaches achieve compliance while tracking a desired trajectory. Hence, any desired motion may be actively changed by the application of forces by a cooperating human. Depending on the control goal, implementation in task coordinates [AOF+03] or in joint coordinates [OAK+08] may be chosen. In task space, for example, the impedance control law is given by

$$\boldsymbol{\tau}_{\text{no}} = \boldsymbol{J}(\boldsymbol{q})^{\intercal}(\boldsymbol{f}_{\text{ext}} + \boldsymbol{M}_{\boldsymbol{p}}\ddot{\boldsymbol{p}}_{\text{des}} + \boldsymbol{D}_{\boldsymbol{p}}(\dot{\boldsymbol{p}}_{\text{des}} - \dot{\boldsymbol{p}}) + \boldsymbol{K}_{\boldsymbol{p}}(\boldsymbol{p}_{\text{des}} - \boldsymbol{p})) + \boldsymbol{C}_{\boldsymbol{q}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_q(\boldsymbol{q}) \tag{2.24}$$

with $\boldsymbol{M}_{\boldsymbol{p}} \in \mathbb{R}^{n_p \times n_p}$, $\boldsymbol{K}_{\boldsymbol{p}} \in \mathbb{R}^{n_p \times n_p}$ and $\boldsymbol{D}_{\boldsymbol{p}} \in \mathbb{R}^{n_p \times n_p}$ being the positive definite Cartesian mass, stiffness and damping matrices, respectively. The desired trajectory $\boldsymbol{p}_{\text{des}} \in \mathbb{R}^{n_p}$ is assumed to be sufficiently smooth to allow for two time derivations and (2.23) relates the external forces/torques $\boldsymbol{f}_{\text{ext}}$ to the corresponding generalized torque $\boldsymbol{\tau}_{\text{ext}}$. The stiffness and damping matrices are chosen to fit the task requirements and may be adapted online, if necessary [BT04].

3

Control Methods for Guaranteed Constraint Satisfaction

Guaranteeing the satisfaction of possibly dynamic constraints while successfully completing the control goal is at the same time challenging and essential in safety-critical control applications. The used control scheme needs to be able to handle high numbers of constraints in real-time even for fast sampling times. Furthermore, especially in scenarios involving the physical interaction with humans, the constraint satisfaction should not deteriorate the quality of the interaction experience. Hence, ideally the robotic system adjusts its dynamics to mimic human-human interaction and to meet the human's expectations.

Related Work and Open Problems

Methods from collision avoidance such as potential fields [RK92] or virtual fixtures [Ros93] are designed to achieve collision-free motions in real-time. They are, however, unable to guarantee constraint satisfaction for all manipulator dynamics. Control approaches such as MPC [MRR+00], reference governors [GK02] and those based on reachable [Gil13] or invariant sets [Bla99] are the most well-known approaches for guaranteeing constraint satisfaction. They are able to handle state and output constraints. However, as they involve costly optimizations or set calculations, finding a solution in real-time might prove hard if not impossible.

This chapter introduces how input/output (I/O)-linearization may be used to design control schemes for the satisfaction of state and/or output constraints. Invariance control [KH15; KH16; KH17] and control barrier function (CBF)-based control [RKH16] are added to arbitrary nominal control laws to be able to satisfy dynamic constraints while executing the imposed task. Both control schemes may be implemented as an add-on to nominal control thus leading to the structure of the control loop being similar to the envisioned structure in Fig. 2.1. As the switching control law of invariance control is prone to chattering at boundaries, which may deteriorate the performance and the quality of the outcome, we additionally introduce a method for chattering reduction from [KH14] and augmented invariance control [KJH16], which allows for more adaptability in the behavior of the controlled system.

3.1 Invariance Control

In the following, we will give a thorough introduction and analysis of invariance control with dynamic constraints. Invariance control usually assumes an undisturbed system. External disturbances such as forces in human-robot interaction, however, lead at least temporarily to an input error $\boldsymbol{e}_u$ different from zero. Therefore, we will explicitly include the effects of an input error in the control scheme. Invariance control is introduced for MIMO systems with state constraints and input disturbances but application to systems with single input and/or single output with output constraints or without input disturbances is straightforward.

In this section, invariance control is developed for systems controlled in continuous time. Based on the system equations with input disturbances (2.20) and the constraint description according to Def. 2.1 and (2.11), we determine a corrective control input, which achieves constraint satisfaction in the presence of external disturbances caused for example by model and measurement uncertainties. The control derivation relies on determining the influence of an input signal on the value of the constraint function.

3.1.1 Input/Output-linearization

Feedback linearization with respect to the constraint functions (2.11) determines the influence of the input $\boldsymbol{u}$ on the change of the constraint function. The input transformation is determined analog to (2.14). Including the input error, it is given by

$$
\begin{aligned}
z_i = y_{\mathrm{c},i}^{(r_i)} &= \boldsymbol{a}_{\mathrm{c},i}^{\intercal}(\boldsymbol{x},\boldsymbol{\eta})\boldsymbol{u} + b_{\mathrm{c},i}(\boldsymbol{x},\boldsymbol{x}_{\boldsymbol{\eta}},\boldsymbol{e}_{\boldsymbol{u}}) \ , \qquad (3.1)\\
\text{with } \boldsymbol{a}_{\mathrm{c},i}^{\intercal}(\boldsymbol{x},\boldsymbol{\eta}) &= \mathscr{L}_{\boldsymbol{x}\boldsymbol{G}}\mathscr{L}_{\boldsymbol{x}\boldsymbol{f}}^{r_i-1} y_{\mathrm{c},i} \ ,\\
b_{\mathrm{c},i}(\boldsymbol{x},\boldsymbol{x}_{\boldsymbol{\eta}},\boldsymbol{e}_{\boldsymbol{u}}) &= \left(\mathscr{L}_{\boldsymbol{x}\boldsymbol{f}} + \mathscr{L}_{\boldsymbol{x}_{\boldsymbol{\eta}}\boldsymbol{f}_{\boldsymbol{\eta}}}\right)^{r_i} y_{\mathrm{c},i} + \boldsymbol{a}_{\mathrm{c},i}^{\intercal}(\boldsymbol{x},\boldsymbol{\eta})\boldsymbol{e}_{\boldsymbol{u}} \ .
\end{aligned}
$$

While the output remains the constraint function $y_{\mathrm{c},i}$, the input is replaced by the new pseudo input z_i, output and input now being connected by a time-invariant integrator chain with r_i integrators [PK97].

Example 3.1 (Example 2.1 continued)**.** In robotic systems (2.5), relative degrees one or two are the most relevant as they correspond to joint, task space and velocity constraints. In this example, we assume that no input disturbances are present, i.e. $\boldsymbol{e}_{\boldsymbol{u}} = \boldsymbol{0}_{m\times 1}$. Since the type of constraint function determines the relative degree, we distinguish two cases: velocity constraint functions $y_{\mathrm{c},i} = h_{\mathrm{c},i}(\boldsymbol{x},\boldsymbol{\eta})$ depending on joint velocities and optionally on the joint positions and position constraint functions $y_{\mathrm{c},i} = h_{\mathrm{c},i}(\boldsymbol{q},\boldsymbol{\eta})$ depending solely on the joint positions.

Velocity constraint function: The constraint functions $y_{\mathrm{c},i} = h_{\mathrm{c},i}(\boldsymbol{x},\boldsymbol{\eta})$ represent for example joint velocity limits or task space velocity limits. If the velocity constraint is on joint level, the output function only depends on the joint velocities $\dot{\boldsymbol{q}}$, whereas if the constraint is on task space level, the output function depends on the joint velocities $\dot{\boldsymbol{q}}$ as well as on the joints $\boldsymbol{q}$ due to the transformation (2.22). Differentiation with respect to time in combination with the system equation (2.5) yields

$$\begin{aligned}
\dot{y}_{\mathrm{c},i} &= \frac{\partial h_{\mathrm{c},i}}{\partial \dot{\boldsymbol{q}}}\ddot{\boldsymbol{q}} + \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{\eta}}\dot{\boldsymbol{\eta}} \\
&= \frac{\partial h_{\mathrm{c},i}}{\partial \dot{\boldsymbol{q}}}\boldsymbol{M}_{\boldsymbol{q}}{}^{-1}\boldsymbol{\tau} - \frac{\partial h_{\mathrm{c},i}}{\partial \dot{\boldsymbol{q}}}\boldsymbol{M}_{\boldsymbol{q}}{}^{-1}\left(\boldsymbol{C}_{\boldsymbol{q}}\dot{\boldsymbol{q}} + \boldsymbol{g}_{\boldsymbol{q}}\right) + \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{\eta}}\dot{\boldsymbol{\eta}} \; .
\end{aligned}$$

As the system input $\boldsymbol{\tau}$ appears already in the first time-derivative, $r_i = 1$ holds and the I/O-linearizing transformation (3.1) is determined by

$$\begin{aligned}
\boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}(\boldsymbol{x},\boldsymbol{\eta}) &= \frac{\partial h_{\mathrm{c},i}}{\partial \dot{\boldsymbol{q}}}\boldsymbol{M}_{\boldsymbol{q}}{}^{-1} && (3.2) \\
b_{\mathrm{c},i}(\boldsymbol{x},\boldsymbol{x}_{\eta}) &= -\frac{\partial h_{\mathrm{c},i}}{\partial \dot{\boldsymbol{q}}}\boldsymbol{M}_{\boldsymbol{q}}{}^{-1}\left(\boldsymbol{C}_{\boldsymbol{q}}\dot{\boldsymbol{q}} + \boldsymbol{g}_{\boldsymbol{q}}\right) + \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{\eta}}\dot{\boldsymbol{\eta}} \; . && (3.3)
\end{aligned}$$

Position constraint function: The constraint functions $y_{\mathrm{c},i} = h_{\mathrm{c},i}(\boldsymbol{q},\boldsymbol{\eta})$ may represent, for example, joint limits or bounds in task space. As task space limits may be transformed into joint space using (2.21), the functions depend solely on the joint positions $\boldsymbol{q}$ and differentiation with respect to time yields

$$\begin{aligned}
\dot{y}_{\mathrm{c},i} &= \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{\eta}}\dot{\boldsymbol{\eta}} \; , \\
\ddot{y}_{\mathrm{c},i} &= \frac{\partial \dot{y}_{\mathrm{c},i}}{\partial \dot{\boldsymbol{q}}}\ddot{\boldsymbol{q}} + \frac{\partial \dot{y}_{\mathrm{c},i}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial \dot{y}_{\mathrm{c},i}}{\partial \boldsymbol{\eta}}\dot{\boldsymbol{\eta}} + \frac{\partial \dot{y}_{\mathrm{c},i}}{\partial \dot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} \\
&= \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\boldsymbol{M}_{\boldsymbol{q}}{}^{-1}\boldsymbol{\tau} - \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\boldsymbol{M}_{\boldsymbol{q}}{}^{-1}\left(\boldsymbol{C}_{\boldsymbol{q}}\dot{\boldsymbol{q}} + \boldsymbol{g}_{\boldsymbol{q}}\right) \\
&\quad + \frac{\partial}{\partial \boldsymbol{q}}\left(\frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}} + 2\frac{\partial}{\partial \boldsymbol{\eta}}\left(\frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}\right)\dot{\boldsymbol{\eta}} + \frac{\partial}{\partial \boldsymbol{\eta}}\left(\frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{\eta}}\dot{\boldsymbol{\eta}}\right)\dot{\boldsymbol{\eta}} + \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{\eta}}\ddot{\boldsymbol{\eta}} \; .
\end{aligned}$$

Here, the system input $\boldsymbol{\tau}$ appears in the second time-derivative, i.e. $r_i = 2$ holds and the I/O-linearizing transformation (3.1) is determined by

$$\begin{aligned}
\boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}(\boldsymbol{x},\boldsymbol{\eta}) &= \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\boldsymbol{M}_{\boldsymbol{q}}{}^{-1} \; , && (3.4) \\
b_{\mathrm{c},i}(\boldsymbol{x},\boldsymbol{x}_{\eta}) &= -\frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\boldsymbol{M}_{\boldsymbol{q}}{}^{-1}\left(\boldsymbol{C}_{\boldsymbol{q}}\dot{\boldsymbol{q}} + \boldsymbol{g}_{\boldsymbol{q}}\right) + \frac{\partial}{\partial \boldsymbol{q}}\left(\frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}} + \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{\eta}}\ddot{\boldsymbol{\eta}} \\
&\quad + \frac{\partial}{\partial \boldsymbol{\eta}}\left(2\frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial h_{\mathrm{c},i}}{\partial \boldsymbol{\eta}}\dot{\boldsymbol{\eta}}\right)\dot{\boldsymbol{\eta}} \; . && (3.5)
\end{aligned}$$

Note that although both velocity and position constraint functions are defined in the generalized coordinates $\boldsymbol{q}$, task space constraints may be implemented as well. Using the forward kinematics (2.21) of the robotic system and the corresponding velocity transformation (2.22), it is possible to express task space constraints in the generalized coordinates, i.e. $h_i(\boldsymbol{p}) = h_i(\boldsymbol{f}_p(\boldsymbol{q}))$, $h_i(\dot{\boldsymbol{p}}) = h_i(\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}})$, and the I/O-linearization and the relative degree are determined by either (3.2)–(3.3) or (3.4)–(3.5).

3.1.2 Prediction of the Constraint Output

As the goal of invariance control is to avoid constraint violations, it is necessary to determine whether a certain input value will eventually lead to a positive constraint function value, i.e. a future constraint violation. The prediction of the constraint output value is enabled by the integrator chains resulting from I/O-linearization which allow the analytic derivation of how the output is influenced by the input. Note that these derivations would become much harder, if not impossible, to solve without the use of the linearizing transformation.

Making use of the integrator chain with the input $z_i(t)$ corresponding to constraint i, it is possible to predict the behavior of the output $y_{\mathrm{c},i}(t+\Delta t)$ for any future time $t+\Delta t$ with $\Delta t \geq 0$ by integration.

$$y_{\mathrm{c},i}^{r_i-1}(t+\Delta t) = y_{\mathrm{c},i}^{r_i-1}(t) + \int\limits_0^{\Delta t} z_i(t+\chi)\mathrm{d}\chi \tag{3.6}$$

$$\begin{aligned} y_{\mathrm{c},i}^{r_i-2}(t+\Delta t) &= y_{\mathrm{c},i}^{r_i-2}(t) + \int\limits_0^{\Delta t} y_{\mathrm{c},i}^{r_i-1}(t+\chi)\mathrm{d}\chi \\ &= y_{\mathrm{c},i}^{r_i-2}(t) + \int\limits_0^{\Delta t} \left(y_{\mathrm{c},i}^{r_i-1}(t) + \int\limits_0^{\chi} z_i(t+\chi)\mathrm{d}\chi \right) \mathrm{d}\chi \\ &= y_{\mathrm{c},i}^{r_i-2}(t) + \Delta t\ y_{\mathrm{c},i}^{r_i-1}(t) + \int\limits_0^{\Delta t} \left(\int\limits_0^{\chi} z_i(t+\chi)\mathrm{d}\chi \right) \mathrm{d}\chi \end{aligned} \tag{3.7}$$

$$\vdots$$

$$\begin{aligned} y_{\mathrm{c},i}(t+\Delta t) &= y_{\mathrm{c},i}(t) + \int\limits_0^{\Delta t} \dot{y}_{\mathrm{c},i}(t+\chi)\mathrm{d}\chi \\ &= \sum_{k=0}^{r_i-1} \frac{(\Delta t)^k}{k!} y_{\mathrm{c},i}^{(k)}(t) + \int\limits_0^{\Delta t}\int\limits_0^{\chi} \ldots \int\limits_0^{\chi} z_i(t+\chi)\mathrm{d}\chi^{r_i} \end{aligned} \tag{3.8}$$

If the pseudo input z_i is constant, the future output value is given by [WB04]

$$y_{\mathrm{c},i}(t+\Delta t) = \frac{(\Delta t)^{r_i}}{r_i!} z_i + \sum_{k=0}^{r_i-1} \frac{(\Delta t)^k}{k!} y_{\mathrm{c},i}^{(k)}(t). \tag{3.9}$$

These results may be used to calculate the instant of time, when a switch to a corrective action for constraint satisfaction is required.

3.1.3 Invariance Functions and Invariant Set

As a result of (3.6)–(3.8), setting the pseudo input z_i to a negative value $\gamma_i < 0$ eventually leads to a decrease in the constraint outputs and their derivatives, which constitute the states of the integrator chains. This means that for $z_i = \gamma_i < 0$, there exists a maximum value for the constraint output function, which it takes before starting to decrease [WB04].

Definition 3.1. The ***invariance function*** $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i)$ analytically determines the maximum future value, the constraint output function will take in case of a constant pseudo input $z_i = \gamma_i < 0$. It is given by

$$\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) = \max_{\Delta t \geq 0} p_i\left(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i, \Delta t\right) \tag{3.10}$$

$$\text{with } p_i\left(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i, \Delta t\right) = \frac{(\Delta t)^{r_i}}{r_i!}\gamma_i + \sum_{k=0}^{r_i-1} \frac{(\Delta t)^k}{k!} h_{\mathrm{c},i}^{(k)}(\boldsymbol{x}, \boldsymbol{x}_\eta) \ . \tag{3.11}$$

The invariance function allows for the continuous evaluation of the future maximum constraint function value. It combines the constraint function with the dynamics of the system and the relative degree to predict constraint violations. If it is negative, no violation is about to occur in the future as the constraint function remains negative and hence, no corrective action is required. For an invariance function value of zero, the corrective pseudo input $z_{\mathrm{c},i} = \gamma_i$ suffices to keep the system within the admissible set as in this case, the maximum future constraint function value is zero. For a positive invariance function, a constraint violation will probably occur in the future as the corrective pseudo input $z_{\mathrm{c},i} = \gamma_i$ does not suffice to keep the future constraint function value non-positive.

Example 3.2 (Invariance function for $r_i = 1$)**.** For $r_i = 1$, the integrator chain has a single state. The polynomial (3.11) is given by

$$p_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i, \Delta t) = \Delta t \gamma_i + h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta}) \ .$$

Deriving the function with respect to Δt

$$\frac{\partial p_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i, \Delta t)}{\partial \Delta t} = \gamma_i$$

shows that the constraint function is constant for $\gamma_i = 0$ and decreasing for $\gamma_i < 0$. As a result, the current value $h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta})$ is the maximum value the function will ever take for $\gamma_i \leq 0$ and the invariance function is hence given by

$$\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) = h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta}) \ . \tag{3.12}$$

Example 3.3 (Invariance function for $r_i = 2$)**.** For $r_i = 2$, the integrator chain has two states. The polynomial (3.11) is given by

$$p_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i, \Delta t) = \frac{\Delta t^2}{2}\gamma_i + \Delta t \ \dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta) + h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta}) \ .$$

Deriving the function with respect to $\Delta t \geq 0$

$$\frac{\partial p_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i, \Delta t)}{\partial \Delta t} = \Delta t \gamma_i + \dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta)$$

shows that two cases need to be distinguished when determining the invariance function. If $\dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta) \leq 0$ holds, the expression is negative for $\gamma < 0$ and hence the constraint function is decreasing from the maximum value $h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta})$. If $\dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta) > 0$ holds, the constraint function is at an extreme value for

$$\Delta t = -\frac{\dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta)}{\gamma_i} \ ,$$

which is a maximum due to

$$\frac{\partial^2 p_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i, \Delta t)}{\partial \Delta t^2} = \gamma_i < 0 \ .$$

Inserting the expression for Δt into $p_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i, \Delta t)$ yields the invariance function

$$\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) = \begin{cases} h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta}) & \dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta) \leq 0 \\ h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta}) - \dfrac{\dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta)^2}{2\gamma_i} & \dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta) > 0 \end{cases} \tag{3.13}$$

Similar to the way how the constraint functions define the admissible set, the invariance functions define the invariant set.

Definition 3.2. The ***invariant set*** contains the states for which all invariance functions take a non-positive value and is defined as

$$\mathcal{G}(\boldsymbol{x}_\eta, \boldsymbol{\gamma}) = \{\boldsymbol{x} \,|\, \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) \leq 0 \quad \forall\, i \in \mathcal{B}\} \tag{3.14}$$

with $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_l]^\intercal \preccurlyeq \boldsymbol{0}_{l\times 1}$.

The invariant set describes the set of states, for which no constraint is violated and for which no constraint will be violated in the future if the pseudo input $z_i = \gamma_i$ is applied whenever an invariance function reaches zero. In addition, it exhibits a useful property for achieving constraint satisfaction.

Lemma 3.1. *The invariant set $\mathcal{G}(\boldsymbol{x}_\eta, \boldsymbol{\gamma})$ according to Def. 3.2 is a subset of the admissible set $\mathcal{H}(\boldsymbol{\eta})$ according to Def. 2.2, i.e.*

$$\mathcal{G}(\boldsymbol{x}_\eta, \boldsymbol{\gamma}) \subseteq \mathcal{H}(\boldsymbol{\eta}) \ .$$

The proof is provided in the appendix. Note that for a constraint with relative degree one, the admissible and the invariant set are equal as the invariance function is equal to the output function as derived in Example 3.2. For constraints with a higher relative degree, e.g. position constraints on torque-controlled manipulators, consider the following illustrating example.

Example 3.4. Let the system be a double integrator $\ddot{y} = u$ with the states y, $\dot{y}$ and the input u, which is constrained to $y \leq 0$. I/O-linearization yields a relative degree $r = 2$ and the input transformation $u = z$. Fig. 3.1 depicts the constraint and the corresponding admissible set composed of the entire left half-plane.

First, we consider $y > 0$, i.e. the right half-plane. Here, the constraint is violated due to the positive value of the output and corrective action is required, which is illustrated by the the states being within the inadmissible set indicated by the red coloring.

For $\dot{y} < 0$ and $y \leq 0$, i.e. the bottom left quadrant, the negative value of $\dot{y}$ causes the

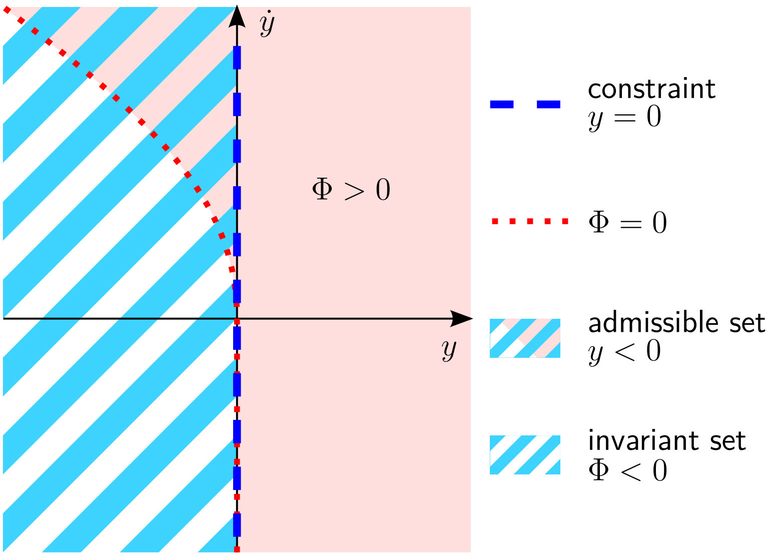


Figure 3.1.: Illustration of admissible and invariant set for $r = 2$.

output y to decrease. This means that while the states remain within the bottom left quadrant, no violation of the constraint occurs, independently from the current input. No corrective action is necessary and the entire quadrant belongs to the admissible set as well as to the invariant set.

For $\dot{y} \geq 0$ and $y \leq 0$, consider the example $\dot{y} > 0$ and $y = 0$, for which the positive derivative will increase the output value y independently from the input, thus causing an immediate violation. Similarly, there are more values of y, $\dot{y}$, for which no corrective action exists which keeps the state within the admissible set. This is illustrated by the fact that the corresponding part of the upper left quadrant is solely within the admissible set but not within the invariant set, whereas the remainder is in both sets.

The invariance function from (3.13) divides the two parts. The values left of the invariance functions are the state values within the invariant set, for which the negative corrective pseudo input γ just suffices to reduce $\dot{y}$ to zero when the output reaches $y = 0$.

The example illustrates that the invariant set is a proper subset of the admissible set for a relative degree of two. Similar relations hold for higher relative degrees, i.e. dynamics of higher order.

3.1.4 Corrective Control for the Input/Output-linearized System

Since the invariant set is a subset of the admissible set by Lemma 3.1, the goal of constraint satisfaction is achieved if the control law is designed to keep the state within the invariant set. This motivates the introduction of the notion of positive invariance.

Definition 3.3. A set $\mathcal{G}$ is ***positively invariant*** from time t onwards if it fulfills

$$\boldsymbol{x}(t) \in \mathcal{G} \;\Rightarrow\; \boldsymbol{x}(t + \Delta t) \in \mathcal{G} \;\forall\; \Delta t > 0$$

Therefore, if the corrective control input renders the invariant set positively invariant, constraint satisfaction is guaranteed as soon as the invariant set is entered the first time.

In order to find such a control input, we first investigate the linearized system, i.e. the integrator chains resulting from the I/O-linearization. In the absence of constraints, the system is controlled by nominal control $\boldsymbol{u}_{\mathrm{no}}$. The corresponding input for the linearized system is derived using (3.1). Hence the nominal pseudo input is given by

$$z_{\mathrm{no},i}(\boldsymbol{x}, \boldsymbol{x}_\eta) = \boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}(\boldsymbol{x}, \boldsymbol{\eta})\boldsymbol{\tau}_{\mathrm{no}} + b_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{e}_{\boldsymbol{u}}) \ . \tag{3.15}$$

A corrective pseudo input should on the one hand pass the nominal pseudo input whenever possible to ensure task execution and on the other hand initiate corrective action if the state is about to violate a constraint, i.e. if the invariance function is non-negative. The switching design of the corrective pseudo input

$$z_{\mathrm{c},i} = \begin{cases} \gamma_i & \text{if } t \in (\mathcal{N}_{a,i}(\gamma_i) \cup \mathcal{N}_{b,i}(\gamma_i)) \\ 0 & \text{if } t \in (\mathcal{N}_{c,i}(\gamma_i) \cup \mathcal{N}_{d,i}(\gamma_i)) \\ z_{\mathrm{no},i} & \text{else.} \end{cases} \tag{3.16}$$

with $\gamma_i < 0$ and the sets

$$\mathcal{N}_{a,i}(\gamma_i) = \{t \mid \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) > 0 \land \lim_{\Delta t \to \infty} (p_i(\boldsymbol{x}, \boldsymbol{x}_\eta, 0, \Delta t)) \geq 0\} \tag{3.17}$$

$$\mathcal{N}_{b,i}(\gamma_i) = \{t \mid \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) = 0 \land \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, 0) > 0\} \tag{3.18}$$

$$\mathcal{N}_{c,i}(\gamma_i) = \{t \mid \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) = 0 \land \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, 0) \leq 0\} \tag{3.19}$$

$$\mathcal{N}_{d,i}(\gamma_i) = \{t \mid \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) > 0 \land \lim_{\Delta t \to \infty} (p_i(\boldsymbol{x}, \boldsymbol{x}_\eta, 0, \Delta t)) < 0\} \tag{3.20}$$

fulfills the requirements and with $\gamma_i < 0$, the corrective pseudo input achieves a motion towards the invariant set for $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) > 0$ as the states of the integrator chain eventually decrease for a negative input.

Example 3.5 (Corrective pseudo input for $r_i = 1$ and $r_i = 2$)**.** Using (3.16)–(3.20), it is possible to derive the corrective pseudo inputs for constraints with relative degree one or two

$$r_i = 1 : z_{\mathrm{c},i} = \begin{cases} \gamma_i & \text{if } t \in \mathcal{N}_{1a,i}(\gamma_i) \\ 0 & \text{if } t \in \mathcal{N}_{1c,i}(\gamma_i) \\ z_{\mathrm{no},i} & \text{else.} \end{cases} \tag{3.21}$$

$$r_i = 2 : z_{\mathrm{c},i} = \begin{cases} \gamma_i & \text{if } t \in (\mathcal{N}_{2a,i}(\gamma_i) \cup \mathcal{N}_{2b,i}(\gamma_i)) \\ 0 & \text{if } t \in (\mathcal{N}_{2c,i}(\gamma_i) \cup \mathcal{N}_{2d,i}(\gamma_i)) \\ z_{\mathrm{no},i} & \text{else.} \end{cases} \tag{3.22}$$

with $\gamma_i < 0$ and the sets

$$\mathcal{N}_{1a,i}(\gamma_i) = \{t \,|\, h_i(\boldsymbol{x}, \boldsymbol{\eta}) > 0\} \tag{3.23}$$
$$\mathcal{N}_{1c,i}(\gamma_i) = \{t \,|\, h_i(\boldsymbol{x}, \boldsymbol{\eta}) = 0\} \tag{3.24}$$
$$\mathcal{N}_{2a,i}(\gamma_i) = \{t \,|\, h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta}) > \frac{\dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta)^2}{2\gamma_i} \wedge \dot{h}_i(\boldsymbol{x}, \boldsymbol{x}_\eta) \geq 0\} \tag{3.25}$$
$$\mathcal{N}_{2b,i}(\gamma_i) = \{t \,|\, h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta}) = \frac{\dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta)^2}{2\gamma_i} \wedge \dot{h}_i(\boldsymbol{x}, \boldsymbol{x}_\eta) > 0\} \tag{3.26}$$
$$\mathcal{N}_{2c,i}(\gamma_i) = \{t \,|\, h_i(\boldsymbol{x}, \boldsymbol{\eta}) = 0 \wedge \dot{h}_i(\boldsymbol{x}, \boldsymbol{x}_\eta) = 0\} \tag{3.27}$$
$$\mathcal{N}_{2d,i}(\gamma_i) = \{t \,|\, h_i(\boldsymbol{x}, \boldsymbol{\eta}) > 0 \wedge \dot{h}_i(\boldsymbol{x}, \boldsymbol{x}_\eta) < 0\}\ . \tag{3.28}$$

Note that for $r_i = 1$, there is a reduction to two sets since $p_i(\boldsymbol{x}, \boldsymbol{x}_\eta, 0, \Delta t)$ is equal to the constraint function and therefore also equal to the invariance function. For $r_i = 2$, the limit is related to the derivative of the constraint function and is therefore replaced in the sets.

Using the previous results, it is possible to show that the proposed corrective pseudo inputs achieve invariance.

Lemma 3.2 (Invariance in the linearized system)**.** *Let an integrator chain with r_i states resulting from the I/O-linearization with respect to $y_{\mathrm{c},i}$* (3.1) *be given. Then any pseudo input $z_i \leq z_{\mathrm{c},i}$, with $z_{\mathrm{c},i}$ from* (3.16) *and $\gamma_i < 0$ renders the invariant set* (3.14) *positively invariant.*

The proof is provided in the appendix. Lemma 3.2 shows that the states of the linearized system remain within the invariant set. The behavior on approaching the bound is influenced by the parameter γ_i.

Remark 3.1. The magnitude of γ_i may be chosen arbitrarily. A large magnitude of γ_i reduces the time, during which corrective control is applied, since it is able to change the output value faster. As a result, it increases the size of the invariant set and the magnitude of the required corrective control input. Equivalently, a small magnitude of γ_i increases the duration of using corrective control and decreases the size of the invariant set and the magnitude of the control input.

Remark 3.2. Due to the switching input, invariance controlled systems are prone to chattering in a sampled data implementation. This issue is addressed in the following section.

The results from the linearized system are the basis for deriving a constraint enforcing control input for the nonlinear system.

3.1.5 Corrective Control

Before determining a corrective control input, it is necessary to identify the constraints, which require corrective action. Therefore, the notion of active constraints is introduced.

Definition 3.4. A constraint $i \in \mathcal{B}$ is ***active*** if its invariance function is non-negative, i.e. $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) \geq 0$.

Definition 3.5. The ***set of active constraints*** $\mathcal{B}_{\text{act}}$ consists of all active constraints.

$$\mathcal{B}_{\text{act}}(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\gamma}) = \{i \in \mathcal{B} \mid \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) \geq 0\} \tag{3.29}$$

An active constraint is already or about to be violated and therefore requires corrective action [SWB08]. For the constraints, which are not active, the application of nominal control suffices. Note that the definition of active constraints is in accordance with the corrective pseudo input (3.21)–(3.22), which passes the nominal pseudo input if the constraint is inactive. With the results from Lemma 3.2 and the I/O-linearization (3.1), we receive conditions on the control input

$$z_i = \boldsymbol{a}_{\text{c},i}^{\mathsf{T}}(\boldsymbol{x}, \boldsymbol{\eta})\boldsymbol{u} + b_{\text{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{e}_{\boldsymbol{u}}) \leq z_{\text{c},i} \quad \forall\ i \in \mathcal{B}_{\text{act}}\ . \tag{3.30}$$

In order to improve readability, the explicit dependency on $\boldsymbol{x}$, $\boldsymbol{x}_\eta$ and $\boldsymbol{e}_{\boldsymbol{u}}$ is omitted for the vectors $\boldsymbol{a}_{\text{c},i}^{\mathsf{T}}(\boldsymbol{x}, \boldsymbol{\eta})$ and $b_{\text{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{e}_{\boldsymbol{u}})$ in the following.

Using the conditions (3.30), a corrective control input may be derived by solving the constrained minimization

$$\begin{aligned} \boldsymbol{u}_{\text{c}} &= \underset{\boldsymbol{u}}{\operatorname{argmin}}\ C\left(\boldsymbol{u}, \boldsymbol{u}_{\text{no}}\right) \\ \text{s.t.}\ & \boldsymbol{a}_{\text{c},i}^{\mathsf{T}}\boldsymbol{u} + b_{\text{c},i} \leq z_{\text{c},i} \quad \forall\ i \in \mathcal{B}_{\text{act}} \end{aligned} \tag{3.31}$$

where

- $C\left(\boldsymbol{u}, \boldsymbol{u}_{\text{no}}\right) : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is strictly convex in $\boldsymbol{u}$ and
- the minimum of the unconstrained minimization is $\boldsymbol{u}_{\text{c}} = \boldsymbol{u}_{\text{no}}$, i.e.

$$\underset{\boldsymbol{u}}{\operatorname{argmin}}\ C\left(\boldsymbol{u}, \boldsymbol{u}_{\text{no}}\right) = \boldsymbol{u}_{\text{no}}\ .$$

Note that this optimization generates a switching control law as the solution may change abruptly whenever a constraint becomes active or inactive. Furthermore, the formulation provides a useful property for the control derivation.

Lemma 3.3. *If the cost function $C\left(\boldsymbol{u}, \boldsymbol{u}_{no}\right)$ is strictly convex in $\boldsymbol{u}$, the minimization* (3.31) *is strictly convex and any local minimum is the unique global minimum.*

The proof is provided in the appendix. Due to the convexity, the optimization may be solved in a computationally efficient manner [Boy04, p.8], which is ideal for control in real-time with high sampling rates. In addition, the fact that any local minimum is the unique global minimum ensures in combination with the second condition on $C\left(\boldsymbol{u}, \boldsymbol{u}_{\text{no}}\right)$ that nominal control $\boldsymbol{u}_{\text{no}}$ is applied if it fulfills the optimization conditions, since it is a minimum.

The most commonly used cost function is the Euclidean norm of the input difference

$$C\left(\boldsymbol{u}, \boldsymbol{u}_{\text{no}}\right) = \left\|\boldsymbol{u} - \boldsymbol{u}_{\text{no}}\right\|_2^2\ , \tag{3.32}$$

which turns the optimization into a quadratic program (QP). This cost function is strictly convex as its Hessian is the identity matrix, which is positive definite. Using this function, (3.31) minimizes the difference between the elements of the control inputs. Sometimes,

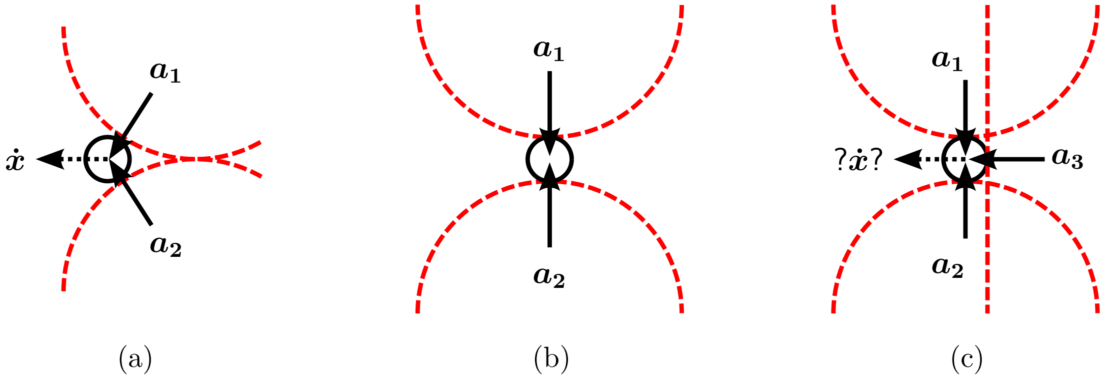


Figure 3.2.: Illustration of Assumption 3.1: In (a), an evasive motion $\dot{\boldsymbol{x}}$ may be derived as a linear combination of $\boldsymbol{a_1}, \boldsymbol{a_2}$, whereas in (b) this is not possible. In (c) an evasive motion is only defined if the constraint corresponding to $\boldsymbol{a_3}$ moves towards the state, otherwise the situation is similar to (b).

however, a more differentiated influence on the outcome of the constraint optimization is desired, e.g. to change how the effort is split between the actuators. In this case, an invertible weighting matrix $\boldsymbol{W} \in \mathbb{R}^{m \times m}$ is introduced in the cost function [SWB07]

$$C\left(\boldsymbol{u}, \boldsymbol{u}_{\mathrm{no}}\right) = \left\|\boldsymbol{W}^{-1}(\boldsymbol{u} - \boldsymbol{u}_{\mathrm{no}})\right\|_2^2 \ . \tag{3.33}$$

Note that in order to preserve the strict convexity of the cost function, $\boldsymbol{W}$ has to be defined such that $(\boldsymbol{W}\boldsymbol{W}^{\intercal})^{-1}$ is positive definite.

In order to check whether the minimization (3.31) has a solution, it is necessary to inspect the optimization conditions. The set over which the minimization is carried out determines whether a solution exists.

Definition 3.6. The ***set of admissible control inputs*** $\mathcal{M}$ is given by the intersection of the single optimization conditions.

$$\mathcal{M}(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\gamma}, \mathcal{B}_{\mathrm{act}}) = \{\boldsymbol{u} \mid \boldsymbol{a}_{\mathrm{c},i}^{\intercal}\boldsymbol{u} + b_{\mathrm{c},i} - z_{\mathrm{c},i} \leq 0 \quad \forall i \in \mathcal{B}_{\mathrm{act}}\} \tag{3.34}$$

If no constraints are active, the set $\mathcal{M}$ contains all possible input values. In this case, a solution exists and is given by $\boldsymbol{u}_{\mathrm{c}} = \boldsymbol{u}_{\mathrm{no}}$. Otherwise, a solution to the minimization problem only exists, if the set (3.34) is non-empty. Therefore, we make the following assumption.

Assumption 3.1. *The set of admissible control values $\mathcal{M}(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\gamma}, \mathcal{B}_{act})$, defined by the active constraints, is non-empty, $\mathcal{M}(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\gamma}, \mathcal{B}_{act}) \neq \emptyset$.*

Figure 3.2 illustrates the meaning of Assumption 3.1. This assumption seems a little restrictive, since obviously in Fig. 3.2b and 3.2c there exist admissible state trajectories. However, cases like in Fig. 3.2b and 3.2c rarely occur in reality. This would require an exact positioning of the limits with respect to the state such that the determined actions for the evasion of two or more constraints exactly oppose one another.

By analyzing the characteristics of $\mathcal{M}$ more closely, it is possible to determine an analytic solution to the minimization (3.31) with the cost function (3.32) or (3.33).

Analytic Solution

Before deriving the analytic solution, it is necessary to analyze how the set $\mathcal{M}$ influences the solution of the optimization. The minimization problem (3.31) has a unique global minimum by Lemma 3.3. Therefore, if $\boldsymbol{u}_{\mathrm{no}}$ is not contained within the set, the solution $\boldsymbol{u}_{\mathrm{c}}$ lies on the boundary of $\mathcal{M}$, i.e. some of the active constraints hold with equality

$$\mathcal{M}_{\mathrm{eq}}(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\gamma}, \mathcal{B}_{\mathrm{act}}) = \{i \in \mathcal{B}_{\mathrm{act}} \mid \boldsymbol{a}_i^\mathsf{T} \boldsymbol{u}_{\mathrm{c}} + b_{\mathrm{c},i} = z_{\mathrm{c},i}\} \tag{3.35}$$

and some with strict inequality

$$\mathcal{M}_{\mathrm{in}}(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\gamma}, \mathcal{B}_{\mathrm{act}}) = \{i \in \mathcal{B}_{\mathrm{act}} \mid \boldsymbol{a}_i^\mathsf{T} \boldsymbol{u}_{\mathrm{c}} + b_{\mathrm{c},i} < z_{\mathrm{c},i}\} = \mathcal{B}_{\mathrm{act}} \setminus \mathcal{M}_{\mathrm{eq}} \ . \tag{3.36}$$

The equality constraints in $\mathcal{M}_{\mathrm{eq}}$ define a sub-manifold of the input space, in which the solution lies. Not all of these constraints are necessarily required to define the sub-manifold. Instead, there exists a maximum subset of linearly independent constraints, which span the sub-manifold indicated by (3.35).

In the following, this maximum subset will be used to derive the analytic solution of (3.31) with the cost function (3.32) or (3.33). Therefore, let the subset

$$\mathcal{I} = \{i \in \mathcal{M}_{\mathrm{eq}} \mid \operatorname{rank}(\boldsymbol{A}_{\mathrm{c},\mathcal{I}}) = |\mathcal{I}| \wedge \operatorname{rank}(\boldsymbol{A}_{\mathrm{c},\mathcal{I}}) = \operatorname{rank}(\boldsymbol{A}_{\mathrm{c},\mathcal{M}_{\mathrm{eq}}})\} \tag{3.37}$$

with $\boldsymbol{A}_{\mathrm{c},\mathcal{I}} = [\boldsymbol{a}_{\mathrm{c},i}^\mathsf{T}]_{i \in \mathcal{I}}$ and $\boldsymbol{A}_{\mathrm{c},\mathcal{M}_{\mathrm{eq}}} = [\boldsymbol{a}_{\mathrm{c},i}^\mathsf{T}]_{i \in \mathcal{M}_{\mathrm{eq}}}$, describe such a maximum subset of the constraints in $\mathcal{M}_{\mathrm{eq}}$ with linearly independent $\boldsymbol{a}_{\mathrm{c},i}^\mathsf{T}$. It is in general not uniquely defined, but any such set spans the sub-manifold $\mathcal{M}_{\mathrm{eq}}$. The remaining constraints $\mathcal{M}_{\mathrm{eq}} \setminus \mathcal{I}$ are determined by a linear combination of the constraints in $\mathcal{I}$. As the constraints in $\mathcal{I}$ span a subspace in the input space and $\boldsymbol{u}_{\mathrm{c}} \in \mathbb{R}^m$, $|\mathcal{I}| \leq m$ holds. If $\mathcal{I}$ is empty, this means that either no constraints are active or all constraints hold with strict inequality, i.e. the nominal control signal is the minimum solution of (3.31). Hence, assuming a set $\mathcal{I}$ has been determined, the solution to the minimization problem (3.31) with the cost function (3.33) is determined similar to [SWB07] and given by

$$\boldsymbol{u}_{\mathrm{c}} = \begin{cases} \boldsymbol{u}_{\mathrm{no}} & \text{if } \mathcal{I} = \emptyset \\ \boldsymbol{W}(\boldsymbol{A}_{\mathrm{c},\mathcal{I}}\boldsymbol{W})^+(\boldsymbol{z}_{\mathrm{c},\mathcal{I}} - \boldsymbol{b}_{\mathrm{c},\mathcal{I}}) + (\boldsymbol{I} - \boldsymbol{W}(\boldsymbol{A}_{\mathrm{c},\mathcal{I}}\boldsymbol{W})^+\boldsymbol{A}_{\mathrm{c},\mathcal{I}})\boldsymbol{u}_{\mathrm{no}} & \text{else} \end{cases} \tag{3.38}$$

with the Moore-Penrose right pseudo inverse $(\boldsymbol{A}_{\mathrm{c},\mathcal{I}}\boldsymbol{W})^+$. The matrix and vectors are determined by $\boldsymbol{A}_{\mathrm{c},\mathcal{I}} = [\boldsymbol{a}_{\mathrm{c},i}^\mathsf{T}]_{i \in \mathcal{I}}$, $\boldsymbol{b}_{\mathrm{c},\mathcal{I}} = [b_{\mathrm{c},i}]_{i \in \mathcal{I}}$ and $\boldsymbol{z}_{\mathrm{c},\mathcal{I}} = [z_{\mathrm{c},i}]_{i \in \mathcal{I}}$.

The cost function (3.32) may be derived from (3.33) by setting the weight matrix $\boldsymbol{W}$ to the identity matrix $\boldsymbol{I}_m$. Therefore, the analytic solution to the minimization problem (3.31) with the cost function (3.32) is determined by substituting $\boldsymbol{W} = \boldsymbol{I}_m$ in (3.38). It remains to show whether the corrective control laws (3.31) and (3.38) are able to guarantee constraint satisfaction.

3.1.6 Control Properties

With corrective control being calculated using the minimization (3.31), it is necessary to determine whether the resulting input exhibits the desired control properties: constraint satisfaction and stability.

Invariance

Corrective control achieves constraint satisfaction if it is able to render the invariant set (3.14) positively invariant. Lemma 3.2 indicates the invariance of the linearized system under the corrective pseudo input $z_{\mathrm{c},i}$. However, as control is applied to the nonlinear system, it is necessary to investigate whether the control input, which is generated by the minimization (3.31), provides the required invariance.

Theorem 3.1. *Let the nonlinear system be given by* (2.1) *and the outputs describing the constraints by* (2.11)*. Let the linearizing input be determined by* (3.1) *and the elements of the corrective pseudo input by* (3.16)*. Let Assumptions 2.1, 2.2, 2.5, 2.6 and 3.1 hold. Then, if the system states are within the invariant set at some time $t = t_0$, i.e. $\boldsymbol{x}(t_0) \in \mathcal{G}(\boldsymbol{x}_\eta, \boldsymbol{\gamma})$, the corrective control input determined by* (3.31) *renders the invariant set* (3.14) *positively invariant for all $t \geq t_0$.*

The proof is provided in the appendix. Once the state of the nonlinear system enters the invariant set, Theorem 3.1 guarantees that it stays within this set for all future times as it is positive invariant. Since the invariant set is a subset of the admissible set, invariance control ensures constraint satisfaction. As the system does not necessarily start within the invariant set, it remains to show that the system will eventually enter the invariant set.

Theorem 3.2. *Let the nonlinear system be given by* (2.1) *and the outputs describing the constraints by* (2.11)*. Let the linearizing input be determined by* (3.1) *and the elements of the corrective pseudo input by* (3.21)–(3.22)*, depending on the relative degree. Let Assumptions 2.1, 2.2, 2.5, 2.6 and 3.1 hold. Then, if the system states lie outside of the invariant set at some instant of time $t = t_0$, i.e. $\boldsymbol{x}(t_0) \notin \mathcal{G}(\boldsymbol{x}_\eta, \boldsymbol{\gamma})$, the corrective control input determined by* (3.31) *guarantees that there exists a finite time interval T such that the system state enters and stays within the invariant set* (3.14) *for all $t \geq t_0 + T$.*

The proof is provided in the appendix. Theorem 3.2 shows that for invariance control, the initial state is not required to lie within the admissible set. Instead the control is such that after a finite time interval, the state becomes (and remains) admissible. By extension, this applies if additional constraints have to be included during runtime, e.g. when new obstacles appear.

Remark 3.3. The results on the invariance of the controlled system are independent from nominal control. This means that whichever nominal control scheme is chosen, whether it is learned or explicitly defined, stable or unstable, the use in combination with invariance control results in constraint satisfaction.

Nevertheless, the investigation of stability is essential for ensuring that the state does not grow unbounded and that the controlled system does not exhibit unexpected unstable behavior, which is especially important for safe interaction with humans.

Boundedness

Naturally, good task performance of a nonlinear system under invariance control is only possible if constraint enforcement does not result in unboundedness of the tracking error.

Theorem 3.3. *Let the nonlinear system be given by* (2.1) *and the outputs describing the constraints by* (2.11)*. Let the linearizing input be determined by* (3.1) *and the elements of the corrective pseudo input* $\boldsymbol{z}_c$ *by* (3.16)*. Let a sufficiently smooth and bounded desired motion be given by* $\boldsymbol{x}_{des} \in \mathbb{R}^n$ *and let the initial state values be within the invariant set* $\boldsymbol{x}(t_0) \in \mathcal{G}(\boldsymbol{x}_\eta, \boldsymbol{\gamma})$*. Let further Assumptions 2.1, 2.2, 2.4–2.6 and 3.1 hold and let* $V(\boldsymbol{e})$ *be a Lyapunov function showing global stability of the nominally controlled system. Then, if*

$$\mathcal{V} = \left\{ \boldsymbol{x} \,\middle|\, \exists\, t \geq t_0 \;:\; \operatorname{rank} \begin{bmatrix} \boldsymbol{A}_{c,\mathcal{B}_{act}}(\boldsymbol{x}, \boldsymbol{\eta}) \\ \dfrac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}} \boldsymbol{G}(\boldsymbol{x}) \end{bmatrix} = \operatorname{rank}(\boldsymbol{A}_{c,\mathcal{B}_{act}}(\boldsymbol{x}, \boldsymbol{\eta})) \right\}$$

with $\boldsymbol{A}_{c,\mathcal{B}_{act}} = [\boldsymbol{a}_{c,i}^\intercal]_{i \in \mathcal{B}_{act}}$ *is bounded, there exist constants* $\alpha, V_{\max} > 0$ *for which*

$$\begin{aligned} \boldsymbol{u}_c = \operatorname*{argmin}_{\boldsymbol{u}} & \left\| \boldsymbol{W}^{-1}(\boldsymbol{u} - \boldsymbol{u}_{no}) \right\|_2^2 \\ s.t.\;\; & \boldsymbol{a}_{c,i}^\intercal \boldsymbol{u} + b_{c,i} \leq z_{c,i} \quad \forall\; i \in \mathcal{B}_{act} \\ & \frac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}}(\dot{\boldsymbol{x}}_{des} - \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}) \leq B_{\dot{V}} \end{aligned} \tag{3.39}$$

$$\begin{aligned} \textit{with } B_{\dot{V}} &= \max(\alpha(V_{\max} - V(\boldsymbol{e})),\; \dot{V}_{no}(\boldsymbol{e}, \dot{\boldsymbol{e}}_{no}))\;, \\ \dot{V}_{no}(\boldsymbol{e}, \dot{\boldsymbol{e}}_{no}) &= \frac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}}(\dot{\boldsymbol{x}}_{des} - \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}_{no}) \end{aligned}$$

yields a uniquely defined corrective control input $\boldsymbol{u}_c$ *and a tracking error* $\boldsymbol{e} = \boldsymbol{x}_{des} - \boldsymbol{x}$ *that is at least bounded.*

The proof is provided in the appendix. Theorem 3.3 extends the minimization (3.31) by an additional condition to bound the tracking error within the isoline $V(\boldsymbol{e}) = \max(V_{\max}, V(\boldsymbol{e}(t_0)))$.

Remark 3.4. The goal of invariance control is to guarantee constraint enforcement also in cases when the desired trajectory leaves the admissible set, which is only possible if the tracking error is allowed to increase for such cases. Therefore it is natural that only boundedness of the tracking error is achieved.

Remark 3.5. Even if the initial state is not within the invariant set, the error is eventually bounded within $\max(V_{\max}, V(\boldsymbol{e}(t_{\text{inv}})))$, where t_{inv} is the instant of time when the state enters the admissible set, with $t_{\text{inv}} < \infty$ by Theorem 3.2.

Remark 3.6. In general, it is not trivial to determine whether the set $\mathcal{V}$ is bounded. If, however, the set of states itself only takes values from a bounded set $\mathcal{X} \subset \mathbb{R}^n$, i.e. $\boldsymbol{x} \in \mathcal{X}$, the set $\mathcal{V}$ is also bounded as $\mathcal{V} \subseteq \mathcal{X}$ holds. Since robotic systems are usually subject to joint and velocity limits, the set of states is bounded thus fulfilling the requirement of Theorem 3.3.

For a more intuitive understanding of Theorem 3.3 consider the following explanations and examples. Bounding the tracking error while satisfying all constraints is only possible if the interior of $V(\boldsymbol{e}) = V_{\max}$ contains constraint admissible states for all instances of the desired trajectory and the constraint parameters. In order to determine such a $V_{\max}$, the behavior of the state on active constraints is examined. If there are active constraints, the derivative of the tracking error may be divided into a tangential and normal component with respect to these constraints. While the normal component is dictated by the constraint avoidance and possibly increases $\dot{V}$, the tangential component may be chosen freely to achieve $\dot{V} < 0$. If however, the state is in a local minimum of the Lyapunov function on the constraints, i.e.

$$V(\boldsymbol{e}) = V_{\text{min}} = \min(V(\boldsymbol{e})|_{\boldsymbol{h}_{\text{c},\mathcal{B}_{\text{act}}}=\boldsymbol{0}_{|\mathcal{B}_{\text{act}}|\times 1}}) \ ,$$

further reduction of $\dot{V}$ via the tangential component is no longer possible. In that case the system needs to be able to follow the constraint motion for constraint satisfaction by allowing a positive $\dot{V}$, which is achieved by choosing V_{max} such that

$$V_{\text{max}} > \min(V(\boldsymbol{e})|_{\boldsymbol{h}_{\text{c},\mathcal{B}_{\text{act}}}=\boldsymbol{0}_{|\mathcal{B}_{\text{act}}|\times 1}})$$

is fulfilled for all these minima and all possible combinations of active constraints and bounded parameters. The parameter α is then determined by the maximum increase in the Lyapunov function that is required for constraint satisfaction in the local minima.

$$\alpha(V_{\text{max}} - V_{\text{min}})) > \max(\dot{V}|_{\boldsymbol{h}_{\text{c},\mathcal{B}_{\text{act}}}=\boldsymbol{0}_{|\mathcal{B}_{\text{act}}|\times 1}, V(\boldsymbol{e})=V_{\text{min}}}) \qquad \forall\ V_{\text{min}}$$

Remark 3.7. If the values of V_{max} and α are chosen too small, the optimization determining corrective control may be rendered infeasible. In view of the application, this issue is solved by increasing the values until a solution exists. However, on changing these values one needs to keep in mind that while α may be increased arbitrarily without deteriorating the task performance, an increase of V_{max} far beyond the minimally required value may lead to an increase in the bound on the tracking error.

The following examples illustrate the implications of Theorem 3.3.

Example 3.6. Consider the system of decoupled integrators $\dot{\boldsymbol{x}} = \boldsymbol{u}$ with the state $\boldsymbol{x} = [x_1,\ x_2]^\intercal$, the initial state $\boldsymbol{x}(t_0) = [11,\ -85]^\intercal$, the input $\boldsymbol{u} = [u_1,\ u_2]^\intercal$ and the nominal control law $\boldsymbol{u} = [-0.1x_1,\ x_1 - 0.1x_2]^\intercal$. The control law globally asymptotically stabilizes the system in the origin, which is validated using the Lyapunov function $V(\boldsymbol{x}) = \boldsymbol{x}^\intercal \boldsymbol{P} \boldsymbol{x}$ with the positive definite matrix

$$\boldsymbol{P} = \begin{bmatrix} 51 & 5 \\ 5 & 1 \end{bmatrix} \ .$$

The system is constrained by the output function $h_{\text{c}}(\boldsymbol{x}, \eta) = 9 + \eta - x_1$ with the bounded parameter $\eta = \cos(2\pi t)$. Derivation yields $r = 1$ and $z = \dot{\eta} + [-1,\ 0]\boldsymbol{u}$. As the Lyapunov function

$$V(\boldsymbol{x}) = 51x_1^2 + 10x_1x_2 + x_2^2$$

and the constraint are convex, there exists exactly one minimum on the constraint, which is determined by

$$\begin{aligned}
V(\boldsymbol{x})|_{h_\mathrm{c}=0} &= V(x_2) = 51(9+\eta)^2 + 10(9+\eta)x_2 + x_2^2 \quad \text{as } x_1 = 9+\eta \text{ for } h_\mathrm{c} = 0 \\
\frac{\partial V(\boldsymbol{x})|_{h_\mathrm{c}=0}}{\partial x_2} &= 10(9+\eta) + 2x_2 \\
\frac{\partial V(\boldsymbol{x})|_{h_\mathrm{c}=0}}{\partial x_2} &= 0 \quad \Rightarrow \quad x_2 = -5(9+\eta) \\
V_\mathrm{min} &= \min(V)|_{h_\mathrm{c}=0} = V(\boldsymbol{x})|_{\frac{\partial V}{\partial x2}=0} = 26(9+\eta)^2 \; .
\end{aligned}$$

This minimum Lyapunov function value reaches its maximum for $\eta = \max(\eta) = 1$.

$$\max(V_\mathrm{min}) = \min(V)|_{h_\mathrm{c}=0,\eta=1} = 2600$$

By choosing $V_\mathrm{max} > 2600$, it is assured that the isoline $V(\boldsymbol{x}) = V_\mathrm{max}$ always contains admissible states. If the state is in a minimum of the Lyapunov function V_min on the constraint, $u_1 = \dot{x}_1 = \dot{\eta}$ and $x_2 = -5x_1$ hold and the derivative

$$\dot{V} = \begin{bmatrix} 102x_1 + 10x_2 & 10x_1 + 2x_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

maximally increases with the rate

$$\dot{V}|_{h_\mathrm{c}=0,V_\mathrm{min}} = \begin{bmatrix} 52x_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\eta} \\ u_2 \end{bmatrix} = 52x_1\dot{\eta} \le 52 \cdot 10 \cdot 2\pi \approx 3268 \; .$$

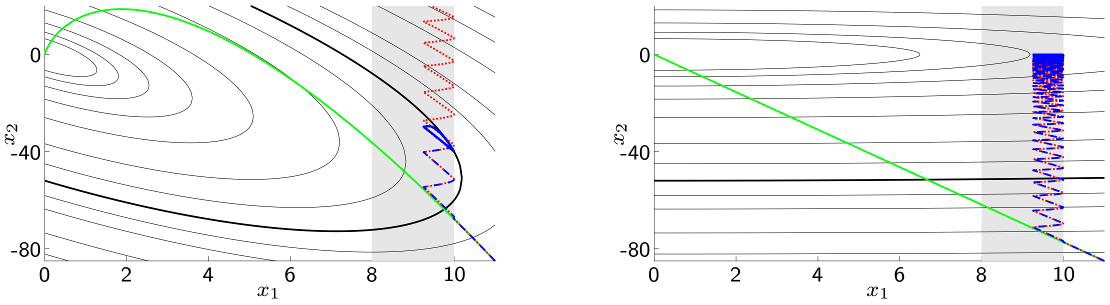
If V_max and α are then chosen to fulfill

$$\dot{V}|_{h_\mathrm{c}=0,V_\mathrm{min}} \le 3268 \le \alpha(V_\mathrm{max} - \max(V_\mathrm{min})) \; ,$$

e.g. $V_\mathrm{max} = 2700$ and $\alpha = 35$, the optimization from Theorem 3.3 is feasible and the tracking error is bounded. Fig. 3.3a depicts the state trajectories of the unconstrained system behavior in comparison to the constrained system controlled with corrective control from (3.31) with the cost function (3.32) and the approach introduced in Theorem 3.3. It is observed that while the unconstrained system approaches the origin asymptotically, both constrained trajectories remain at state values with $x_1 > 8$ due to the added constraint and the achieved invariance. The oscillation in the trajectories is due to the dynamic constraint. However, the state trajectory of the system controlled by (3.31) exhibits a growth in x_2, while the trajectory with control from Theorem 3.3 eventually ends up in a bounded limit cycle close to $V(\boldsymbol{x}) = V_\mathrm{max}$.

Example 3.7. Consider the same system and constraint as in the previous example but with nominal control $\boldsymbol{u} = [-0.1x_1, \; -0.1x_2]^\intercal$. The nominally controlled system is again globally asymptotically stable, which may be shown using the Lyapunov function

$$V(\boldsymbol{x}) = \boldsymbol{x}^\intercal \boldsymbol{x} \; .$$



(a)

(b)

Figure 3.3.: State trajectories of (a) Example 3.6 and (b) Example 3.7 for —— sole application of nominal control, ⋯⋯ corrective control from (3.31) and - - - - corrective control according to Theorem 3.3. The constraint moves within the ▒ shaded area and —— are the isolines of the Lyapunov function, where —— represents $V(\boldsymbol{x}) = V_{\max}$.

The same values of $V_{\max}$ and α as in the previous example are used. Fig. 3.3b depicts the state trajectories of the unconstrained system behavior in comparison to the constrained system controlled with corrective control from (3.31) with the cost function (3.32) and Theorem 3.3. Consistent with the previous example, the unconstrained system approaches the origin asymptotically but both constrained trajectories remain at state values with $x_1 > 8$. Here, however, the state trajectories of both constrained systems are equivalent and are both bounded. This is due to the fact that in this case, the projection of nominal control onto the constraint results in a behavior that does not increase the Lyapunov function thus automatically fulfilling the additional condition of Theorem 3.3.

Both examples show that the proposed invariance control approach successfully achieves the desired constraint satisfaction while keeping the tracking error bounded. It is, however, not yet clear, how the control method performs in sampled time implementations and whether the dynamic behavior at the bounds is suitable in applications that include physical interaction with humans. These questions are addressed in the following section.

3.2 Extensions of Invariance Control

The invariance control scheme for continuous time, developed in the previous section, guarantees constraint satisfaction by introducing a constraint admissible set that is rendered positively invariant. Due to the switching control input, however, the resulting state trajectory may not be as smooth as required for some applications. In physical human-robot interaction, for example, the human interaction partner may expect a certain dynamic behavior and smoothness of the interaction, which the switching invariance control is not able to provide. Introducing augmented invariance control [KJH16] allows designing the smoothness characteristics of the state to address the requirements of the respective application.

Furthermore, in real applications, control is usually executed in sampled time. The introduced constraint satisfaction guarantee does, however, not transfer directly to sampled

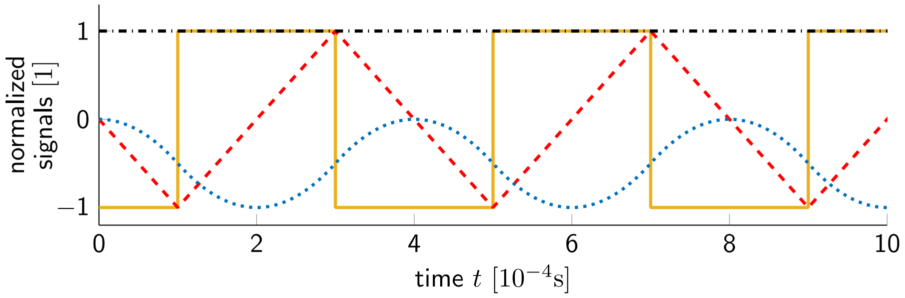


Figure 3.4.: Invariance controlled integrator chain $\dot{x}_1 = x_2$, $\dot{x}_2 = u$ with the constraint $x_1 \leq 0$ and the nominal control $u_{\text{no}} = -(x_1 - 1)$. Depicted are the normalized signals ······ $x_1/10^{-8}$, - - - - $x_2/10^{-4}$, —— $u/1$ and -·-·- $u_{\text{no}}/1$.

time implementations, as the switch to corrective control may be delayed, thus leading to constraint violations. This issue is resolved by the chattering reduction method from [KH14].

3.2.1 Augmented Invariance Control

Although the control law (3.31) guarantees constraint enforcement, the switching caused by constraints becoming active or inactive leads to input discontinuities. This becomes obvious in sampled time implementations, where the input switches from nominal to corrective control and vice versa every couple of time steps, when the state is at a constraint. Figure 3.4 shows an example for a constrained double integrator. The switching input exhibits discontinuities, stresses the system unnecessarily and causes a continuous but not continuously differentiable state $\boldsymbol{x}(t) \in \mathcal{C}^0$, which is evident from the inflection points in x_2. This is, however, unwanted in applications requiring a smoother state $\boldsymbol{x}(t) \in \mathcal{C}^v$, $v \in \mathbb{N}$ such as, for example, human-in-the-loop control design.

Smoothness Design by Augmentation

Whenever a certain dynamic behavior of the constraint enforcing input and the states is desired, standard invariance control is unsuitable due to the switching characteristics. Therefore, we introduce augmented invariance control [KJH16], which moves the switching to a higher order time derivative of the control input thus leading to a smoother state. The basic idea is to augment the system dynamics (2.1) by an additional dynamic structure adjusting the input dynamics. The desired control structure is illustrated in Fig. 3.5. The structure corresponds to the envisioned control structure in Fig. 2.1. In addition to system, nominal control and corrective invariance control, here, augmented dynamics are introduced in between invariance control and system, thus augmenting the system. If these dynamics fulfill certain requirements, this allows to design invariance control using the steps introduced in Section 3.1. However, the augmentation poses some challenges on the nominal control design since using the original nominal control as control input of the augmented system, i.e. system and augmentation, does in general not result in the desired nominal behavior. Therefore, the augmented version of nominal control has to be designed such that in the unconstrained case, the augmented system displays a similar behavior to the original dynamics without augmentation under nominal control. For this purpose an inverse augmentation is introduced in between nominal control and invariance control. Naturally, the design of

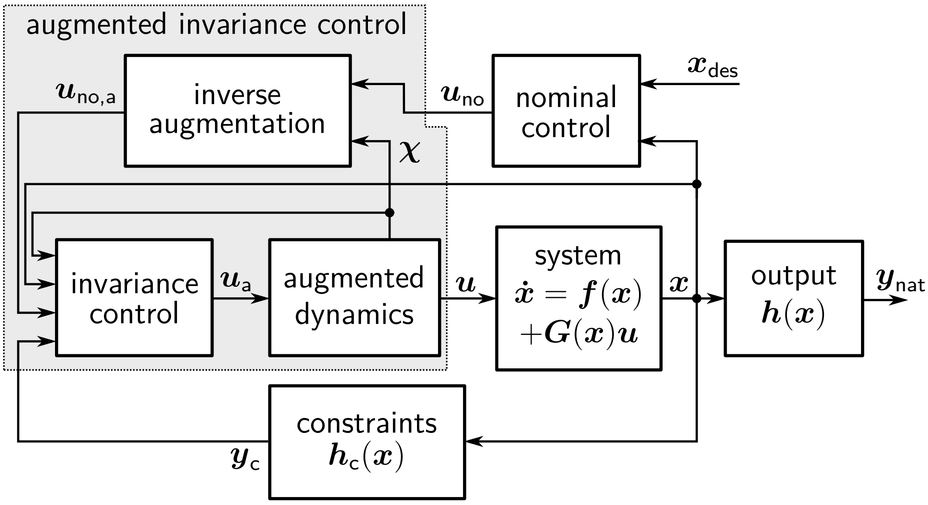


Figure 3.5.: Nonlinear dynamical system controlled by augmented invariance control, which combines nominal control with constraint enforcing control and desired input dynamics encoded in the augmentation.

nominal control depends on the augmented dynamics.

Note that for the sake of clarity, we assume the input error to equal zero in this section.

Assumption 3.2. *The input error $\boldsymbol{e}_u$ is equal to zero.*

$$\boldsymbol{e}_u = \boldsymbol{0}_{m\times 1}$$

Naturally, the following deductions may be applied in the presence of input errors. This would, however, require additional assumptions on the smoothness of the input error.

Augmented Dynamics

We define the augmentation in the following way.

Definition 3.7. The ***augmented dynamics*** are given by a nonlinear, control affine system

$$\begin{array}{rcl} \dot{\boldsymbol{\chi}} &=& \boldsymbol{f}_{\mathrm{a}}(\boldsymbol{\chi}) + \boldsymbol{G}_{\mathrm{a}}(\boldsymbol{\chi})\boldsymbol{u}_{\mathrm{a}} \\ \boldsymbol{u} &=& \boldsymbol{h}_{\mathrm{c,a}}(\boldsymbol{\chi}) \ , \end{array} \tag{3.40}$$

with the constant degree of augmentation $v \in \mathbb{N}$, with $\boldsymbol{\chi} \in \mathbb{R}^{m\cdot v}$ being the state, $\boldsymbol{u} \in \mathbb{R}^m$ being the control input of (2.1), $\boldsymbol{u}_{\mathrm{a}} \in \mathbb{R}^m$ being the input of the augmentation and the vector and matrix vector fields $\boldsymbol{f}_{\mathrm{a}} : \mathbb{R}^{m\cdot v} \to \mathbb{R}^{m\cdot v}$ and $\boldsymbol{G}_{\mathrm{a}} : \mathbb{R}^{m\cdot v} \to \mathbb{R}^{m\cdot v\times m}$ being sufficiently smooth to allow for I/O-linearization.

The goal of the augmentation is an increased smoothness of the state such that it fulfills $\boldsymbol{x} \in \mathcal{C}^v$ instead of $\boldsymbol{x} \in \mathcal{C}^0$. Therefore, we make the following assumption on the design of the augmented dynamics.

Assumption 3.3. *I/O-linearization of* (3.40) *with respect to each element of the output $u_{a,i}$, $1 \leq i \leq m$ and the input $\boldsymbol{u}_a$ yields a well-defined relative degree $v \in \mathbb{N}$, i.e.*

$$\begin{aligned}\det(\mathscr{L}_{\boldsymbol{\chi}\boldsymbol{G}_a}\mathscr{L}_{\boldsymbol{\chi}\boldsymbol{f}_a}^{v-1}h_{c,a,i}(\boldsymbol{\chi})) \neq 0 \ \ \forall 1 \leq i \leq m \ \textit{and} \\ \mathscr{L}_{\boldsymbol{\chi}\boldsymbol{G}_a}\mathscr{L}_{\boldsymbol{\chi}\boldsymbol{f}_a}^{k}\boldsymbol{h}_{c,a}(\boldsymbol{\chi}) = \boldsymbol{0}_m \ \ \textit{for}\ 0 \leq k < v-1 \ .\end{aligned}$$

This assumption imposes little restriction on the augmented dynamics as they may be chosen freely. As the I/O-linearization of (3.40) yields the input transformation

$$\begin{aligned}\boldsymbol{u}^{(v)} &= \mathscr{L}_{\boldsymbol{\chi}\boldsymbol{G}_\mathrm{a}}\mathscr{L}_{\boldsymbol{\chi}\boldsymbol{f}_\mathrm{a}}^{v-1}\boldsymbol{h}_{\mathrm{c,a}}(\boldsymbol{\chi})\boldsymbol{u}_\mathrm{a} + \mathscr{L}_{\boldsymbol{\chi}\boldsymbol{f}_\mathrm{a}}^{v}\boldsymbol{h}_{\mathrm{c,a}}(\boldsymbol{\chi}) \\ &= \boldsymbol{A}_\mathrm{a}(\boldsymbol{\chi})\boldsymbol{u}_\mathrm{a} + \boldsymbol{b}_\mathrm{a}(\boldsymbol{\chi}) \ . \end{aligned} \tag{3.41}$$

with $\boldsymbol{A}_\mathrm{a}(\boldsymbol{\chi}) \in \mathbb{R}^{m\times m}$ being non-singular due to Assumption 3.3, the augmentation has a well-defined vector relative degree $[v \ \dots \ v]^\intercal$ according to Def. 2.3 and the augmented dynamics are fully linearized by the I/O-linearization.

Nevertheless, it is also possible to choose different dynamics (3.40) with more than mv states and outputs with relative degrees greater then v that do not fulfill Assumption 3.3. In this case, the total relative degree is smaller than the number of states, which means that the dynamics introduce additional internal dynamics which may affect the stability of the controlled system. In contrast, by choosing (3.40) according to Assumption 3.3, the total relative degree according to Def. 2.4 is

$$\sum_{i=1}^{m} v = mv \ ,$$

which avoids introducing internal dynamics.

Example 3.8 (Augmentation by a single integrator)**.** Augmentation by $v = 1$ is achieved by a single integrator in each element of the input

$$\dot{\boldsymbol{\chi}} = \boldsymbol{u}_\mathrm{a} \ , \quad \boldsymbol{u} = \boldsymbol{\chi} \ . \tag{3.42}$$

The relative degree of each element is $v = 1$. The vector relative degree $[1 \ \dots \ 1]^\intercal$ yields the total relative degree m, which is equal to the number of states in (3.42) and the I/O-linearization results in $\boldsymbol{A}_\mathrm{a} = \boldsymbol{I}_m$ and $\boldsymbol{b}_\mathrm{a} = \boldsymbol{0}_{m\times 1}$.

Combining (2.1) with (3.40) defines the augmented system.

Definition 3.8. The ***augmented system*** is given by

$$\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}) + \tilde{\boldsymbol{G}}(\tilde{\boldsymbol{x}})\boldsymbol{u}_\mathrm{a} \tag{3.43}$$

$$\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}) = \begin{bmatrix}\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{h}_{\mathrm{c,a}}(\boldsymbol{\chi}) \\ \boldsymbol{f}_\mathrm{a}(\boldsymbol{\chi})\end{bmatrix} \ , \ \tilde{\boldsymbol{G}}(\tilde{\boldsymbol{x}}) = \begin{bmatrix}\boldsymbol{0}_n \\ \boldsymbol{G}_\mathrm{a}(\boldsymbol{\chi})\end{bmatrix} \tag{3.44}$$

with the augmented state $\tilde{\boldsymbol{x}} = [\boldsymbol{x}^\intercal, \boldsymbol{\chi}^\intercal]^\intercal \in \mathbb{R}^{n+m\cdot v}$.

Augmented Corrective Control

Based on the augmented system (3.43)–(3.44) and the constraint functions (2.11), invariance control determines a constraint enforcing input $\boldsymbol{u}_\mathrm{a}$ for the augmented system using the methods from Sec. 3.1. As before, this requires I/O-linearization with respect to each constraint function. As the goal is to increase the smoothness of the state, which is achieved by increasing the relative degree, additional smoothness assumptions on the constraint parameters extending Assumption 2.2 are required.

Assumption 3.4. *The constraint parameters $\boldsymbol{\eta}$ are such that for a constant $v \in \mathbb{N}$*

(i) each η_j, $1 \le j \le n_\eta$ is $r_\mathrm{max} + v + 1$ times differentiable w.r.t. time and $\eta_j \in \mathcal{C}^{r_\mathrm{max}+v}$,

(ii) each η_j, $1 \le j \le n_\eta$ and its $r_\mathrm{max} + v$ derivatives are measurable and bounded,

The dynamics of the parameters are modeled by (2.13) with the state $\boldsymbol{x}_{\boldsymbol{\eta}} \in \mathbb{R}^{(r_\mathrm{max}+v+1)n_\eta}$. Note that if the conditions on the parameters do not hold naturally, the parameter variation has to be represented by a sufficiently smooth approximation. Then, the input transformation of the augmented system derived from I/O-linearization is determined by

$$\begin{aligned} y_{\mathrm{c},i}^{(r_{\mathrm{a},i})} &= \tilde{\boldsymbol{a}}_i^\intercal(\tilde{\boldsymbol{x}}, \boldsymbol{x}_{\boldsymbol{\eta}})\boldsymbol{u}_\mathrm{a} + \tilde{b}_i(\tilde{\boldsymbol{x}}, \boldsymbol{x}_{\boldsymbol{\eta}}) \\ \text{with } \tilde{\boldsymbol{a}}_i^\intercal(\tilde{\boldsymbol{x}}, \boldsymbol{x}_{\boldsymbol{\eta}}) &= \mathscr{L}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{G}}}\mathscr{L}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{f}}}^{r_{\mathrm{a},i}-1} y_{\mathrm{c},i} \ , \ \tilde{b}_i(\tilde{\boldsymbol{x}}, \boldsymbol{x}_{\boldsymbol{\eta}}) = \mathscr{L}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{f}}}^{r_{\mathrm{a},i}} y_{\mathrm{c},i} \end{aligned} \tag{3.45}$$

with the augmented relative degree $r_{\mathrm{a},i} \in \mathbb{N}$.

Definition 3.9. The ***augmented relative degree*** $r_{\mathrm{a},i}$ is the relative degree resulting from the I/O-linearization of each constraint output $i \in \mathcal{B}$ with respect to the the input $\boldsymbol{u}_\mathrm{a}$ of the augmented system.

The augmented relative degree depends on the original relative degree r_i and on the degree of augmentation v.

Lemma 3.4. *Let Assumptions 2.2, 3.2 and 3.3 hold. Then the I/O-linearization of the augmented system* (3.43)–(3.44) *with respect to each constraint function $y_{\mathrm{c},i} = h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta})$ from* (2.11) *and the control input $\boldsymbol{u}_a$ yields $\tilde{\boldsymbol{a}}_i^\intercal(\tilde{\boldsymbol{x}}, \boldsymbol{x}_{\boldsymbol{\eta}}) = \boldsymbol{a}_{c,i}^\intercal \boldsymbol{A}_a$ and has a well-defined augmented relative degree $r_{a,i} = r_i + v$.*

The proof is provided in the appendix. The corresponding invariance functions are then defined by (3.10) using the relative degree $r_{\mathrm{a},i} = r_i + v$. They determine the invariant set (3.14) and the set of active constraints (3.29). Constraint enforcing control is then derived from the minimization problem

$$\boxed{\begin{aligned} \boldsymbol{u}_{\mathrm{c,a}} &= \underset{\boldsymbol{u}_\mathrm{a}}{\operatorname{argmin}}\ C\left(\boldsymbol{u}_\mathrm{a}, \boldsymbol{u}_{\mathrm{no,a}}\right) \\ \text{s.t. } & \boldsymbol{a}_{\mathrm{c},i}^\intercal \boldsymbol{A}_\mathrm{a} \boldsymbol{u}_\mathrm{a} + \tilde{b}_i \le z_{\mathrm{c,a},i} \quad \forall i \in \mathcal{B}_\mathrm{act} \end{aligned}} \tag{3.46}$$

where $\boldsymbol{u}_{\mathrm{c,a}} \in \mathbb{R}^m$ is the corrective input of the augmented system, $\boldsymbol{u}_{\mathrm{no,a}} \in \mathbb{R}^m$ represents nominal control for the augmented system, the cost function $C\left(\boldsymbol{u}_\mathrm{a}, \boldsymbol{u}_{\mathrm{no,a}}\right)$ fulfills the same conditions as in (3.31) and $z_{\mathrm{c,a},i}$ is determined according to (3.16) with the nominal pseudo input $z_{\mathrm{no,a},i} = \boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}} \boldsymbol{A}_{\mathrm{a}} \boldsymbol{u}_{\mathrm{no,a}} + \tilde{b}_i$ instead of $z_{\mathrm{no},i}$. As (3.46) represents invariance control being applied to enforce constraints with relative degree $r_i + v$, Theorems 3.1 and 3.2 apply which means that constraint satisfaction is guaranteed. Note that if guaranteed boundedness of the tracking error is required as well, the additional boundedness condition has to be included in the control derivation (3.46) in accordance with Theorem 3.3.

This leaves the question of how the augmented nominal control $\boldsymbol{u}_{\mathrm{no,a}}$ has to be designed to ensure that the augmented system is able to reach the nominal control goal if no constraints are present.

Augmented Nominal Control

Using $\boldsymbol{u}_{\mathrm{no,a}} = \boldsymbol{u}_{\mathrm{no}}$ does, in general, not result in the desired nominal behavior due to the added dynamics that are not considered in the design of $\boldsymbol{u}_{\mathrm{no}}$. In addition, the augmentation prevents an instantaneous switch to nominal control. Therefore, an inverse augmentation needs to be added to the control loop as depicted in Fig. 3.5 in order to compensate for the augmentation. We propose designing a feedback control law for the nominal control input such that if no constraints are active, the input $\boldsymbol{u}$ approaches original nominal control $\boldsymbol{u}_{\mathrm{no}}$ for $t \to \infty$, which is achieved by the control design

$$\boldsymbol{u}_{\mathrm{no,a}} = \boldsymbol{A}_{\mathrm{a}}^{-1}(\boldsymbol{\chi}) \left(\boldsymbol{u}_{\mathrm{no}}^{(v)} - \boldsymbol{b}_{\mathrm{a}}(\boldsymbol{\chi}) - \sum_{j=0}^{v-1} k_j \boldsymbol{e}_{\mathrm{no}}^{(j)} \right) \tag{3.47}$$

with $\boldsymbol{e}_{\mathrm{no}}^{(j)} = \boldsymbol{u}^{(j)} - \boldsymbol{u}_{\mathrm{no}}^{(j)}$, $\boldsymbol{u}^{(j)} = \mathscr{L}_{\boldsymbol{\chi} \boldsymbol{f}_{\mathrm{a}}}^{j} \boldsymbol{h}_{\mathrm{c,a}}(\boldsymbol{\chi})$, the design parameters $k_j \in \mathbb{R}$ and the inverse $\boldsymbol{A}_{\mathrm{a}}^{-1}(\boldsymbol{\chi})$, which exists if Assumption 3.3 holds. The control law (3.47) requires $\boldsymbol{u}_{\mathrm{no}}$ to be v times differentiable.

Assumption 3.5. *The nominal control input $\boldsymbol{u}_{no}$ is v times continuously differentiable, i.e. $\boldsymbol{u}_{no} \in \mathcal{C}^v$, where v is the degree of augmentation.*

As $\boldsymbol{u}_{\mathrm{no}}$ is a function of the desired trajectory $\boldsymbol{x}_{\mathrm{des}}(t)$ and $\boldsymbol{x}$, which is at least v times continuously differentiable due to the augmentation, Assumption 3.5 is equal to the condition $\boldsymbol{x}_{\mathrm{des}}(t) \in \mathcal{C}^v$. If $\boldsymbol{x}_{\mathrm{des}}(t) \notin \mathcal{C}^v$, i.e. $\boldsymbol{u}_{\mathrm{no}} \notin \mathcal{C}^v$, a different control law has to replace (3.47). In that case, however, it is questionable whether it is sensible to introduce the augmentation at all, as then the state might never be able to sufficiently track the desired trajectory due to difference in smoothness.

Whether the control law (3.47) is able to achieve the desired nominal behavior depends on the choice of the design parameters $k_j \in \mathbb{R}$.

Lemma 3.5. *Let* (3.40) *be controlled by $\boldsymbol{u}_a = \boldsymbol{u}_{no,a}$ from* (3.47). *Let Assumptions 3.3 and 3.5 hold. Then, if*

$$d(s) = s^v + \sum_{j=0}^{v-1} k_j s^j$$

is Hurwitz, the control error $\boldsymbol{e}_{no} = \boldsymbol{u} - \boldsymbol{u}_{no}$ is uniformly exponentially stable with respect to $\boldsymbol{e}_{no} = \boldsymbol{0}_{m \times 1}$.

The proof is provided in the appendix. As k_j, $j \in \{1, 2, ..., v-1\}$ are design parameters, they may be chosen arbitrarily to satisfy the Routh-Hurwitz criterion [Hur95]. Using these results, the characteristics of the augmented system (3.43)–(3.44) under nominal control are shown.

Theorem 3.4. *Let Assumptions 2.1,3.2, 3.3 and 3.5 hold. Then, if* $\boldsymbol{u}_{no}$ *stabilizes the tracking error* $\boldsymbol{e}_w = \boldsymbol{x}_{des}(t) - \boldsymbol{x}(t)$ *of* (2.1) *uniformly exponentially in* $\boldsymbol{e}_w = \boldsymbol{0}_{n\times 1}$*, the tracking error of the augmented system* (3.43)–(3.44) *controlled by* (3.47) *is uniformly asymptotically stabilized in* $\boldsymbol{e}_w = \boldsymbol{0}_{n\times 1}$.

The proof is provided in the appendix. Theorem 3.4 establishes the asymptotic stability of the nominally controlled system. As $\boldsymbol{u}_{\mathrm{no}}$ is reached asymptotically, other performance specifications are fulfilled asymptotically. Note that, compliance control [SK08, Ch. 7] does generally not fulfill the requirement of exponentially stable tracking. This issue is addressed in [JH17] where uniform asymptotic stability in the sense of Lyapunov is shown for augmented systems, which are nominally under impedance control.

Numerical Evaluation

In this section, the characteristics of augmented invariance control are illustrated using the model of an inverted pendulum with a velocity constraint.

Simulation Setup We consider the nonlinear control affine single-input single-output (SISO) system

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u \\
\boldsymbol{f}(\boldsymbol{x}) &= \begin{bmatrix} x_2 \\ \frac{g}{l}\sin(x_1) - \frac{m\cos(x_1)\sin(x_1)(lx_2^2+g\cos(x_1))}{l(m\sin^2(x_1)+M)} \end{bmatrix} \\
\boldsymbol{g}(\boldsymbol{x}) &= \begin{bmatrix} 0 & \dfrac{\cos(x_1)}{l(m\sin^2(x_1)+M)} \end{bmatrix}^{\mathsf{T}} , \\
y_{\mathrm{c}} &= x_2 - \dot{\theta}_{\max}
\end{aligned}$$

with the angle to the upright position x_1, the angular velocity x_2, the length l, the gravity constant g, the masses of the pendulum and the cart m and M and a constant velocity bound $\dot{\theta}_{\max}$. The ***nominal control*** goal of keeping the pendulum upright is achieved by the PD control law

$$u_{\mathrm{no}} = -k_P x_1 - k_D x_2$$

with the proportional and differential gains $k_P \in \mathbb{R}^+$ and $k_D \in \mathbb{R}^+$. In order to achieve a continuous control input, the augmentation by a single integrator from Example 3.8 is used. As the above control law is continuously differentiable, ***augmented nominal control*** is given by

$$u_{\mathrm{no,a}} = \dot{u}_{\mathrm{no}} - k_0(u - u_{\mathrm{no}})$$

according to (3.47) with $k_0 \in \mathbb{R}^+$ to fulfill the stability criterion from Lemma 3.5. Derivation of the constraint function y_{c} yields a well-defined ***augmented relative degree*** $r_a = 2$ for states fulfilling $x_1 \neq \pm\frac{\pi}{2} + k$, $k \in \mathbb{Z}$ and

$$\begin{aligned}
\tilde{a} &= \frac{\cos(x_1)}{l(m\sin^2(x_1)+M)} , \\
\tilde{b} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{g}{l}\sin(x_1) - \frac{m\cos(x_1)\sin(x_1)(lx_2^2+g\cos(x_1))}{l(m\sin^2(x_1)+M)}\right) + \left(\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\cos(x_1)}{l(m\sin^2(x_1)+M)}\right)\right) u .
\end{aligned}$$

Table 3.1.: Model parameters for the numerical evaluation of augmented invariance control.

Sampling time	T_A	$1 \cdot 10^{-5}\,\mathrm{s}$
System parameters	$\boldsymbol{x}(0)$	$[-5\frac{\pi}{180}\,\mathrm{rad}\ ,\ 0\,\mathrm{rad/s}]^\intercal$
	l; g	$0.2\,\mathrm{m}$; $9.81\,\mathrm{m/s}^2$
	m; M	$0.1\,\mathrm{kg}$; $1\,\mathrm{kg}$
Constraint	$\dot{\theta}_{\mathrm{max},1}$	$1\frac{\pi}{180}\,\mathrm{rad/s}$
	$\dot{\theta}_{\mathrm{max},2}$	$1.5\frac{\pi}{180}\,\mathrm{rad/s}$
Nominal control	k_P; k_D	$10\,\mathrm{N}$; $1\,\mathrm{N\,s}$
Invariance from Sec. 3.1	γ	-0.1 1/s
Augmented invariance	γ	-10 1/s^2
	k_0	1 or 5

The invariance function is defined according to (3.13)

$$\Phi(\tilde{\boldsymbol{x}}, \gamma) = \begin{cases} y_{\mathrm{c}} & \text{for } \dot{y}_{\mathrm{c}} \leq 0 \\ -\dfrac{\dot{y}_{\mathrm{c}}^2}{2\gamma} + y_{\mathrm{c}} & \text{else} \end{cases}$$

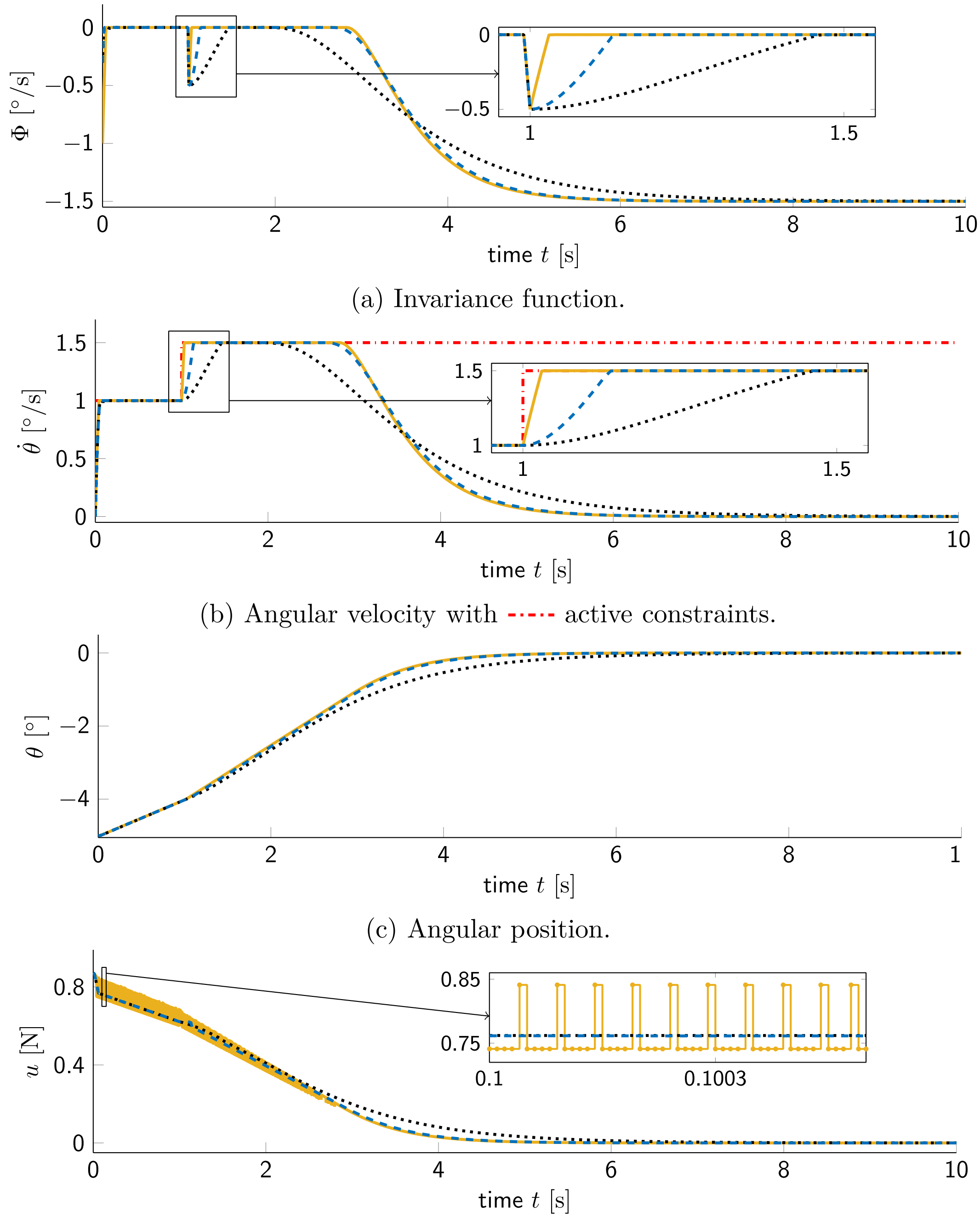
and corrective control is then determined by

$$u_{\mathrm{c,a}} = \begin{cases} u_{\mathrm{no,a}} & \text{for } \Phi(\tilde{\boldsymbol{x}}, \gamma) < 0 \\ \dfrac{z_{\mathrm{c}} - \tilde{b}}{\tilde{a}} & \text{for } \Phi(\tilde{\boldsymbol{x}}, \gamma) \geq 0 \end{cases} \tag{3.48}$$

with z_{c} from (3.22).

Implementation The model is implemented in *Matlab/Simulink* using the parameters in Table 3.1. The forward Euler method determines the solution with a step size of $1 \cdot 10^{-5}\,\mathrm{s}$. The constraint parameter $\dot{\theta}_{\mathrm{max}}$ is switched from $\dot{\theta}_{\mathrm{max},1}$ to $\dot{\theta}_{\mathrm{max},2}$ at $t = 1\,\mathrm{s}$. Augmented invariance control with two different values of k_0 is compared to standard invariance control from Sec. 3.1.

Results The results generated by standard and augmented invariance control using $k_0 = 1$ and $k_0 = 5$ in the augmented nominal control law (3.47) are depicted in Fig. 3.6. Apart from some tiny chattering effects, the invariance function in Fig. 3.6a has a non-positive value and the angular velocity in Fig. 3.6b lies below the constraint for all times, thus showing that both the standard and the augmented invariance control enforce the constraints. In addition, all control schemes are able to stabilize the angular position, which approaches the desired value zero as depicted in Fig. 3.6c. The difference between the approaches is visible in the control input in Fig. 3.6d. Whenever a constraint is active, i.e. $\Phi(\boldsymbol{x}, \gamma) \geq 0$, the input u generated by invariance control according to Sec. 3.1 is fast and discontinuously switching between nominal and corrective control, whereas augmented invariance control provides a continuous input. Furthermore, whenever the constraint is inactive, i.e. $\Phi(\boldsymbol{x}, \gamma) < 0$, standard invariance



(a) Invariance function.

(b) Angular velocity with -·-·- active constraints.

(c) Angular position.

(d) Control input.

Figure 3.6.: Results for the pendulum with — standard invariance control and augmented invariance control for ⋯⋯ $k_0 = 1$ and - - - - $k_0 = 5$.

control immediately applies the nominal control input u_{no}, which is why this control scheme reaches the constraints and the goal angle fastest as depicted in Fig. 3.6a-3.6c. It may further be noted that with increasing values of k_0, the tracking of the augmented invariance control improves and approaches the performance of standard invariance control.

Concluding, it may be noted that augmented invariance control is successful in increasing the smoothness of the state while guaranteeing constraint satisfaction in continuous time implementations. It does, however, not remove chattering effects occurring in sampled time

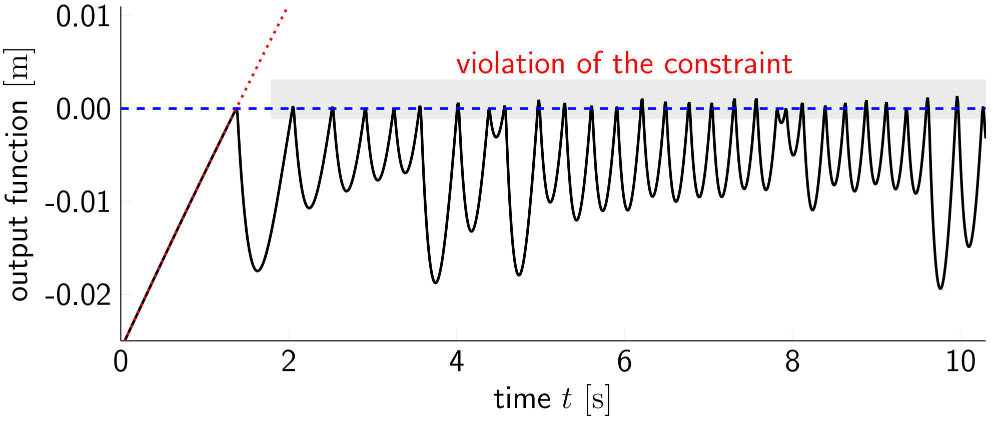


Figure 3.7.: Chattering and boundary violations ░ resulting from continuous invariance control executed in sampled time for — a state tracking ⋯⋯ a desired trajectory while satisfying - - - - a constraint restricting the state to non-positive values.

implementations, as it merely moves the switching input to a higher order derivative.

3.2.2 Control in Sampled Time

In sampled time implementations of invariance control the invariance function is no longer continuously evaluated. The resulting effects are illustrated in Fig. 3.7 for a constraint with relative degree two. There, the state is controlled to track a desired trajectory while invariance control is employed to restrict the state to non-positive values. It may be observed that the state trajectory does not smoothly follow the constraint but instead shows a high-frequency oscillation along the bound, known as chattering. This effect is caused by corrective control being applied for an entire sampling interval instead of the exactly necessary duration, which results in over-actuation. Furthermore, the reversal point when the motion towards inadmissible states changes into a motion towards admissible states is often within the inadmissible set, i.e. at state values corresponding to positive invariance functions. This is due to the fact that the switch to corrective control is not executed at the required instant of time when the invariance function equals zero. Instead, the switch is executed at the next sampling instant when the invariance function already takes a positive value.

As we consider a sampled time implementation in the following, we introduce the notation

$$\boldsymbol{x}[k] = \boldsymbol{x}(t_k) = \boldsymbol{x}(k \cdot T_A)$$

for the value at the k-th sampling instant with $T_A \in \mathbb{R}^+$ being the sampling time.

Causes for Chattering and Constraint Violations

In order to find a solution how to remove the chattering and constraint violations in sampled time implementations, it is necessary to understand the two causes responsible for these effects:

- a switch from corrective control to nominal control and vice versa is only possible at the sampling instants,

- the input $\boldsymbol{u}$ is constant during each sampling interval, i.e.

$$\boldsymbol{u}(t) = \boldsymbol{u}[k] \quad \forall\, t \in [t_k, t_{k+1}[\quad \text{with } \boldsymbol{u}[k] = \boldsymbol{u}(t_k)\ .$$

Naturally, if a switch from corrective to nominal control is carried out later than necessary, this is not ideal as the control goal is not followed during that time but it is not safety-critical. If, however, the switch from nominal to corrective control occurs later than required, the invariance function increases further during that time and violations occur. In order to handle this issue, it is therefore essential to predict the system behavior one step into the future and to start applying corrective control early if a violation during the next step is detected.

When determining corrective control, the second issue needs to be kept in mind. Invariance is guaranteed if (3.30) holds at all times. As $\boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}$ and $b_{\mathrm{c},i}$ change over time depending on state and parameter values, it does not suffice to determine corrective control solely based on their values at the sampling instant as then the invariance condition (3.30) may be violated during the time interval when $\boldsymbol{u}_{\mathrm{c}}$ remains constant and $\boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}$ and $b_{\mathrm{c},i}$ change.

Therefore, in order to reduce chattering and remove the constraint violations we propose combining the knowledge about the system behavior with the sampling time to adapt corrective control accordingly.

Adaptation of the Controller Parameter

The idea of invariance control with chattering reduction [KH14] is to adapt the corrective pseudo inputs $z_{\mathrm{c},i}$ depending on the current and predicted future value of the constraint and invariance functions. The goal is to avoid the overshoots and reduce the oscillations at the boundaries.

The output functions (2.11) and the corresponding invariance functions (3.10) at the k-th sampling instant are given by

$$y_{\mathrm{c},i}[k] := h_{\mathrm{c},i}(\boldsymbol{x}(t_k), \boldsymbol{\eta}(t_k)) \tag{3.49}$$
$$\Phi_i[k] := \Phi_i(\boldsymbol{x}(t_k), \boldsymbol{x}_\eta(t_k), \gamma_i)\ . \tag{3.50}$$

As it is only possible to change the input at the sampling instants, a corrective control input has to be determined which keeps the invariance function at a non-positive value for the entire sampling interval, i.e.

$$\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) \leq 0 \quad \forall\, t \in [t_k, t_{k+1}]\ . \tag{3.51}$$

The value of the invariance function depends on the output functions, which in turn depend on the constraint parameters and the states, the change of which is dependent on the input. The nonlinearity of system and constraint functions renders it hard, if not impossible to determine the exact change of the constraint and invariance functions. Instead, a maximum value may be determined by exploiting the continuity of the involved functions.

Lemma 3.6. *Let the nonlinear system be given by* (2.1) *and the outputs describing the constraints by* (2.11)*. Let Assumptions 2.1, 2.2 and 2.6 hold. Then, there exists an upper* *bound on the constraint functions, their derivatives and the respective invariance function during each time interval* $[t_k, t_{k+1}]$*, i.e.*

$$
\begin{aligned}
h_{c,i}^{(j)}(\boldsymbol{x}, \boldsymbol{x}_\eta) &\leq h_{c,i\ [k]}^{(j)\ \max}(\boldsymbol{x}[k], \boldsymbol{x}_\eta[k], z_i[k]) \quad \forall\ 0 \leq j \leq r_i - 1,\ t \in [t_k, t_{k+1}] \\
\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) &\leq \Phi_{i\ [k]}^{\max}(\boldsymbol{x}[k], \boldsymbol{x}_\eta[k], z_i[k]) \quad \forall\ t \in [t_k, t_{k+1}]
\end{aligned}
$$

The proof is provided in the appendix. Similarly to the constraint and invariance functions, there exist bounds on $\boldsymbol{a}_{\mathrm{c},i}^\intercal$ and $b_{\mathrm{c},i}$, which determine the condition for invariance (3.30).

Lemma 3.7. *Let the nonlinear system be given by* (2.1) *and the outputs describing the constraints by* (2.11)*. Let the linearizing input be determined by* (3.1)*. Let Assumptions 2.1, 2.2 and 2.6 hold. Then, during each time interval* $[t_k, t_{k+1}]$*,* $\boldsymbol{a}_{c,i}^\intercal$ *and* $b_{c,i}$ *from the linearizing transformation are contained within bounded sets* $\mathcal{S}_{a,i}[k] \subset \mathbb{R}^m$*,* $\mathcal{S}_{b,i}[k] \subset \mathbb{R}$*, i.e.*

$$
\begin{aligned}
\boldsymbol{a}_{c,i}^\intercal &\in \mathcal{S}_{a,i}[k] \\
b_{c,i} &\in \mathcal{S}_{b,i}[k]\ .
\end{aligned}
$$

The proof is provided in the appendix. Using Lemma 3.6, we extend the definition of the the set of active constraints for sampled implementations to include all active constraints and the constraints that may become active during the next time step.

$$
\mathcal{B}_{\mathrm{act,chat}}(\boldsymbol{x}[k], \boldsymbol{x}_\eta[k], \boldsymbol{z}[k]) = \left\{ i \in \mathcal{B} \mid \Phi_{i\ [k]}^{\max}(\boldsymbol{x}[k], \boldsymbol{x}_\eta[k], z_i[k]) \geq 0 \right\} \tag{3.52}
$$

For those active constraints, that do not yet have a positive invariance function, the corrective pseudo input is extended by an additional case to ensure that during the sampling interval, the invariance function does not increase to a positive value. It is given by

$$
z_{\mathrm{c},i}[k] = \begin{cases} \gamma_i & \text{if } t \in (\mathcal{N}_{a,i}(\gamma_i) \cup \mathcal{N}_{b,i}(\gamma_i)) \\ 0 & \text{if } t \in (\mathcal{N}_{c,i}(\gamma_i) \cup \mathcal{N}_{d,i}(\gamma_i)) \\ \hat{\gamma}_i & \text{if } t \in \mathcal{N}_{e,i}(\gamma_i, \boldsymbol{x}[k], \boldsymbol{x}_\eta[k], z_i[k]) \\ z_{\mathrm{no},i} & \text{else.} \end{cases} \tag{3.53}
$$

with $\gamma_i < 0$, the sets from (3.17)–(3.20),

$$
\mathcal{N}_{e,i}(\gamma_i, \boldsymbol{x}[k], \boldsymbol{x}_\eta[k], z_i[k]) = \{ t \mid \Phi_i[k] < 0\ \wedge\ \Phi_{i\ [k]}^{\max}(\boldsymbol{x}[k], \boldsymbol{x}_\eta[k], z_i[k]) \geq 0 \} \tag{3.54}
$$

and the adapted parameter $\hat{\gamma}_i < 0$ chosen such that

$$
\Phi_{i\ [k]}^{\max}(\boldsymbol{x}[k], \boldsymbol{x}_\eta[k], \hat{\gamma}_i) = 0 \tag{3.55}
$$

holds. The adapted value $\hat{\gamma}_i$ ensures that the invariance function remains at non-positive values during the following sampling interval. Furthermore, its magnitude is generally smaller than $|\gamma_i|$ thus reducing the corrective effort and the over-actuation. The corrective pseudo input for sampled system is the basis for determining corrective control for sampled systems

$$
\begin{aligned}
\boldsymbol{u}_{\mathrm{c}}[k] = \underset{\boldsymbol{u}}{\operatorname{argmin}}\ & C\left(\boldsymbol{u}, \boldsymbol{u}_{\mathrm{no}}[k]\right) \\
\text{s.t. }\ & \boldsymbol{a}_{\mathrm{c},i}^\intercal \boldsymbol{u} + b_{\mathrm{c},i} \leq z_{\mathrm{c},i}[k] \\
\forall\ \boldsymbol{a}_{\mathrm{c},i}^\intercal \in \mathcal{S}_{a,i}[k],\ & b_{\mathrm{c},i} \in \mathcal{S}_{b,i}[k],\ i \in \mathcal{B}_{\mathrm{act,chat}}(\boldsymbol{x}[k], \boldsymbol{x}_\eta[k], \boldsymbol{z}[k])
\end{aligned} \tag{3.56}
$$

Algorithm 1 Corrective control in sampled systems.

```
Set $\mathcal{B}_{\max} = \emptyset$
Set $\boldsymbol{u}_{\mathrm{c}}[k] = \boldsymbol{u}_{\mathrm{no}}[k]$
Set $\boldsymbol{z}[k] = \boldsymbol{z}_{\mathrm{no}}[k]$
Determine $\mathcal{B}_{\mathrm{act,chat}}(\boldsymbol{x}[k], \boldsymbol{x}_{\eta}[k], \boldsymbol{z}[k])$
while $\mathcal{B}_{\mathrm{act,chat}}(\boldsymbol{x}[k], \boldsymbol{x}_{\eta}[k], \boldsymbol{z}[k]) \not\subseteq \mathcal{B}_{\max}$ do
    Set $\mathcal{B}_{\max} = \mathcal{B}_{\max} \cup \mathcal{B}_{\mathrm{act,chat}}(\boldsymbol{x}[k], \boldsymbol{x}_{\eta}[k], \boldsymbol{z}[k])$
    for $i \in \mathcal{B}_{\max}$ do
        Determine $z_{\mathrm{c},i}[k]$
    Calculate $\boldsymbol{u}_{\mathrm{c}}[k]$ using (3.57)
    for $i \in \mathcal{B}$ do
        Calculate $z_i[k] = \boldsymbol{a}_{\mathrm{c},i}^{\intercal}[k]\boldsymbol{u}_{\mathrm{c}}[k] + b_{\mathrm{c},i}[k]$
    Determine $\mathcal{B}_{\mathrm{act,chat}}(\boldsymbol{x}[k], \boldsymbol{x}_{\eta}[k], \boldsymbol{z}[k])$
```

where the cost function $C\left(\boldsymbol{u}, \boldsymbol{u}_{\mathrm{no}}[k]\right)$ fulfills the same properties as defined in (3.31). It may be observed that the corrective pseudo input $z_{\mathrm{c},i}$ and the set of active constraints $\mathcal{B}_{\mathrm{act,chat}}$ both depend on the pseudo input $\boldsymbol{z}$, which in turn depends on the input $\boldsymbol{u} = \boldsymbol{u}_{\mathrm{c}}$. As the determination of the input (3.56), again depends on the corrective pseudo input and the active constraints.

In order to resolve this issue, we propose determining a maximum set of active constraints $\mathcal{B}_{\max}$ as shown in Algorithm 1, which allows to define the explicit minimization

$$\boxed{\begin{aligned} \boldsymbol{u}_{\mathrm{c}}[k] &= \underset{\boldsymbol{u}}{\operatorname{argmin}}\; C\left(\boldsymbol{u}, \boldsymbol{u}_{\mathrm{no}}[k]\right) \\ \text{s.t. } \boldsymbol{a}_{\mathrm{c},i}^{\intercal}\boldsymbol{u} + b_{\mathrm{c},i} &\leq z_{\mathrm{c},i}[k] \quad \forall\, \boldsymbol{a}_{\mathrm{c},i}^{\intercal} \in \mathcal{S}_{a,i}[k],\ b_{\mathrm{c},i} \in \mathcal{S}_{b,i}[k],\ i \in \mathcal{B}_{\max}\ . \end{aligned} \qquad (3.57)}$$

In Algorithm 1, the set of active constraints is derived by iteratively determining the control input, the active constraints and adding newly active constraints to a maximum set of active constraints $\mathcal{B}_{\max}$ until no new constraints are added. The set $\mathcal{B}_{\max}$ is guaranteed to converge, as it maximally contains all constraints, i.e. $\mathcal{B}_{\max} = \mathcal{B}$. Using the algorithm, the implicit dependence in (3.56) is resolved, thus allowing the determination of corrective control.

Theorem 3.5 (Invariance in sampled systems)**.** *Let the nonlinear system, which is sampled with a sampling time T_A, be given by* (2.1) *and the outputs describing the constraints by* (2.11)*. Let the linearizing input be determined by* (3.1) *and the elements of the corrective pseudo input by* (3.53)*. Let Assumptions 2.1, 2.2, 2.5, 2.6 and 3.1 hold. Then, if the system states are within the invariant set at some sampling instant t_k, i.e. $\boldsymbol{x}(t_k) \in \mathcal{G}(\boldsymbol{x}_{\eta}, \boldsymbol{\gamma})$, the corrective control input determined by* (3.57) *and Algorithm 1 renders the invariant set* (3.14) *positively invariant for all $t \geq t_k$.*

The proof is provided in the appendix. Using the theorem, invariance control for sampled systems with arbitrary sampling times T_A may be designed. Determining the necessary bounds on the invariance function and the sets $\mathcal{S}_{a,i}$ and $\mathcal{S}_{b,i}$ may however proof difficult depending on the application. Furthermore, solving the optimization requires an approach for solving a minimization with uncertain constraints. This may be resolved by using robust invariance control, which is introduced in Section 4.1.

In cases where the following assumption holds, the approach may, however, be simplified such that robust invariance control is not required.

Assumption 3.6. *During one sampling interval, the pseudo control input is constant, i.e.* $\boldsymbol{z}(t) = \boldsymbol{z}(t_k)\ \forall t \in [t_k, t_{k+1})$.

This relation holds exactly if all $\boldsymbol{a}_{\mathrm{c},i}^{\intercal}$ and $b_{\mathrm{c},i}$ are constant or approximately for small sampling times when the changes are negligible. In this case, (3.6)–(3.9) may be used to determine the invariance function in the following time step, which replaces the maximum invariance function. Furthermore, it is no longer necessary to determine the sets $\mathcal{S}_{a,i}$ and $\mathcal{S}_{b,i}$ as $\boldsymbol{a}_{\mathrm{c},i}^{\intercal}$ and $b_{\mathrm{c},i}$ are (nearly) constant. The following example illustrates how the corrective pseudo input for a constraint with relative degree two is determined using Assumption 3.6.

Example 3.9 (Parameter adaptation for relative degree two)**.** The output functions in the following time step depend on the current pseudo control input $z_i[k]$. Using Assumption 3.6 means that for a system with relative degree r_i the derivative $y_{\mathrm{c},i}^{(r_i-1)}$ changes linearly during one time step

$$y_{\mathrm{c},i}^{(r_i-1)}[k+1] = y_{\mathrm{c},i}^{(r_i-1)}[k] + \int_{t_k}^{t_k+T_A} \underbrace{y_{\mathrm{c},i}^{(r_i)}[k]}_{=z_i[k]}\,\mathrm{d}t = y_{\mathrm{c},i}^{(r_i-1)}[k] + T_A z_i[k]\ . \tag{3.58}$$

Furthermore, a finite Taylor series determines the values of the lower order derivatives $y_{\mathrm{c},i}[k+1], \dots, y_{\mathrm{c},i}^{(r_i-2)}[k+1]$

$$y_{\mathrm{c},i}^{(r)}[k+1] = \sum_{j=0}^{r_i-r} \frac{(T_A)^j}{(j)!} y_{\mathrm{c},i}^{(r+j)}[k]\ , \tag{3.59}$$

where $y_{\mathrm{c},i}^{(r_i)}[k] = z_i[k]$.

For $r_i = 2$, this means that

$$y_{\mathrm{c},i}[k+1] = y_{\mathrm{c},i}[k] + T_A\dot{y}_{\mathrm{c},i}[k] + \frac{T_A^2}{2} z_i[k] \tag{3.60}$$

$$\dot{y}_{\mathrm{c},i}[k+1] = \dot{y}_{\mathrm{c},i}[k] + T_A z_i[k]\ . \tag{3.61}$$

For a system with relative degree 2, the invariance function is given by (3.13), which evaluates to

$$\Phi_i[k+1] = \begin{cases} y_{\mathrm{c},i}[k] + T_A\dot{y}_{\mathrm{c},i}[k] + \frac{T_A^2}{2} z_i[k] & \dot{y}_{\mathrm{c},i}[k+1] \le 0 \\ y_{\mathrm{c},i}[k] + T_A\dot{y}_{\mathrm{c},i}[k] + \frac{T_A^2}{2} z_i[k] - \frac{(\dot{y}_{\mathrm{c},i}[k] + T_A z_i[k])^2}{2\gamma_i} & \dot{y}_{\mathrm{c},i}[k+1] > 0 \end{cases} \tag{3.62}$$

in the $(k+1)$-th time step. Pseudo corrective control is given by (3.53) and $\hat{\gamma}_i$ is determined by setting $\Phi_i[k+1] = 0$ and solving the equation for $z_i[k] = \hat{\gamma}_i$.

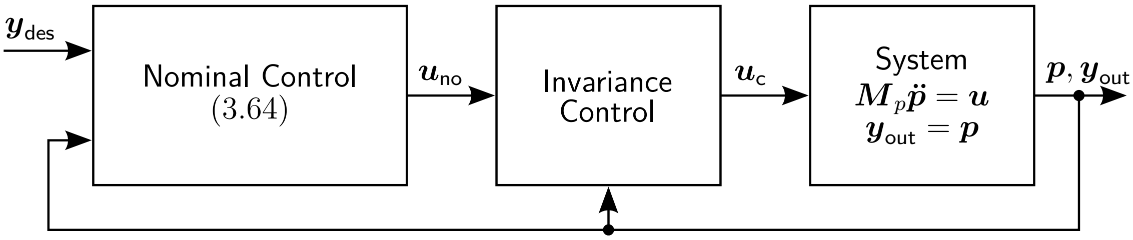


Figure 3.8.: Structure of the simulation model for the evaluation of invariance control in sampled time.

Numerical Evaluation

In order to illustrate the findings a simulation is carried out in *Matlab/Simulink*. We compare the proposed novel method to the standard invariance controller [SWB08] and the chattering reduction method presented in our earlier work [KLH12] for different sets of boundaries.

Simulation Setup We consider a trajectory following problem for a robotic system. The simulation model with nominal control, invariance control and system is shown in Fig. 3.8. The system is a simplified Cartesian robot model

$$\boldsymbol{M}_p\, \ddot{\boldsymbol{p}} = \boldsymbol{u} \;, \qquad \boldsymbol{y}_{\text{out}} = \boldsymbol{p} \tag{3.63}$$

where $\boldsymbol{M}_p \in \mathbb{R}^{3\times3}$ is the mass matrix and $\boldsymbol{p} \in \mathbb{R}^3$ the position vector consisting of the translational Cartesian directions. Nominal control is given by a PD control law

$$\boldsymbol{u}_{\text{no}} = \boldsymbol{D}_p(\dot{\boldsymbol{y}}_{\text{des}} - \dot{\boldsymbol{y}}_{\text{out}}) + \boldsymbol{K}_p(\boldsymbol{y}_{\text{des}} - \boldsymbol{y}_{\text{out}}) \;, \tag{3.64}$$

with the proportional constant $\boldsymbol{K}_p \in \mathbb{R}^{3\times3}$ and the damping constant $\boldsymbol{D}_p \in \mathbb{R}^{3\times3}$, enforcing the desired motion $\dot{\boldsymbol{y}}_{\text{des}}$.

The reference trajectory of the system $\boldsymbol{y}_{\text{des}}$ is a circular movement starting and ending in the same point. Three constraints as discussed in Example 2.2 are defined by minimum p_1- and p_2-values and a maximum p_1-value. The fourth constraint is given by a tilted plane. The simulation is carried out for three different tilted planes. The simulation parameters and constraint functions corresponding to the tilted planes are provided in Table 3.2.

Note that for the numerical evaluation, the corrective pseudo input (3.53) determined in Example 3.9 is slightly refined by distinguishing two cases for $\hat{\gamma}_i$:

- Determine $\hat{\gamma}_i$ such that $\Phi_i[k] = 0$.
- Determine $\hat{\gamma}_i$ such that $\Phi_i[k+1] \leq 0$ for $z_i[k+1] = 0$.

The first case ensures that violations are avoided as $\Phi_i[k] = \Phi_i[k+1]$ for constant z_i. The second case ensures that a value $z_{\text{c},i} = 0$ suffices to render the system invariant from the next time step onwards. In order to fulfill both cases, the minimal value is chosen.

$$\hat{\gamma}_i = \min\left(-\frac{\dot{y}_{\text{c},i}[k]}{T_A}, \frac{\dot{y}_{\text{c},i}^2[k]}{2y_{\text{c},i}[k]}\right)$$

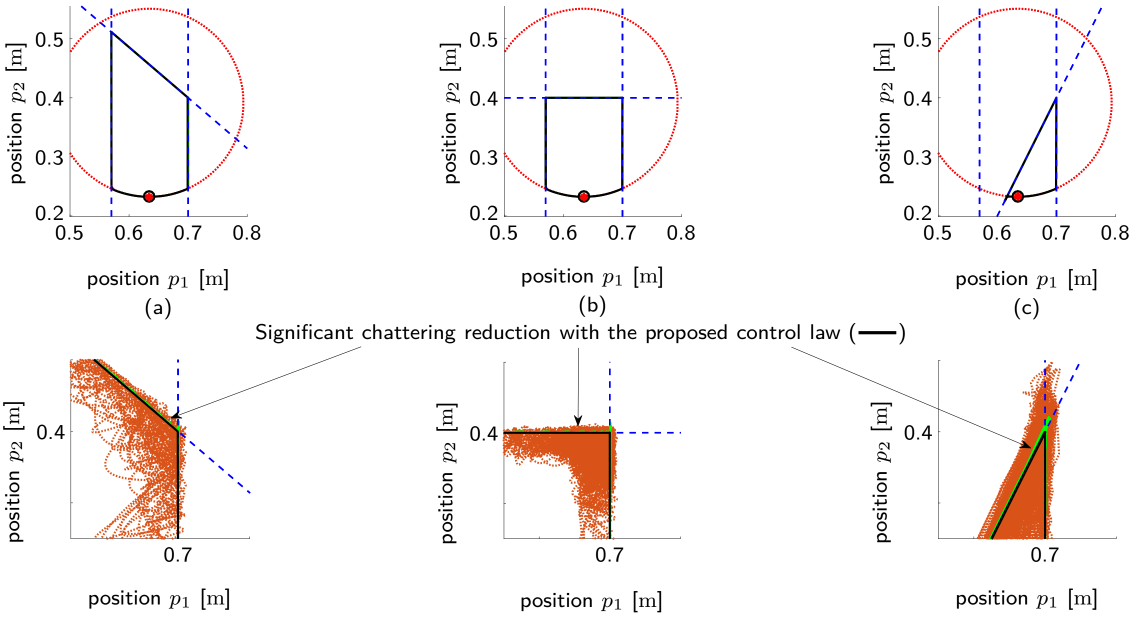
Table 3.2.: Simulation parameters for the numerical evaluation of chattering reduction.

Sampling time	T_A	$0.001\,\mathrm{s}$
Simulation time	T_{end}	$60.000\,\mathrm{s}$
Cartesian impedance	$\boldsymbol{K}_p$	$\boldsymbol{I}_3 \cdot 600\,\mathrm{N/m}$
Cartesian damping	$\boldsymbol{D}_p$	$\boldsymbol{I}_3 \cdot 80\,\mathrm{N\,s/m}$
Mass matrix	$\boldsymbol{M}_p$	$\boldsymbol{I}_3 \cdot 15\,\mathrm{kg}$
Lower position constraints	$p_{1,\min}$	$0.57\,\mathrm{m}$
	$p_{2,\min}$	$0.13\,\mathrm{m}$
Upper position constraint	$p_{1,\max}$	$0.70\,\mathrm{m}$
Tilted plane constraints	$y_{\mathrm{c},1}$	$y_{\mathrm{c},1} = +0.86\,p_1 + p_2 - 1.0\,\mathrm{m}$
	$y_{\mathrm{c},2}$	$y_{\mathrm{c},2} = +0.00\,p_1 + p_2 - 0.4\,\mathrm{m}$
	$y_{\mathrm{c},3}$	$y_{\mathrm{c},3} = -2.00\,p_1 + p_2 + 1.0\,\mathrm{m}$
Control parameter	γ	$-18\,\mathrm{m/s^2}$
Parameters for [KLH12]	α_1	0.25
	α_0	0.75
	$\bar{k}$	50

Results The simulation is carried out three times for each control law, once for every set of constraints. The results are shown in Fig. 3.9. Fig. 3.9a–3.9c depict the complete trajectory in p_1- and p_2-coordinates, the corresponding reference trajectory and the boundaries. In Fig. 3.9d–3.9f, the behavior of the system in the right upper corner of the admissible set is shown in more detail. In order to illustrate, that the configuration of the boundaries does not effect the invariance, the constraints are chosen such that three significantly different angles occur in the upper right corner.

In Fig. 3.9a–3.9c, the trajectories generated by the three control laws almost coincide. The figures show that the boundaries are overall followed by each control law. In Fig. 3.9d–3.9f, we observe the differences between the control schemes. The novel control law eliminates the chattering effect and the trajectory follows the boundaries almost exactly. It also shows no violation of the constraints, emphasizing that the system is made controlled invariant as stated in Theorem 3.5. In contrast, the standard control scheme and the chattering reduction method from [KLH12], show chattering effects and do slightly violate the constraints. While invariance control from Section 3.1 ignores the sampled time implementation of the controller, in [KLH12], Euler's explicit method is used to approximate the invariance function in the following time step, which is overly simplified. The approach proposed in this section, on the other hand, explicitly considers the effects of the sampled implementation of the controller in combination with a continuous system, which yields a more accurate solution and achieves a better control performance.

Naturally, the simplifying Assumption 3.6, which holds for the numerical evaluation, does not hold for all systems, especially not for larger sampling times or highly nonlinear systems with quickly changing parameters. Therefore, it may be preferable to use a control method for constraint enforcement that does not generate a switching control law and hence does not require chattering reduction. In such cases, modeling the constraints and deriving control



(a) (b) (c) (d) (e) (f)

Figure 3.9.: Simulation results for three different sets of boundaries - - - - for a system following the desired position ······ from start ○ to end point • controlled via continuous invariance control from Section 3.1 ······, the chattering reduction method from [KLH12] —— and the proposed approach ——. Depicted are the view of the complete trajectory with a) constraint $y_{c,1}$, b) constraint $y_{c,2}$, c) constraint $y_{c,3}$; and the detailed view of the upper right corner of the admissible area with a) constraint $y_{c,1}$, b) constraint $y_{c,2}$, c) constraint $y_{c,3}$.

via control barrier functions (CBFs) is an alternative option.

3.3 Control Barrier Function-based Control

There are various ways to include barrier certificates and control barrier functions (CBFs) in control schemes in order to achieve the satisfaction of output and state constraints [Wil03; TTH13; WS15; Xu18]. The general idea is to define a CBF as a function of the constraint and to design a control input that restricts the growth of the CBF such that an invariant subset of the admissible set is generated within which the state evolves.

In this section, we introduce the feedback linearization based control approach for control affine systems as suggested for single, static constraints by [AGT14; HXA15]. It allows for the desired implementation as add-on to an existing control loop of system (2.1) and nominal control as depicted in Fig. 2.1 [RKH16]. This means that, similar to invariance control, CBF-based control enables designing nominal control to achieve a desired performance and stable tracking without accounting for possible constraints. However, CBF-based control does not rely on invariance functions and a switching control law for constraint satisfaction but instead introduces CBFs which resemble infinite potentials to derive a continuous corrective control input.

Note that, in contrast to invariance control, CBFs generally use a constraint description that is defined inversely to Def. 2.2.

3.3.1 Alternative Constraint Definition

In order to be consistent with literature [AGT14; RKH16] on CBFs, we define the constraint functions $h_{\mathrm{B},i}(\boldsymbol{x}, \boldsymbol{x}_\eta)$ with $i \in \mathcal{B}$ inversely to the functions $h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta)$ introduced in Def. 2.1.

$$y_{\mathrm{B},i} = h_{\mathrm{B},i}(\boldsymbol{x}, \boldsymbol{x}_\eta) = -h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta) = -y_{\mathrm{c},i} \tag{3.65}$$

Note that this definition does not change the admissible set from Def. 2.2 but it allows for an alternative representation of the set $\mathcal{H}$ and its bound $\partial\mathcal{H}$

$$\mathcal{H}(\boldsymbol{x}_\eta) = \{\boldsymbol{x} \mid h_{\mathrm{B},i}(\boldsymbol{x}, \boldsymbol{x}_\eta) \geq 0 \quad \forall\, i \in \mathcal{B}\} \tag{3.66}$$

$$\partial\mathcal{H}(\boldsymbol{x}_\eta) = \{\boldsymbol{x} \mid h_{\mathrm{B},i}(\boldsymbol{x}, \boldsymbol{x}_\eta) \geq 0 \quad \forall\, i \in \mathcal{B} \;\wedge\; \exists\, j \in \mathcal{B}:\; h_{\mathrm{B},j}(\boldsymbol{x}, \boldsymbol{x}_\eta) = 0\} \tag{3.67}$$

As $\mathcal{H}$ remains unchanged, there exist admissible states if Assumption 2.2 holds. The goal of the following control derivation is to design a control law, which renders a subset of the admissible set positively invariant according to Def. 3.3.

For CBF-based control, the initial step is to use the functions (3.65) to determine a CBF for each constraint.

3.3.2 Control Barrier Functions

Essentially, a CBF $B_i : \mathbb{R}^n \times \mathbb{R}^{(r_{\max}+1)n_\eta} \to \mathbb{R}^+$ is a non-negative function with a small value for states far from a constraint, which grows to infinity as the state approaches the bound of the admissible set $\partial\mathcal{H}$. For a function B_i to qualify as a CBF, it has to fulfill the following properties.

(CBF-p$_1$) A valid CBF is non-negative on $\mathcal{H}$

$$\inf_{\boldsymbol{x}\in\mathcal{H}} B_i(\boldsymbol{x}, \boldsymbol{x}_\eta) \geq 0 \ .$$

(CBF-p$_2$) For any $\boldsymbol{x}_\eta \in \mathbb{R}^{(r_{\max}+1)n_\eta}$, the barrier grows to infinity as a solution $\boldsymbol{x}$ approaches the constraint from inside the admissible set

$$\lim_{\boldsymbol{x}\to\partial\mathcal{H}^+} B_i(\boldsymbol{x}, \boldsymbol{x}_\eta) = \infty \ .$$

(CBF-p$_3$) The CBF $B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$ grows with the growth rate

$$\dot{B}_i \leq \frac{\mu_i}{B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)}, \ \text{where } \mu_i > 0 \ .$$

With these properties, the CBF may grow quickly, if the state is far away from a constraint as the right side of CBF-p$_3$ takes large values in this case. If, however, the state is close to the bound, CBF-p$_2$ ensures that CBF-p$_3$ becomes more restrictive, i.e. the CBF grows slower and finally stops as the state approaches the constraint and the derivative is restricted to non-positive values. By enforcing this condition, constraint violation is prevented as it ensures that the growth of the function stops when the state approaches the constraint.

Remark 3.8. For guaranteed constraint satisfaction, it is necessary to be able to actively enforce that CBF-p$_3$ holds. Therefore, it is necessary to design the CBF such that its time derivative depends on the control input $\boldsymbol{u}$, i.e. $\dot{B}_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{u})$. Otherwise constraint satisfaction may only be achieved if CBF-p$_3$ holds naturally on all boundaries of the admissible set.

For the actual design of CBF $B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$, it is important to realize that it essentially behaves like the inverse of a so-called class κ function [Kha96].

Definition 3.10. A continuous function $\alpha : [0, \infty) \to [0, \infty)$ belongs to ***class*** κ if it is strictly increasing and $\alpha(0) = 0$.

Combining the properties

$$\inf_{\boldsymbol{x} \in \mathcal{H}} \frac{1}{\alpha(y_{\mathrm{B},i})} \geq 0, \qquad \lim_{x \to \partial\mathcal{H}^+} \frac{1}{\alpha(y_{\mathrm{B},i})} = \infty$$

of inverse class κ functions with the desired properties CBF-p$_1$ to CBF-p$_3$ yields the definition of CBFs [AGT14].

Definition 3.11. A ***control barrier function (CBF)*** $B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$ corresponding to a constraint function $y_{\mathrm{B},i} = h_{\mathrm{B},i}(\boldsymbol{x}, \boldsymbol{x}_\eta)$ is a locally Lipschitz function, for which there exist class κ functions α_1, α_2 and a constant $\mu_i > 0$ such that $B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$ fulfills for all $\boldsymbol{x} \in \mathcal{H}$

$$\frac{1}{\alpha_1(y_{\mathrm{B},i})} \leq B_i(\boldsymbol{x}, \boldsymbol{x}_\eta) \leq \frac{1}{\alpha_2(y_{\mathrm{B},i})} \tag{3.68}$$

$$\inf_{\boldsymbol{u} \in \mathbb{R}^m} \left[\dot{B}_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{u}) - \frac{\mu_i}{B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)} \right] \leq 0 \tag{3.69}$$

with

$$\begin{aligned} \dot{B}_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{u}) &= \boldsymbol{a}_{\mathrm{B},i}^{\mathsf{T}} \boldsymbol{u} + b_{\mathrm{B},i} \\ \boldsymbol{a}_{\mathrm{B},i}^{\mathsf{T}} &= \mathscr{L}_{\boldsymbol{x}\boldsymbol{G}} B_i(\boldsymbol{x}, \boldsymbol{x}_\eta) \\ b_{\mathrm{B},i} &= \mathscr{L}_{\boldsymbol{x}_\eta \boldsymbol{f}_\eta} B_i(\boldsymbol{x}, \boldsymbol{x}_\eta) + \mathscr{L}_{\boldsymbol{x}\boldsymbol{f}} B_i(\boldsymbol{x}, \boldsymbol{x}_\eta) \end{aligned} \tag{3.70}$$

and locally Lipschitz Lie derivatives $\mathscr{L}_{\boldsymbol{x}_\eta \boldsymbol{f}_\eta} B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$, $\mathscr{L}_{\boldsymbol{x}\boldsymbol{f}} B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$, $\mathscr{L}_{\boldsymbol{x}\boldsymbol{G}} B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$.

This definition is the basis for designing suitable CBFs.

Remark 3.9. If $\dot{B}_i$ fulfills (3.69) independently from $\boldsymbol{u}$, it is called a ***barrier function***.

Remark 3.10. By designing the CBFs according to Def. 3.11, I/O-linearization with respect to a CBF B_i always yields a relative degree of one, which may be different to the relative degree of the constraint function $y_{\mathrm{B},i}$.

The following example shows how CBFs for constraint functions of relative degree one, such as velocity constraints on robotic systems, may be derived from a class κ function.

Example 3.10. Consider the function

$$f(r) = -\frac{1}{\ln\left(\dfrac{r}{r+1}\right)} \, .$$

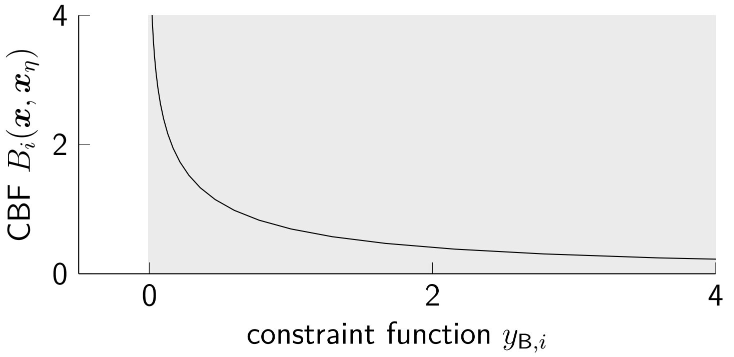


Figure 3.10.: Illustration of the control barrier function (CBF) — from Example 3.10 and the related admissible set ░.

This function fulfills

$$
\begin{aligned}
f(r) &\geq 0 \quad \forall\, r \in \mathbb{R}^+ \\
f(0) &= \lim_{r \to 0^+} f(r) = 0 \\
\lim_{r \to \infty} f(r) &\to \infty
\end{aligned}
$$

and has a strictly positive derivative

$$
\frac{\mathrm{d}f(r)}{\mathrm{d}r} = \left(\ln\left(\frac{r}{r+1}\right)\right)^{-2} \frac{1}{r(r+1)} > 0 \quad \forall r \in \mathbb{R}^+ \ ,
$$

which means that it is strictly increasing, thus showing that it is a class κ function. Hence its inverse is a CBF candidate as it fulfills (3.68) naturally due to being an inverse class κ function.

$$
B_i(\boldsymbol{x}, \boldsymbol{x}_\eta) = -\ln\left(\frac{y_{\mathrm{B},i}}{1 + y_{\mathrm{B},i}}\right) \qquad \text{with } y_{\mathrm{B},i} = h_{\mathrm{B},i}(\boldsymbol{x}, \boldsymbol{x}_\eta) \tag{3.71}
$$

The time derivative is given by

$$
\dot{B}_i = -\frac{1}{y_{\mathrm{B},i}(1 + y_{\mathrm{B},i})}\, \dot{y}_{\mathrm{B},i}
$$

and transforms into

$$
\dot{B}_i = \boldsymbol{a}_{\mathrm{B},i}^\intercal \boldsymbol{u} + b_{\mathrm{B},i}
$$

with $\boldsymbol{a}_{\mathrm{B},i} \neq 0$ if the constraint function fulfills Assumption 2.2 with relative degree one and $(y_{\mathrm{B},i}(1 + y_{\mathrm{B},i}))^{-1} > 0$. Therefore there exists an input $\boldsymbol{u}$ such that (3.69) is fulfilled and hence (3.71) is a valid CBF. Figure 3.10 shows the CBF as a function of $y_{\mathrm{B},i}$ and illustrates the positivity of the function and it growth towards infinity on approaching the bound $y_{\mathrm{B},i} = 0$.

As in torque-controlled robotic systems with the generalized dynamics (2.4), position constraints have relative degree two, $\dot{y}_{\mathrm{B},i}$ does not depend on the input, which means that (3.71) derived in Example 3.10 is not a valid CBF. For such cases, the function

$$B_i(\boldsymbol{x}, \boldsymbol{x}_\eta) = -\ln\left(\frac{y_{\mathrm{B},i}}{1 + y_{\mathrm{B},i}}\right) + \zeta_a \frac{\zeta_b \dot{y}_{\mathrm{B},i}^2}{1 + \zeta_b \dot{y}_{\mathrm{B},i}^2} \tag{3.72}$$

with positive design parameters $\zeta_a, \zeta_b \in \mathbb{R}^+$ is an admissible CBF, which is shown in [HXA15, Theorem 2]. The function includes $\dot{y}_{\mathrm{B},i}$ in the CBF and with Assumption 2.2, which ensures that $\mathscr{L}_{\boldsymbol{x}\boldsymbol{G}} B_i(\boldsymbol{x}, \boldsymbol{x}_\eta) \neq \boldsymbol{0}_{1\times m}$ holds. Note that small values of the design parameters cause the system to stop further from the constraints, whereas large values increase the risk of a violation in the presence of uncertainties, i.e. a reasonable trade-off has to be found depending on the application.

Remark 3.11. In case of relative degrees $r_i > 2$ a suitable CBF may be derived using the results from [HXA15] or with a slightly different condition (3.69) the results from [Xu18].

In the following section, we use the characteristics of the CBFs to develop a control structure, which combines multiple constraints with an arbitrary desired nominal behavior.

3.3.3 Admissible Control Inputs

The property CBF-p$_3$ and the condition (3.69) on the derivative of the CBF provide the basis for deriving suitable control inputs. By enforcing this condition, constraint violation is prevented as it ensures that the growth of the function stops when the state approaches the constraint.

Lemma 3.8. *Let the nonlinear system be given by* (2.1) *and the outputs describing the constraints by* (2.11). *Let Assumptions 2.1 and 2.2 hold. Let $B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$ be a CBF corresponding to the constraint function $y_{B,i}$. Then any Lipschitz control input $\boldsymbol{u} \in \mathcal{M}_{B,i}$ with*

$$\mathcal{M}_{B,i}(\boldsymbol{x}, \boldsymbol{x}_\eta, \mu_i) = \{\boldsymbol{u} \in \mathbb{R}^m \mid \boldsymbol{a}_{B,i}^\intercal \boldsymbol{u} + b_{B,i} \leq \frac{\mu_i}{B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)}\}$$

with $\boldsymbol{a}_{B,i}$, $b_{B,i}$ from (3.70) *renders the interior of the set*

$$\mathcal{H}_i(\boldsymbol{x}_\eta) = \{\boldsymbol{x} \mid h_{B,i}(\boldsymbol{x}, \boldsymbol{x}_\eta) \geq 0\}$$

controlled positively invariant.

This lemma states that if there exists a CBF and a control input fulfilling (3.69), constraint satisfaction of a single constraint is guaranteed. Proof of this result is provided in [AGT14, Corollary 1].

Extending this concept to multiple constraints yields the set of admissible control values $\mathcal{M}_B(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\mu})$

$$\mathcal{M}_B(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\mu}) = \left\{\boldsymbol{u} \in \mathbb{R}^m \mid \boldsymbol{a}_{\mathrm{B},i}^\intercal \boldsymbol{u} + b_{\mathrm{B},i} \leq \frac{\mu_i}{B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)}, \forall i \in \mathcal{B}\right\}, \tag{3.73}$$

which is derived be combining the individual conditions from (3.69) and where $\mu_i > 0$ are design parameters corresponding to each B_i. Hence, $\mathcal{M}_B$ is the intersection of the set $\mathcal{M}_{B,i}$ from Lemma 3.8 corresponding to the single constraints. For admissible control values to exist, the following assumption has to hold.

Assumption 3.7. *The set of admissible control values $\mathcal{M}_B(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\mu})$, defined in* (3.73)*, is non-empty, i.e. $\mathcal{M}(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\mu}) \neq \emptyset$.*

This is similar to Assumption 3.1 for invariance control, the implications of which are illustrated in Fig. 3.2. Combining the properties of $\mathcal{H}$ with the set of control values $\mathcal{M}_B$ and Lemma 3.8, positive invariance of $\mathcal{H}$ is shown.

Theorem 3.6. *Let the nonlinear system be given by* (2.1) *and the outputs describing the constraints by* (2.11)*. Let Assumptions 2.1, 2.2 and 3.7 hold. Let $B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$ be CBFs corresponding to each constraint function $y_{B,i}$, $i \in \mathcal{B}$. Then any control input $\boldsymbol{u} \in \mathcal{M}_B(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\mu})$ renders the interior of the admissible set $\mathcal{H}$ positively invariant.*

The proof is provided in the appendix. Intuitively, Theorem 3.6 shows that by choosing only control values from the set $\mathcal{M}_B$, the system is forced to stay within the admissible set of states $\mathcal{H}$ for all times, thus guaranteeing constraint satisfaction. Additionally, choosing a Lipschitz continuous input $\boldsymbol{u}$ avoids large, instantaneous changes in the control value, thus enabling torque-controlled robotic systems to follow the desired torque and reducing the overall stress of the system. Based on these results, corrective control is derived to unify any nominal control with the CBF approach for multiple constraints.

3.3.4 Corrective Control

Enforcing multiple constraints while executing a desired task requires the adjustment of nominal control to ensure the satisfaction of all constraints. This is achieved by merging the conditions on the control input for constraint satisfaction given by (3.73) with any stabilizing nominal control law via a quadratic program (QP)

$$\begin{aligned} \boldsymbol{u}_\mathrm{c} = \underset{\boldsymbol{u}}{\operatorname{argmin}}\ & \|\boldsymbol{u} - \boldsymbol{u}_\mathrm{no}\|_2^2 \\ \text{s.t. } & \boldsymbol{a}_{\mathrm{B},i}^\intercal \boldsymbol{u} + b_{\mathrm{B},i} \leq \tfrac{\mu_i}{B_i(\boldsymbol{x},\boldsymbol{x}_\eta)} \quad \forall\, i \in \mathcal{B} \end{aligned} \tag{3.74}$$

with $\boldsymbol{a}_{\mathrm{B},i}$ and $b_{\mathrm{B},i}$ from (3.70).

Similar to invariance control, this formulation yields a convex optimization with a unique global minimum.

Lemma 3.9. *The minimization* (3.74) *is strictly convex and any local minimum is the unique global minimum.*

The proof is analog to the proof of Lemma 3.3. Hence, the optimization may be solved in a computationally efficient manner [Boy04, p.8], which is ideal for control in real-time with high sampling rates. In addition, the fact that any local minimum is the unique global minimum ensures that nominal control $\boldsymbol{u}_\mathrm{no}$ is applied if it fulfills the optimization conditions, since it is the minimum.

Theorem 3.7. *Let the nonlinear system be given by* (2.1) *and the outputs describing the constraints by* (2.11)*. Let a locally Lipschitz nominal control signal $\boldsymbol{u}_{no}$ be given. Let further Assumptions 2.1, 2.2, 2.4–2.6 and 3.7 hold. Let $B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$ be CBFs corresponding to each constraint function $y_{B,i}$, $i \in \mathcal{B}$. Then, the control law $\boldsymbol{u}_c$ obtained by solving the minimization* (3.74) *is Lipschitz continuous and renders the interior of the admissible set $\mathcal{H}$ positively invariant.*

The proof is provided in the appendix. Theorem 3.7 shows that the CBF-based approach is applicable to nominally controlled systems with multiple, non-conflicting constraints. By solving the QP (3.74) the invariance of $\mathrm{Int}(\mathcal{H})$ and a minimal error to nominal control are obtained simultaneously. Thus, the constraints are satisfied and nominal control is executed whenever the state $\boldsymbol{x}$ is sufficiently far away from the bounds.

Additional characteristics and capabilities of the approach are stated without proof in the following remarks.

Remark 3.12. Even though Theorem 3.7 solely considers the optimization (3.74) with the cost function (3.32), introducing a weighting matrix in the cost function as in (3.33) may be advantageous in some applications to influence the outcome of the optimization. This does not change the validity of Theorem 3.7 as the weighting matrix does not change the proof.

Remark 3.13. Theorem 3.7 does not require nominal control to be Lyapunov stable to render $\mathrm{Int}(\mathcal{H})$ controlled invariant. However, in order to achieve guaranteed boundedness of the tracking error, a boundedness condition derived from the Lyapunov function of nominal control needs to be included in the optimization, similar to invariance control (3.39).

Remark 3.14. The methods introduced for augmented invariance control in Sec. 3.2.1 may also be used with CBF-based control to increase the relative degree of a constraint function. Introducing an augmentation of the system requires a change of the used CBF but allows for more influence on the design of the resulting dynamics.

This concludes the formal introduction of invariance control and CBF-based control for guaranteed constraint satisfaction. In the following, we provide the results from an experimental evaluation of both methods before turning to the questions of how to satisfy constraints with uncertain parameters and constraints on multi-agent systems.

3.4 Experimental Evaluation

In this section, the developed control schemes are verified in experiments on an anthropomorphic manipulator with seven degrees of freedom ($n_q = 7$). For information on the manipulator, the interested reader is referred to [Sta06].

In general, the goal of introducing a safety control scheme is to avoid damage to the environment and to keep humans in the vicinity safe. Therefore, both invariance control and CBF-based control are evaluated on two Cartesian constraint setups: one which requires the satisfaction of multiple static box constraints representing environmental constraints on the end effector and one which requires the satisfaction of a single dynamic spherical constraint on the end effector representing a human moving in the vicinity. In order to include the effects of physical interaction with humans, a stiffness control law is chosen to represent nominal control. This enables tracking a desired trajectory while allowing a human partner to apply interaction forces to adjust the position of the manipulator. Especially the application of the previously unknown, external disturbance forces illustrates the advantage of the proposed control schemes over collision avoidance approaches such as potential fields [RK92]. Even without prior knowledge of the magnitude or the direction of the applied force, both approaches are able to guarantee constraint satisfaction, whereas potential functions are only able to give such a guarantee if they are designed to absorb all energy from the dynamics and the external forces, which would require prior knowledge.

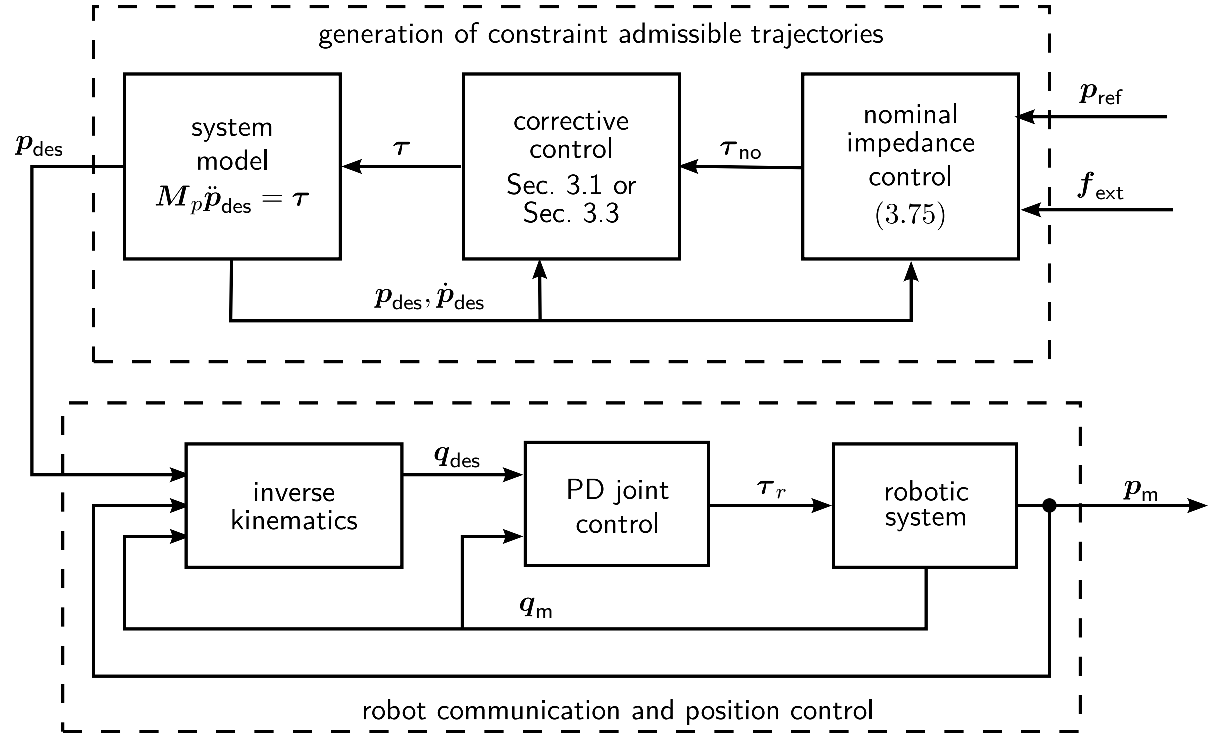


Figure 3.11.: Control structure for the experiments consisting of a control loop for the generation of constraint admissible trajectories and the position-controlled robotic system.

Note that although in the following, constraints are imposed solely on the Cartesian position of the end effector, in general, it is also possible to introduce constraints on joint level, on the entire manipulator or on the respective velocities by defining appropriate constraint functions.

3.4.1 Setup

The control structure used in the experiments is depicted in Fig. 3.11. The robotic system is position controlled by PD joint control in combination with inverse kinematics to derive the desired joint position $\boldsymbol{q}_{\text{des}} \in \mathbb{R}^7$ from a desired Cartesian position $\boldsymbol{p}_{\text{des}} \in \mathbb{R}^3$. This generates the input $\boldsymbol{\tau}_r \in \mathbb{R}^7$. Note that $\boldsymbol{\tau}_r$ does not correspond to the input of a torque-controlled robotic system, which is why it is not possible to use (2.4) to derive corrective control for constraint satisfaction. Instead corrective control, i.e. invariance control or CBF-based control, is included in a preceding control loop, which generates a constraint admissible trajectory from a desired trajectory $\boldsymbol{p}_{\text{ref}} \in \mathbb{R}^3$ and the applied external forces and torques $\boldsymbol{f}_{\text{ext}} \in \mathbb{R}^3$. Corrective control is inserted in between a simplified system model, which neglects Coriolis and gravitational terms as these are compensated by the robotic system, and nominal control. Note that the input error $\boldsymbol{e}_{\text{u}}$ equals zero in this setup as corrective control is included in the preceding control loop, where the parameters are exactly known.

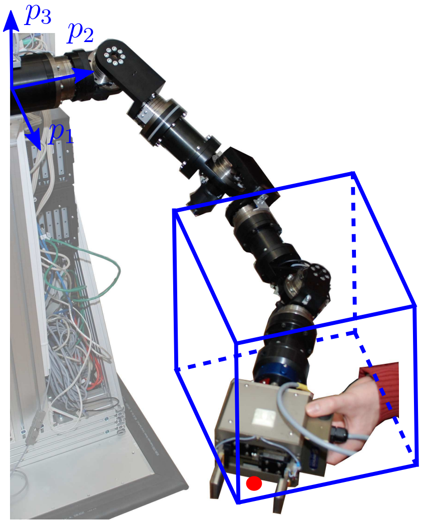


Figure 3.12.: Positioning of the static Cartesian box constraints —, which are imposed on the end effector position • during the first experiment [RKH16].

Nominal Control

The nominal control scheme is impedance control, which achieves the desired compliant reaction of the manipulator to external forces exerted by the human interaction partner. Using task space impedance control (2.24) enables the design of the robot compliance in task space. Note that the use of the simplified robot model without Coriolis and gravitational terms in the setup results in a simplification of (2.24).

$$\boldsymbol{\tau}_{\text{no}} = \boldsymbol{f}_{\text{ext}} + \boldsymbol{M}_p \ddot{\boldsymbol{p}}_{\text{des}} + \boldsymbol{D}_p(\dot{\boldsymbol{p}}_{\text{des}} - \dot{\boldsymbol{p}}_{\text{des}}) + \boldsymbol{K}_p(\boldsymbol{p}_{\text{des}} - \boldsymbol{p}_{\text{des}}) \tag{3.75}$$

A sensible choice of the parameters achieves the desired behavior. The impedance control scheme does, however, not enforce the constraints.

Constraints

Constraints as in (2.11) are conveniently defined in Cartesian space, thus limiting the translational motion of the end effector. In the task at hand, the task space is the translational Cartesian space with the task coordinates $\boldsymbol{p} \in \mathbb{R}^3$. Typical shapes, which are used to model obstacles, are box or spherical constraints as introduced in Examples 2.2 and 2.3. For the evaluation of the proposed control approaches, we use two sets of constraints.

Static Constraints In a first experiment, the desired Cartesian end effector position $\boldsymbol{p}_{\text{des}}$ is constrained by static box constraints, which are a set of six linear constraints as introduced in Example 2.2. The constraint functions are given by

$$y_{\text{c,low},i} = \eta_{i,\min} - p_{i,\text{des}} \tag{3.76}$$

$$y_{\text{c,up},i} = p_{i,\text{des}} - \eta_{i,\max} \tag{3.77}$$

where $i \in \{1, 2, 3\}$ and $\eta_{i,\min}$ and $\eta_{i,\max}$ are the constant lower and upper bound parameters, respectively. The positioning of the constraints is illustrated in Fig. 3.12.

Differentiating (3.76)–(3.77) with respect to time

$$\dot{y}_{\mathrm{c,low},i} = -\dot{p}_{i,\mathrm{des}} \tag{3.78}$$
$$\dot{y}_{\mathrm{c,up},i} = \dot{p}_{i,\mathrm{des}} \tag{3.79}$$
$$\ddot{y}_{\mathrm{c,low},i} = -\ddot{p}_{i,\mathrm{des}} \tag{3.80}$$
$$\ddot{y}_{\mathrm{c,up},i} = \ddot{p}_{i,\mathrm{des}} \tag{3.81}$$

and using the inverted simplified robot model

$$\ddot{\boldsymbol{p}}_{\mathrm{des}} = \boldsymbol{M_p}^{-1}\boldsymbol{\tau} \tag{3.82}$$

yields the relative degree $r = 2$ for each constraint as $\boldsymbol{M_p}^{-1}$ is non-singular since it it invertible.

Dynamic Constraint In a second experiment, a single dynamic spherical constraint is imposed on the end effector position. This constraint is chosen as an exemplary constraint to keep the human hand safe from collisions with the end effector. Naturally, more constraints may be added to account for the entire human body.

The radius $\eta_r \in \mathbb{R}$ of the constraint is set to a constant value, while the Cartesian center position $\boldsymbol{\eta_m} \in \mathbb{R}^3$ varies over time to account for the moving human hand. The constraint output function is then given by

$$y_{\mathrm{c}} = h_{\mathrm{c}}\left(\boldsymbol{p}_{\mathrm{des}}, \begin{bmatrix} \boldsymbol{\eta_m} \\ \eta_r \end{bmatrix}\right) = \eta_r - \|\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m}\|_2 \ . \tag{3.83}$$

Note that the parameters have to be chosen such that the human (or object) that has to be kept safe is contained entirely. Figure 3.13 depicts the experimental setup with the robot and the human, while Fig. 3.14 provides a more detailed schematic view.

Differentiating (3.83) with respect to time

$$\dot{y}_{\mathrm{c}} = -\frac{(\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m})^{\mathsf{T}}(\dot{\boldsymbol{p}}_{\mathrm{des}} - \dot{\boldsymbol{c}}_{\boldsymbol{m}})}{\|\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m}\|_2} \tag{3.84}$$
$$\ddot{y}_{\mathrm{c}} = -\frac{(\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m})^{\mathsf{T}}(\ddot{\boldsymbol{p}}_{\mathrm{des}} - \ddot{\boldsymbol{c}}_{\boldsymbol{m}}) + \|\dot{\boldsymbol{p}}_{\mathrm{des}} - \dot{\boldsymbol{c}}_{\boldsymbol{m}}\|_2^2}{\|\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m}\|_2} + \frac{((\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m})^{\mathsf{T}}(\dot{\boldsymbol{p}}_{\mathrm{des}} - \dot{\boldsymbol{c}}_{\boldsymbol{m}}))^2}{\|\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m}\|_2^3} \tag{3.85}$$

and using (3.82) yields the relative degree $r = 2$ if $\boldsymbol{p}_{\mathrm{des}} \neq \boldsymbol{\eta_m}$ as then $\frac{(\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m})^{\mathsf{T}}}{\|\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m}\|_2}\boldsymbol{M_p}^{-1}$ is non-singular.

Corrective control for the satisfaction of the defined static and dynamic constraints is determined using the constraints, their respective derivative and relative degree.

Invariance Control

Based on the chosen constraint, invariance control derives corrective control according to Sec. 3.1. As the relative degree equals two, the invariance function is given by (3.13). The pseudo input depends on the constraints.

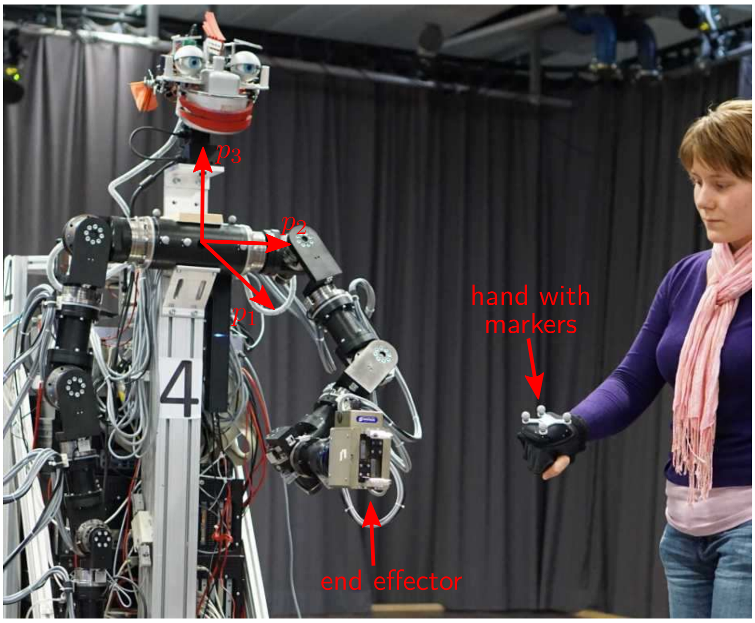


Figure 3.13.: A motion tracking system detects the markers on the hand to determine the center position $\boldsymbol{\eta_m}$, which is defined by the centroid of the marker body [KH15].

Pseudo Input for the Static Constraints Combining (3.80)–(3.81) and (3.82) yields the pseudo input (3.15) for the box constraints

$$\begin{aligned} z_i &= \ddot{y}_{\mathrm{c}} = \boldsymbol{a}_{\mathrm{c},i}^{\intercal}\boldsymbol{\tau} + b_{\mathrm{c},i} \\ \boldsymbol{a}_{\mathrm{c},i}^{\intercal} &= -\boldsymbol{M}_p \begin{bmatrix} d_{i,1} & d_{i,2} & d_{i,3} \end{bmatrix} \quad \forall\ y_{\mathrm{c,low},i} \\ \boldsymbol{a}_{\mathrm{c},i}^{\intercal} &= \boldsymbol{M}_p \begin{bmatrix} d_{i,1} & d_{i,2} & d_{i,3} \end{bmatrix} \quad \forall\ y_{\mathrm{c,up},i} \\ b_{\mathrm{c},i} &= 0 \quad \forall\ y_{\mathrm{c,low},i},\ y_{\mathrm{c,up},i} \end{aligned}$$

with $i \in \{1, 2, 3\}$ and

$$\boldsymbol{d}_{i,j} = \begin{cases} 0 & \text{if } j \neq i \\ 1 & \text{if } j = i\ . \end{cases} \tag{3.86}$$

Pseudo Input for the Dynamic Constraint Combining (3.85) and (3.82) yields the pseudo input (3.15) for the dynamic constraint

$$\begin{aligned} z &= \ddot{y}_{\mathrm{c}} = \boldsymbol{a}_{\mathrm{c}}^{\intercal}\boldsymbol{\tau} + b_{\mathrm{c}} \\ \boldsymbol{a}_{\mathrm{c}}^{\intercal} &= -\frac{(\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m})^{\intercal}}{\|\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m}\|_2}\boldsymbol{M_p}^{-1} \\ b_{\mathrm{c}} &= \frac{(\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m})^{\intercal}\ddot{\boldsymbol{\eta}}_{\boldsymbol{m}}}{\|\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m}\|_2} + \frac{\|\dot{\boldsymbol{p}}_{\mathrm{des}} - \dot{\boldsymbol{\eta}}_{\boldsymbol{m}}\|_2^2}{\|\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m}\|_2} + \frac{((\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m})^{\intercal}(\dot{\boldsymbol{p}}_{\mathrm{des}} - \dot{\boldsymbol{\eta}}_{\boldsymbol{m}}))^2}{\|\boldsymbol{p}_{\mathrm{des}} - \boldsymbol{\eta_m}\|_2^3}\ . \end{aligned}$$

Corrective Control The corrective pseudo input is determined by (3.22). The corrective control input is calculated using the minimization (3.31) with the cost function (3.32). Note that for the first experiment with the static constraints, maximally three constraints are

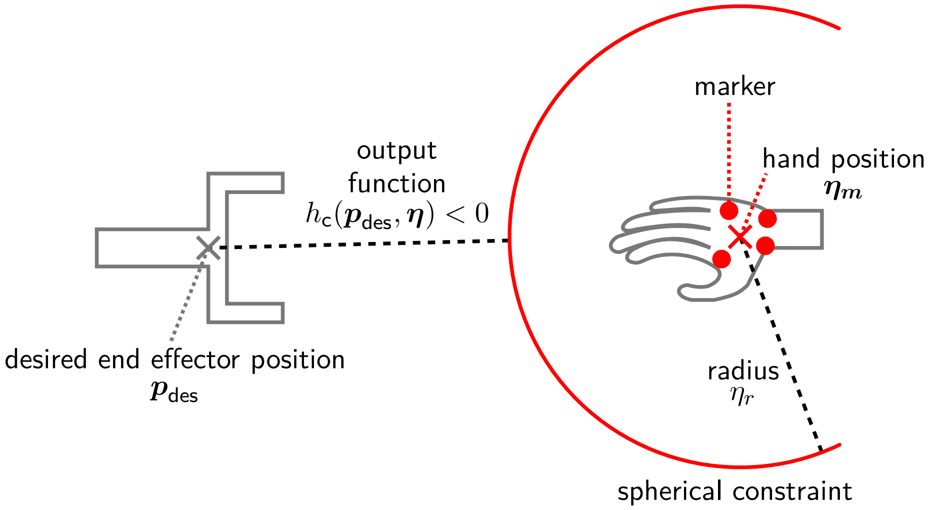

marker
hand position
$\boldsymbol{\eta_m}$
output
function
$h_\mathsf{c}(\boldsymbol{p}_\mathsf{des}, \boldsymbol{\eta}) < 0$
desired end effector position
$\boldsymbol{p}_\mathsf{des}$
radius
η_r
spherical constraint

Figure 3.14.: The hand position determines the center $\boldsymbol{\eta_m}$ of the spherical constraint with the constant radius η_r. The output function $h(\boldsymbol{p}_{\text{des}}, \boldsymbol{\eta})$ is a measure for the distance to the end effector [KH15].

active at the same time while the second experiment includes a single constraint. Therefore, determining the set $\mathcal{I}$ (3.37) is trivial and (3.38) with $\boldsymbol{W} = \boldsymbol{I}_3$ may be used to determine corrective control. Constraint satisfaction is then guaranteed by Theorem 3.1. Note that the additional condition for boundedness from Theorem 3.3 is not included in the experimental evaluation. Similar to Example 3.7, the combination of the constraints with nominal stiffness control yields a naturally bounded system, which is also true for the CBF-based control.

CBF-based Control

CBF-based control derives corrective control according to Sec. 3.3. Keep in mind that for the CBFs use an inverse constraint definition $y_\text{B} = -y_\text{c}$. As the relative degree of the constraints equals two, we use the CBF from (3.72) in the experiments. Note that the parameters ζ_a and ζ_b related to this CBF may differ for each constraint, i.e. in a setup with multiple constraints, each constraint may be assigned a different set of parameters.

Set of Admissible Control Values The derivation of the set of admissible control values (3.73) requires the differentiation of B_i with respect to time.

$$\dot{B}_i = \frac{\partial B_i}{\partial y_\text{B}} \dot{y}_\text{B} + \frac{\partial B_i}{\partial \dot{y}_\text{B}} \ddot{y}_\text{B} = -\frac{1}{y_\text{c}(1 - y_\text{c})} \dot{y}_\text{c} + 2\zeta_a \zeta_b \frac{\dot{y}_\text{c}}{(1 + \zeta_b \dot{y}_\text{c}^2)^2} \ddot{y}_\text{c}$$

The derivative depends on the constraint functions and their respective derivatives as derived in (3.76)–(3.85). The set of admissible control values

$$\left\{ \boldsymbol{\tau} \in \mathbb{R}^3 \mid \boldsymbol{a}_{\text{B},i}^\intercal \boldsymbol{\tau} + b_{\text{B},i} \leq \frac{\mu_i}{B_i}, \forall i \in \mathcal{B} \right\}$$

is then determined for the ***static constraints*** by

$$
\begin{aligned}
\boldsymbol{a}_{\mathrm{B},i}^{\mathsf{T}} &= 2\zeta_a\zeta_b \frac{\dot{y}_{\mathrm{c,low},i}}{(1+\zeta_b\dot{y}_{\mathrm{c,low},i}^2)^2}(-\boldsymbol{M}_p)\begin{bmatrix} d_{i,1} & d_{i,2} & d_{i,3}\end{bmatrix} \quad \forall\, y_{\mathrm{c,low},i}\\
\boldsymbol{a}_{\mathrm{B},i}^{\mathsf{T}} &= 2\zeta_a\zeta_b \frac{\dot{y}_{\mathrm{c,up},i}}{(1+\zeta_b\dot{y}_{\mathrm{c,up},i}^2)^2}\boldsymbol{M}_p\begin{bmatrix} d_{i,1} & d_{i,2} & d_{i,3}\end{bmatrix} \quad \forall\, y_{\mathrm{c,up},i}\\
b_{\mathrm{B},i} &= -\frac{1}{y_{\mathrm{c},i}(1-y_{\mathrm{c},i})}\dot{y}_{\mathrm{c},i} \quad \forall\, \dot{y}_{\mathrm{c},i} = y_{\mathrm{c,low},i},\ y_{\mathrm{c,up},i}
\end{aligned}
$$

with $i \in \{1,2,3\}$ and $d_{i,j}$ as in (3.86) and for the ***dynamic constraint*** by

$$
\begin{aligned}
\boldsymbol{a}_{\mathrm{B}}^{\mathsf{T}} &= 2\zeta_a\zeta_b \frac{\dot{y}_{\mathrm{c}}}{(1+\zeta_b\dot{y}_{\mathrm{c}}^2)^2}\left(-\frac{(\boldsymbol{p}_{\mathrm{des}}-\boldsymbol{\eta_m})^{\mathsf{T}}}{\|\boldsymbol{p}_{\mathrm{des}}-\boldsymbol{\eta_m}\|_2}\boldsymbol{M_p}^{-1}\right)\\
b_{\mathrm{B}} &= -\frac{1}{y_{\mathrm{c}}(1-y_{\mathrm{c}})}\dot{y}_{\mathrm{c}} + 2\zeta_a\zeta_b\frac{\dot{y}_{\mathrm{c}}}{(1+\zeta_b\dot{y}_{\mathrm{c}}^2)^2}\cdot\\
&\quad \left(\frac{(\boldsymbol{p}_{\mathrm{des}}-\boldsymbol{\eta_m})^{\mathsf{T}}\ddot{\boldsymbol{\eta}}_{\boldsymbol{m}}}{\|\boldsymbol{p}_{\mathrm{des}}-\boldsymbol{\eta_m}\|_2} + \frac{\|\dot{\boldsymbol{p}}_{\mathrm{des}}-\dot{\boldsymbol{\eta}}_{\boldsymbol{m}}\|_2^2}{\|\boldsymbol{p}_{\mathrm{des}}-\boldsymbol{\eta_m}\|_2} + \frac{((\boldsymbol{p}_{\mathrm{des}}-\boldsymbol{\eta_m})^{\mathsf{T}}(\dot{\boldsymbol{p}}_{\mathrm{des}}-\dot{\boldsymbol{\eta}}_{\boldsymbol{m}}))^2}{\|\boldsymbol{p}_{\mathrm{des}}-\boldsymbol{\eta_m}\|_2^3}\right).
\end{aligned}
$$

Corrective Control The corrective control input is calculated using (3.74). Constraint satisfaction is then guaranteed by Theorem 3.6 and Theorem 3.7.

3.4.2 Implementation

The control structure is implemented on a Linux system with a preemptive real-time kernel using the *Matlab/Simulink Real-Time Workshop*. The code runs with a sampling rate of 1 kHz. The real time QP solver used in CBF-based control is generated by CVXGEN [MB11].

Closed-loop differential inverse kinematics determine the position of the end effector from joint position encoders with 1024 ticks per motor revolution. Backlash-free interaction is ensured by *Harmonic Drive* gears with transmission ratios of 1 : 100 in shoulder and elbow joints (J1 to J4) and 1 : 160 in the wrist joints (J5 to J7). The external forces $\boldsymbol{f}_{\mathrm{ext}}$, applied to the end effector by the interacting human, are measured by a *JR3 sensor*, which senses forces and torques with 6 degrees of freedom.

The center position of the dynamic constraint is determined by the position of the human hand, which requires the tracking of human body parts, e.g. by a vision-based perception system. Here, for the sake of demonstration, we employ our marker-based *Qualisys Motion Tracking System* to track and avoid the collision with a human hand. The hand is marked with a rigid body, the centroid of which defines the center position $\boldsymbol{\eta_m}$ of the spherical constraint (3.83). The value of $\boldsymbol{\eta_m}$ and its time derivatives up to the second order are derived in real-time from the data collected using the motion tracking system. The constant radius η_r of the spherical constraint is chosen such that the sphere encloses the human's hand. The system and control parameters are provided in Tab. 3.3.

Table 3.3.: Model and control parameters in the experimental evaluation of invariance and CBF-based control for constraint satisfaction.

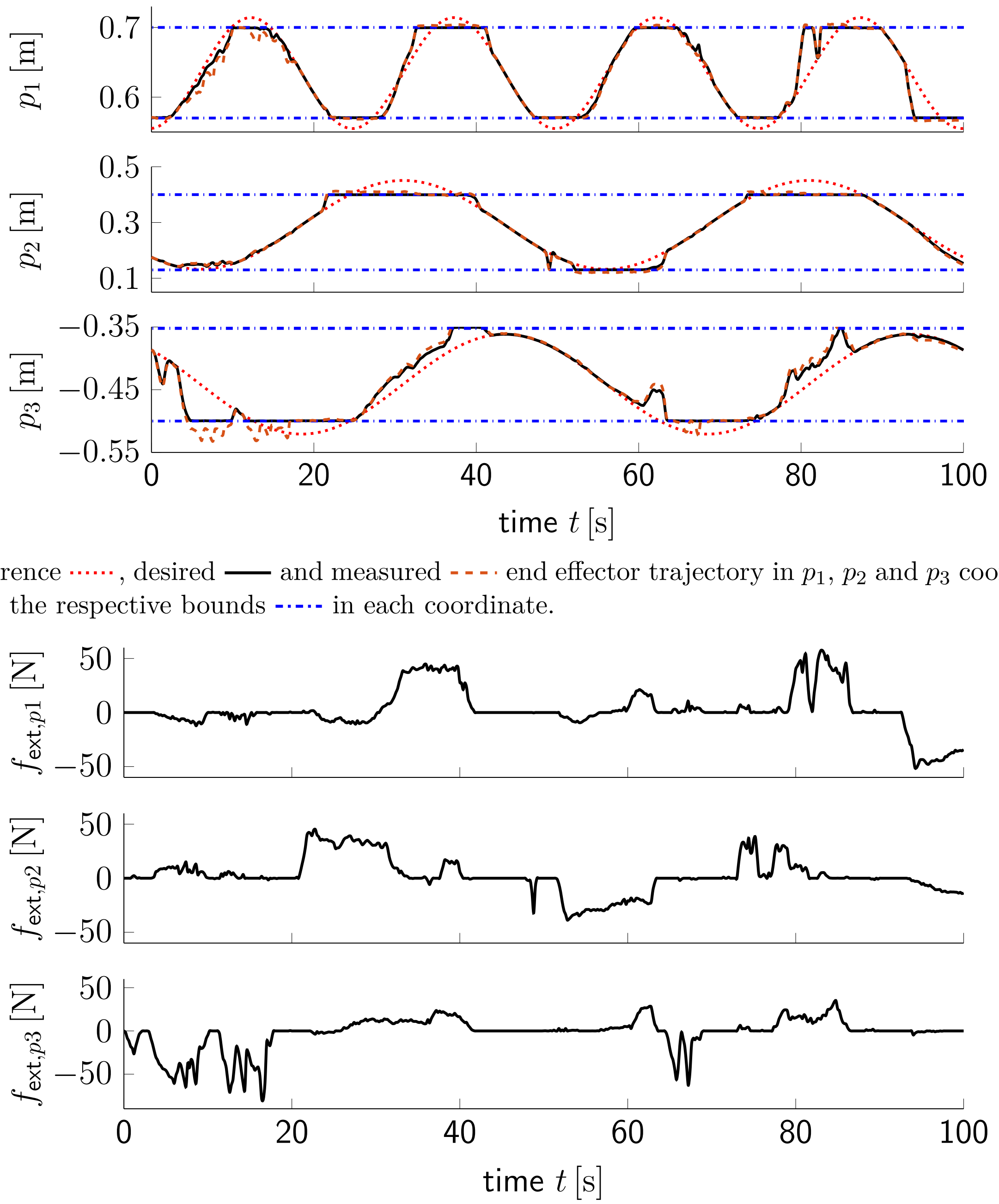
General		
Sampling time	T_A	$0.001\ \mathrm{s}$
Robot mass	$\boldsymbol{M}_p$	$\begin{bmatrix} 10 & & \\ & 10 & \\ & & 10 \end{bmatrix}\ \mathrm{kg}$
Desired position (dynamic constraints)	$\boldsymbol{p}_{\mathrm{des}}$	$[0.635\ , 0.133\ , -0.441]^\intercal\ \mathrm{m}$
Nominal control		
Cartesian stiffness (CBF)	$\boldsymbol{K}_p$	$200 \cdot \boldsymbol{I}_3\ \mathrm{N/m}$
Cartesian stiffness (invariance)	$\boldsymbol{K}_p$	$600 \cdot \boldsymbol{I}_3\ \mathrm{N/m}$
Cartesian damping	$\boldsymbol{D}_p$	$80 \cdot \boldsymbol{I}_3\ \mathrm{N\,s/m}$
Static constraints		
Upper bounds	$\boldsymbol{\eta}_{\max}$	$[0.70\ 0.40\ -0.35]^\intercal\ \mathrm{m}$
Lower bounds	$\boldsymbol{\eta}_{\min}$	$[0.57\ 0.13\ -0.50]^\intercal\ \mathrm{m}$
Dynamic constraint		
Radius (CBF)	η_r	$0.25\ \mathrm{m}$
Radius (invariance)	η_r	$0.4\ \mathrm{m}$
Invariance control		
Control parameter	γ	$-18\,\mathrm{m/s^2}$
CBF-based control		
Static constraints	μ	$10\ 1/\mathrm{s}$
	ζ_a	100
	ζ_b	10
Dynamic constraint	μ	$10\ 1/\mathrm{s}$
	ζ_a	1
	ζ_b	1

3.4.3 Results

This section presents the experimental results generated by invariance control with static box constraints [Kim11; KLH12] and a dynamic constraint [KH15; KH17] as well as the results generated by CBF-based control with static and dynamic constraints [RKH16].

Static Constraints

In the first experiments with the static constraints, the end effector is controlled to track a reference trajectory while a human applies forces of random direction and magnitude to the end effector that lead to a deviation from the reference trajectory. The corrective control schemes ensure that even though the reference trajectory may violate the constraints and even though the forces might push the end effector towards the bounds, the end effector does not violate any constraints.



(a) Reference ⋯⋯, desired —— and measured - - - - end effector trajectory in p_1, p_2 and p_3 coordinates with the respective bounds -·-·- in each coordinate.

(b) Measured external forces $\boldsymbol{f}_{\mathrm{ext}} = [f_{\mathrm{ext},p1}\ f_{\mathrm{ext},p2}\ f_{\mathrm{ext},p3}]^\intercal$ applied in each Cartesian direction.

Figure 3.15.: Experimental evaluation of invariance control with static bounds: (a) Cartesian end effector trajectories with the respective bounds and (b) external forces applied by the human in each Cartesian direction during the experiment.

Invariance Control Figure 3.15a depicts the generated desired trajectory and the measured trajectory of the end effector in all three Cartesian directions. It may be observed that the desired trajectory is always compliant with the boundaries even though the original reference violates the constraints as e.g. in p_2 for $t \in [25, 38]$ s, which confirms Theorem 3.1. The forces depicted in Fig. 3.15b cause deviations from the reference trajectory in the direction of the force as e.g. in p_1 for $t \in [80, 82]$ s but do not cause the desired trajectory to violate the constraints. It should, however, be noted that the measured position of the end effector does violate the constraints, especially in the p_3-coordinate at $t \in [5, 18]\,\mathrm{s}$. This is due to the fact that the counterforces generated by the manipulator may not be arbitrarily high. Therefore, if the force, as shown in Fig. 3.15b, reaches a high magnitude, the end effector is not able to follow the desired trajectory and violates the constraint. However, a reduction of the force immediately leads to a return of the measured end effector position into the admissible set as e.g. at $t = 18\,\mathrm{s}$. It may further be observed, that the measured end effector position exactly follows the reference trajectory whenever it satisfies the constraints and no forces are applied, which is due to the nominal stiffness control.

CBF-based Control Figure 3.16 depicts the Cartesian position of the end effector and the measured external forces during the first experiment with six static constraints and CBF-based control. During the first few seconds, the desired position follows the reference trajectory as the end effector is away from the bounds and no forces are applied. If the end effector is at a distance from the constraint and an external force is applied, as in p_3 for $t \in [25, 45]\,\mathrm{s}$, or if the system is close to a constraint and the force is directed away from the limit, as in p_3 for $t \in [5, 10]\,\mathrm{s}$, the end effector gives way to the external force. This shows the effect of the nominal impedance control law.

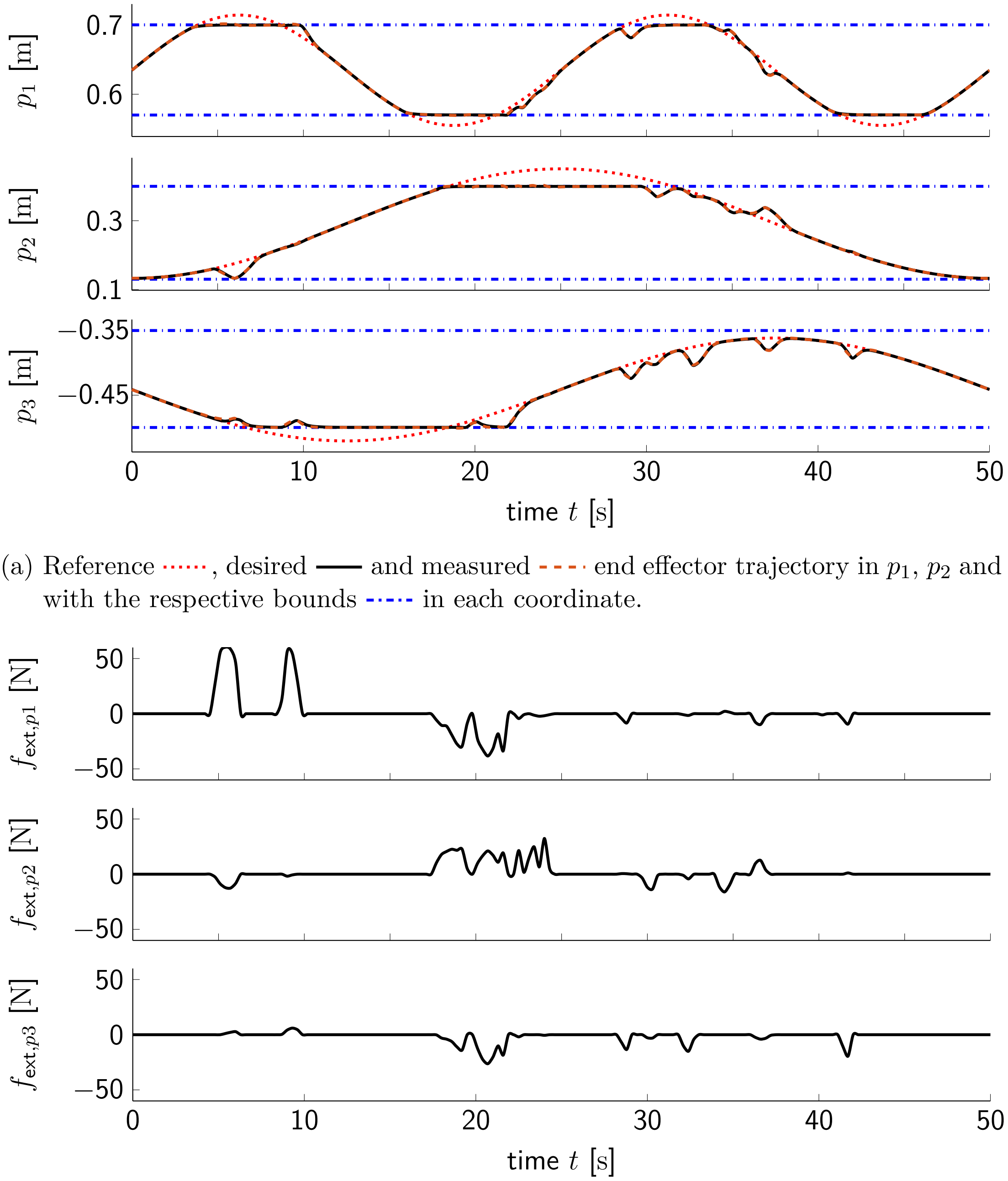
Figure 3.16a also shows that the desired trajectory never violates a constraint even though the reference trajectory does not satisfy the constraints, as in p_1 for $t \in [3, 9]\,\mathrm{s}$. In addition, when the trajectory is close to a boundary and forces are applied pushing the end effector towards the limit, as in p_1 and p_2 for $t \in [15, 25]\,\mathrm{s}$, no violation occurs. Hence the controller enforces the constraints in the presence of external forces and even if the reference leaves the admissible set of states, which illustrates Theorems 3.6 and 3.7.

It may further be observed that the measured position of the end effector $\boldsymbol{p}_\mathrm{m}$ almost exactly coincides with the desired position. However, the measured position shows some minor violations ($< 1\,\mathrm{mm}$) of the constraint, similar to the experiments with invariance control due to the actuators not being able to generate high enough counterforces. Note that the violations are smaller for this experiment in comparison to the experiment with invariance control as the applied external forces depicted in Fig. 3.16b are smaller than in the previous experiment, see Fig. 3.15b. The slight violations do, however, not challenge the general feasibility of both approaches as they are solely due to hardware limitations.

Dynamic Constraint

In the second experiment with the dynamic constraint, the end effector tries to hold a static end effector position $\boldsymbol{p}_\mathrm{ref}$ and there is no physical contact between human and robot, i.e. no external forces. The marked hand then approaches the end effector until an evading motion is carried out by the robot. For invariance control, a second trial is executed, in which a physical coupling between a human and the end effector is introduced, meaning that a second human firmly grasps the end effector and exerts forces to move the end effector in arbitrary directions. The end effector reacts compliantly to exerted forces due to the nominal impedance control law, while still satisfying the constraint.

Note that since the control law only depends on the motion measurements but not on any model of the human movements, the results for different human subjects resemble one another apart from different humans choosing different trajectories. Therefore, we only show



(a) Reference ⋯⋯, desired —— and measured - - - - end effector trajectory in p_1, p_2 and p_3 coordinates with the respective bounds -·-·- in each coordinate.

(b) Measured external forces $\boldsymbol{f}_{\mathrm{ext}} = [f_{\mathrm{ext},p1}\ f_{\mathrm{ext},p2}\ f_{\mathrm{ext},p3}]^{\intercal}$ applied in each Cartesian direction.

Figure 3.16.: Experimental evaluation of CBF-based control with static bounds: (a) Cartesian end effector trajectories with the respective bounds and (b) external forces applied by the human in each Cartesian direction during the experiment.

the results generated with one human.

Invariance Control The experimental data is evaluated with respect to constraint satisfaction, i.e. invariance, and boundedness of the tracking error. The results obtained without external forces are illustrated in Fig. 3.17. The figure shows the value of the invariance function as well as the deviation of the end effector position from the reference position $\boldsymbol{p}_{\mathrm{ref}}$.

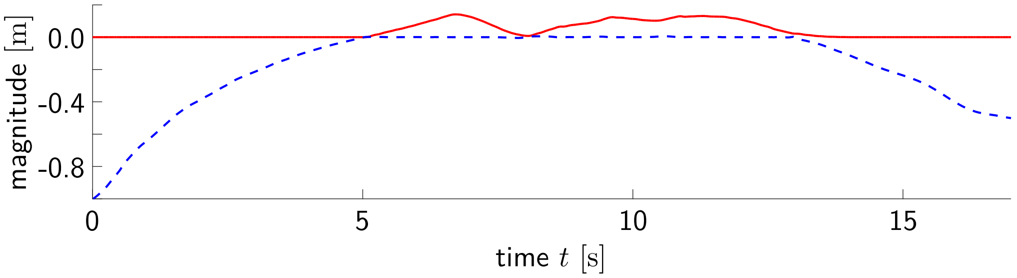
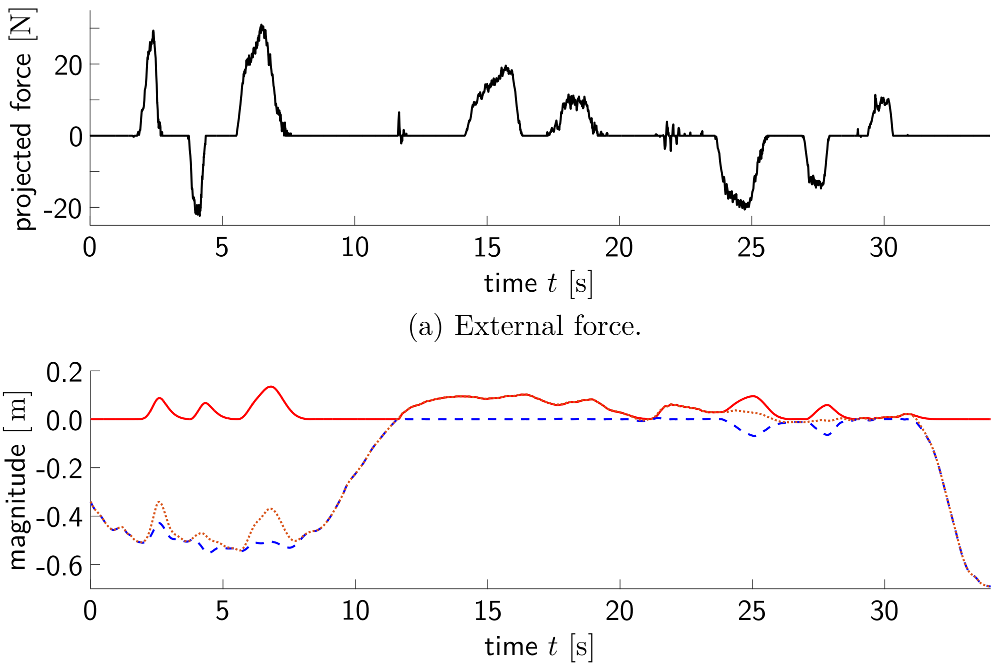


Figure 3.17.: Evaluation of the - - - - invariance function and the —— deviation $\|\boldsymbol{p}_{\mathrm{ref}} - \boldsymbol{p}_{\mathrm{m}}\|_2$ from the reference position $\boldsymbol{p}_{\mathrm{ref}}$ without external forces.

It may be observed that a bounded deviation from the desired position occurs when the invariance function is reduced to zero, i.e. for $t \in [5, 13]$ s. Since the invariance function by definition equals zero when the system approaches the constraint, it is clear that a deviation has to occur in order to avoid a violation of the constraint. Apart from a slight chattering effect at zero, which results from the sampled time implementation of the continuous control scheme from Sec. 3.1, the invariance function is never positive, which illustrates the invariance of the controlled system according to Theorem 3.1. It may further by observed for $t > 13$ s that a removal of the constraint, i.e. a motion of the human away from the end effector, allows the end effector to return to the reference position.

Then, the physical contact with a human is established which results in external forces different from zero. Figure 3.18a shows the fraction of the applied force, which is directed towards the constraint. The bounded deviation of the end effector from $\boldsymbol{p}_{\mathrm{ref}}$ as shown in Fig. 3.18b is caused by the applied forces as well as by the approaching constraint. During the first 10 s, the distance between the end effector and the constraint is large and the invariance function Φ is negative, i.e. the deviation is solely caused by the force. Naturally, the forces affect the value of Φ, since the output function (3.83) and Φ depend on the end effector position. The figure shows that Φ solely takes non-positive values, meaning that even in the presence of external forces, the robotic system is kept invariant by the control scheme and the constraint is not violated. For $\Phi = 0$, the system is at the boundary of the invariant set and the constraint is directly at the end effector, which is illustrated by the fact that the lines of the deviation and the relative constraint position coincide. It may further by observed that at this point the system reacts on the one hand compliantly to forces directed away from the constraint as for example at $t = 25$ s, which decrease the value of Φ and move the end effector away from the constraint. On the other hand, forces directed towards the constraint do not lead to a violation of the constraint or a positive value of Φ. In this case, the system is stiff and Φ keeps its value of zero as for example at $t = 15$ s. This emphasizes the fact, that invariance control achieves invariance and constraint satisfaction as stated in Theorem 3.1. Note that since the applied forces and the constraint motion are bounded, the deviation of the end effector from the reference position is bounded throughout the experiment.

CBF-based Control Figure 3.19 shows the results of CBF-based control with a dynamic constraint, a static reference position and no external forces. During the first few seconds, the constraint is at a distance from the end effector and the end effector holds the desired position, i.e. there is no deviation as depicted in the figure. However, if the constraint moves



(a) External force.

(b) The —— deviation $\|\boldsymbol{p}_{\mathrm{ref}} - \boldsymbol{p}_{\mathrm{m}}\|_2$, the ⋯⋯ relative constraint position $\|\boldsymbol{p}_{\mathrm{ref}} - \boldsymbol{p}_{\mathrm{m}}\|_2 + y_{\mathrm{c}}$ and the - - - - invariance function value.

Figure 3.18.: Influence of (a) the magnitude of the applied external forces in direction of the constraint (positive for application in direction of the constraint) on (b) the deviation from the reference position, the relative position of the constraint with respect to the end effector and the value of the invariance function Φ.

towards the end effector as for $t \in [2, 4]\,\mathrm{s}$, this means that $-\|\boldsymbol{p}_{\mathrm{ref}} - \boldsymbol{p}_{\mathrm{m}}\|_2 - y_{\mathrm{c}}$ decreases in value and a deviation from the reference position occurs as the system has to move to avoid a violation of the constraint. The almost coinciding lines corresponding to the deviation and the relative distance show that the end effector only deviates as much from the desired position as necessary to avoid a violation. This is also emphasized by the constraint function approaching zero, which corresponds to the distance between the end effector and the constraint. It may further be observed that the barrier constraint function y_{B} always has a positive value, which shows constraint satisfaction and illustrates the invariance stated in Theorems 3.6 and 3.7. Note that the constraint function never reaches zero but remains at positive values. This is caused by the characteristic of the CBF approaching infinity towards the constraints, resembling the behavior of potential functions, which absorb the energy of the system until it comes to a rest at a small distance from the bound.

3.5 Discussion

Concluding this chapter, we provide a short summary of the capabilities of the developed I/O-linearization-based control approaches for constraint satisfaction.

Both invariance control and CBF-based control combine corrective control with nominal control using convex minimization

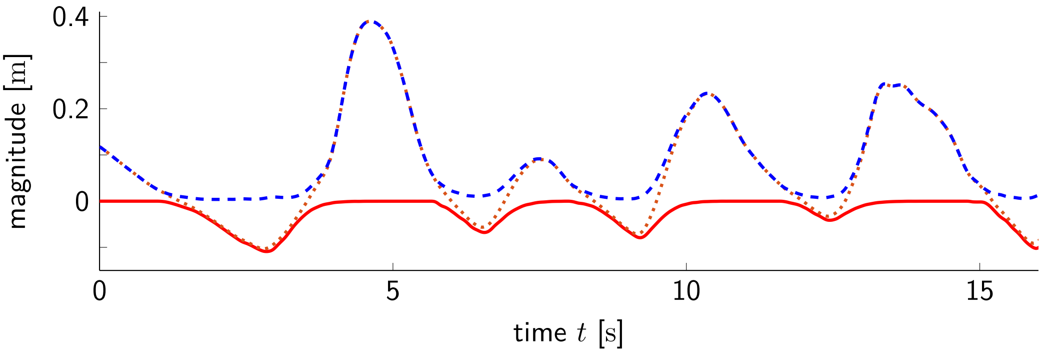


Figure 3.19.: Reaction of the end effector to the moving constraint with — the negative deviation of the end effector from the reference position $-\|\boldsymbol{p}_\mathrm{m} - \boldsymbol{p}_\mathrm{ref}\|_2$, ⋯⋯ the relative distance between the surface of the sphere and the reference position $-\|\boldsymbol{p}_\mathrm{ref} - \boldsymbol{p}_\mathrm{m}\|_2 - y_\mathrm{c}$ and - - - - the barrier constraint function $y_\mathrm{B} = -y_\mathrm{c}$.

$$\begin{aligned} \boldsymbol{u}_\mathrm{c} = \underset{\boldsymbol{u}}{\operatorname{argmin}} \; & \|\boldsymbol{u}_\mathrm{c} - \boldsymbol{u}_\mathrm{no}\|_2^2 \\ \text{s.t. } \; & \boldsymbol{a}_i^\intercal \boldsymbol{u} + b_i \leq d_i \quad \forall\, i \in \tilde{\mathcal{B}} \end{aligned} \tag{3.87}$$

with $\boldsymbol{a}_i = \boldsymbol{a}_{\mathrm{c},i}$, $b_i = b_{\mathrm{c},i}$ from (3.1), $\tilde{\mathcal{B}} = \mathcal{B}_\mathrm{act}$ and $d_i = z_{\mathrm{c},i}$ for invariance control according to Sec. 3.1 and $\boldsymbol{a}_i = \boldsymbol{a}_{\mathrm{B},i}$, $b_i = b_{\mathrm{B},i}$ from (3.70), $\tilde{\mathcal{B}} = \mathcal{B}$ and $d_i = \frac{\mu_i}{B_i(\boldsymbol{x},\boldsymbol{x}_\eta)}$ for CBF-based control from Sec. 3.3.

This allows the system to follow the given task at a distance from the constraints, while satisfying any static and dynamic constraints in joint or work space. The approaches are designed to render a subset of the admissible set controlled positively invariant thus being able to guarantee constraint satisfaction.

Remark 3.15. Since both invariance control and CBF-based control generate similar conditions on the control input, the approches may be combined, i.e. some constraints may be enforced using invariance control while the others are enforced using CBFs.

The differences between invariance control and CBF-based control lie in the way how corrective control is determined. Invariance control relies on switching between nominal and corrective control whenever the invariance function, which is designed based on the constraint dynamics, indicates the necessity of corrective action for constraint satisfaction. The invariance function and the generated optimization conditions are defined in the entire state space. This leads to a control input that allows the state to re-enter the admissible set in case of constraint violations, which may for example occur if new constraints are added during runtime. In sampled implementations, however, the switching induces chattering effects which may lead to a decreased performance. This effect is reduced by using the approaches presented in Sec. 3.2.

CBF-based control, on the other hand, avoids any discontinuities in the control input by generating a Lipschitz-continuous control law. However, as the CBFs only have a defined function value within the set of admissible states by Def. 3.11, CBF-based control is not able to resolve constraint violations, caused by inadmissible initial state values or new constraints added during runtime. In addition, invariance control stops the state directly at the constraint, while CBF-based control stops it at a small distance when the entire energy is absorbed, which means that it is not able to exploit the entire admissible set.

Further note that both approaches assume the system to be able to handle arbitrarily high input values. Even though input constraints may be added to the minimization, this is not recommended as it may lead to infeasibility of the minimization (3.87). Instead, it is preferable to adjust the control parameters γ and μ to reduce the required control input.

The characteristics and capabilities of the proposed control approaches are illustrated in the conducted experiments. The presented results encourage the application of the presented I/O-linearization-based control approaches to robotic systems in scenarios involving humans and physical interaction with humans. Note that the proposed control schemes may also be applied to enforce constraints on the orientation as demonstrated for invariance control in [KLH12].

However, as both control approaches introduced in this chapter rely on I/O-linearization to determine the influence of the input on the constraints, they require an exact model of system and constraints, which means they are susceptible to uncertainties in the constraint parameters $\boldsymbol{\eta}$. The following chapter deals with this issue and presents robust and probabilistic approaches which enable the satisfaction of constraints with uncertain parameters.

4

Satisfaction of Constraints with Uncertain Parameters

The reliable enforcement of the restrictions imposed on robots by closely interacting humans depends strongly on the accurate estimation of the human pose, motion and intent. However, any variability in the human motion causes uncertainties, especially in combination with an inexact model and any offsets in measurements. This is also true if the I/O-linearization is derived by using learning-based methods [UBK+17]. These uncertainties need to be accounted for when designing a control scheme, that aims at guaranteeing constraint satisfaction.

Related Work and Open Problems

Optimization-based control schemes such as MPC [MRR+00] handle uncertain constraints by adjusting the constrained optimization. While robust approaches guarantee constraint satisfaction for every instance of the uncertainty, both chance-constrained and scenario-based approaches provide a satisfaction probability [CC05]. Including uncertainties in set-based control approaches [Bla99; AFG+14] is possible by computing robust sets [FAC10]. As the introduction of uncertainties may however lead to an increased computational effort, the solution of the optimization for MPC and the computation of the invariant sets may become too costly for real-time applications with high sampling rates. In [BG10; MB99], preliminary steps towards invariance control with uncertain parameters are taken, but only for SISO-systems and without offering a systematic approach to choosing the invariant set and the control input for robust and probabilistic guarantees.

This chapter introduces I/O-linearization-based control approaches for the satisfaction of constraints with uncertain parameters based on the results for invariance control as presented in [KPHed]. If the uncertainty distribution is continuous, there are infinitely many instances of the uncertainty. This means that the optimization problem (3.87) is subject to infinitely many optimization constraints as each instance of the uncertainty generates one condition on the input. As a result, directly solving the optimization in real-time is hard if not impossible [CC05]. In order to resolve this issue, we present three alternative control methods: robust, probabilistic and scenario-based constraint satisfaction. Note that the methods preserve the characteristic of the derived control law being implemented in addition to any nominal control scheme. Furthermore the control types are shown to formally guarantee robust or probabilistic constraint satisfaction and are interchangeable, thus allowing for different constraint types to be straightforwardly combined.

Remark 4.1. With each method, we associate a set containing the indices of the constraints which are desired to be modeled in such a way: $\mathcal{B}_{\text{rob}}$ contains the constraints for robust constraint satisfaction, $\mathcal{B}_{\text{pr}}$ those for probabilistic constraint satisfaction and $\mathcal{B}_{\text{sc}}$ those for the scenario-based approach.

4.1 Robust Constraint Satisfaction

For the sake of being able to achieve robust constraint satisfaction, we make the following assumption in this section.

Assumption 4.1. *The set $\mathcal{D}$ from* (2.18) *is bounded, i.e. $\boldsymbol{\Delta_\eta}$ is bounded with $\boldsymbol{\Delta_\eta} \in \mathcal{D} = \mathcal{D}_{rob}$ and the bounded set $\mathcal{D}_{rob} \subset \mathbb{R}^{n_\eta(r_{\max}+1)}$. Furthermore, the bounds are known.*

If the uncertainties are bounded and guaranteed satisfaction of constraints $i \in \mathcal{B}_{\text{rob}} \subseteq \mathcal{B}$ is desired, the control goal should be to include every instance of the uncertainty in the determination of the corrective control input in order to achieve robust satisfaction.

Definition 4.1. A control scheme achieves ***robust satisfaction*** of constraints with uncertain parameters, if constraint violation is avoided for any value of the uncertainty, i.e. if the state remains in the ***robust admissible set***.

$$\mathcal{H}_{\text{rob}}(\overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) = \{\boldsymbol{x} \in \mathbb{R}^n \mid h_{\text{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_\eta) \leq 0 \ , \quad \forall i \in \mathcal{B}_{\text{rob}}, \boldsymbol{\Delta}_\eta \in \mathcal{D}_{\text{rob}}\}$$

Examining the effect of the bounded uncertainty on the constraint functions (2.11) and their derivatives yields

$$y_{\text{c},i} = h_{\text{c},i}(\boldsymbol{x}, \boldsymbol{x_\eta}) = h_{\text{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}) + \Delta_{h_{\text{c},i}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}, \boldsymbol{\Delta_\eta}) \tag{4.1}$$

$$\dot{y}_{\text{c},i} = \dot{h}_{\text{c},i}(\boldsymbol{x}, \boldsymbol{x_\eta}) = \dot{h}_{\text{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}) + \Delta_{\dot{h}_{\text{c},i}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}, \boldsymbol{\Delta_\eta}) \tag{4.2}$$

$$\vdots$$

$$y_{\text{c},i}^{(r_i)} = (\boldsymbol{a}_{\text{c},i}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}) + \boldsymbol{\Delta}_{\boldsymbol{a}_{\text{c},i}}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}, \boldsymbol{\Delta}_\eta))\boldsymbol{u} + b_{\text{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}) + \Delta_{b_{\text{c},i}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}, \boldsymbol{\Delta_\eta}) \tag{4.3}$$

with $\boldsymbol{a}_{\text{c},i}^{\mathsf{T}} = [a_{\text{c},i,1}, \dots, a_{\text{c},i,m}]$ and

$$\Delta_{h_{\text{c},i}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}, \boldsymbol{\Delta_\eta}) = h_{\text{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}} + \boldsymbol{\Delta_\eta}) - h_{\text{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}) \tag{4.4}$$

$$\Delta_{\dot{h}_{\text{c},i}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}, \boldsymbol{\Delta_\eta}) = \dot{h}_{\text{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}} + \boldsymbol{\Delta_\eta}) - \dot{h}_{\text{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}) \tag{4.5}$$

$$\vdots$$

$$\boldsymbol{\Delta}_{\boldsymbol{a}_{\text{c},i}}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}, \boldsymbol{\Delta}_\eta) = \boldsymbol{a}_{\text{c},i}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}} + \boldsymbol{\Delta}_\eta) - \boldsymbol{a}_{\text{c},i}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}) \tag{4.6}$$

$$\Delta_{b_{\text{c},i}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}, \boldsymbol{\Delta_\eta}) = b_{\text{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}} + \boldsymbol{\Delta_\eta}) - b_{\text{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}) \tag{4.7}$$

for the I/O-linearization of the constraint outputs required for invariance control. Similarly, the control barrier functions (CBFs) and their respective derivatives may be written as

$$B_i(\boldsymbol{x}, \boldsymbol{x_\eta}) = B_i(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}) + \Delta_{B_i}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}, \boldsymbol{\Delta_\eta}) \tag{4.8}$$

$$\dot{B}_i(\boldsymbol{x}, \boldsymbol{x_\eta}, \boldsymbol{u}) = (\boldsymbol{a}_{\text{B},i}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}) + \boldsymbol{\Delta}_{\boldsymbol{a}_{\text{B},i}}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}, \boldsymbol{\Delta}_\eta))\boldsymbol{u} + b_{\text{B},i}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}) + \Delta_{b_{\text{B},i}}(\boldsymbol{x}, \overline{\boldsymbol{x_\eta}}, \boldsymbol{\Delta_\eta}) \tag{4.9}$$

with $\boldsymbol{a}_{\mathrm{B},i}^{\intercal} = [a_{\mathrm{B},i,1}, \ldots, a_{\mathrm{B},i,m}]$ and

$$\begin{aligned}\Delta_{B_i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}}) &= B_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_{\boldsymbol{\eta}}) - B_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) &(4.10)\\ \boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{B},i}}^{\intercal}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\eta}) &= \boldsymbol{a}_{\mathrm{B},i}^{\intercal}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_{\eta}) - \boldsymbol{a}_{\mathrm{B},i}^{\intercal}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) &(4.11)\\ \Delta_{b_{\mathrm{B},i}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}}) &= b_{\mathrm{B},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_{\boldsymbol{\eta}}) - b_{\mathrm{B},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) \ . &(4.12)\end{aligned}$$

Analyzing the expressions shows that under Assumption 4.1, the uncertainties $\Delta_k(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}})$ in these functions are bounded as well.

Lemma 4.1. *Let the system and constraints be given by* (2.1) *and* (2.11), *respectively. Let Assumptions 2.1–2.3, 4.1 hold. Then, the uncertainties* $\Delta_k(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}})$ *with*

$$k \in \{h_{c,i}, \dot{h}_{c,i}, \ldots, h_{c,i}^{(r-1)}, a_{c,i,1}, \ldots, a_{c,i,m}, b_{c,i}, B_i, a_{B,i,1}, \ldots, a_{B,i,m}, b_{B,i}\}$$

from (4.4)–(4.12) *are bounded at each instant of time.*

The proof is provided in the appendix. As a result of Lemma 4.1, (4.4)–(4.7) and (4.8)–(4.9) are subject to a uniform bound independent from the value of $\boldsymbol{\Delta}_{\boldsymbol{\eta}}$

$$\Delta_k^{\min_{\mathcal{D}_{\mathrm{rob}}}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) \leq \Delta_k(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}}) \leq \Delta_k^{\max_{\mathcal{D}_{\mathrm{rob}}}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) \tag{4.13}$$

$$\begin{aligned}\text{where } \Delta_k^{\min_{\mathcal{D}_{\mathrm{rob}}}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) &= \inf_{\boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}_{\mathrm{rob}}} \Delta_k(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}})\\ \Delta_k^{\max_{\mathcal{D}_{\mathrm{rob}}}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) &= \sup_{\boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}_{\mathrm{rob}}} \Delta_k(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}})\end{aligned}$$

with $k \in \{h_{\mathrm{c},i}, \dot{h}_{\mathrm{c},i}, \ldots, h_{\mathrm{c},i}^{(r-1)}, a_{\mathrm{c},i,1}, \ldots, a_{\mathrm{c},i,m}, b_{\mathrm{c},i}, B_i, a_{\mathrm{B},i,1}, \ldots, a_{\mathrm{B},i,m}, b_{\mathrm{B},i}\}$. With these results, it is possible to derive invariance control and CBF-based control for robust constraint satisfaction.

4.1.1 Robust Invariance Control

Based on Def. 4.1 and Assumption 4.1, the notion of a robust invariant set is defined.

Definition 4.2. The ***robust invariant set*** is a subset of the state space, in which all invariance functions (3.10) are non-positive for all possible instances of the uncertainty $\boldsymbol{\Delta}_{\boldsymbol{\eta}}$.

$$\mathcal{G}_{\mathrm{rob}}(\overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{\mathrm{rob}}, \gamma) = \{\boldsymbol{x} \in \mathbb{R}^n \mid \Phi_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_{\boldsymbol{\eta}}, \gamma_i) \leq 0 \quad \forall\, i \in \mathcal{B}_{\mathrm{rob}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}_{\mathrm{rob}}\}$$

If a control scheme is able to render this set controlled positively invariant, the constraints are robustly satisfied for all possible instances of the uncertainty. This requires the determination of a robust invariance function, which determines when a switch to corrective control is necessary to achieve the desired positive invariance.

Theorem 4.1. *Let the system and constraints be given by* (2.1) *and* (2.11)*, respectively. Let Assumptions 2.1–2.3, 4.1 hold. Then, the robust invariance function*

$$\Phi_{rob,i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{rob}, \gamma_i) = \max_{\Delta t \geq 0} p_i^{\max}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{rob}, \gamma_i, \Delta t) \tag{4.14}$$

with the polynomial

$$p_i^{\max}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{rob}, \gamma_i, \Delta t) = \sum_{j=0}^{r_i-1} \frac{\Delta t^j}{j!} \Delta_{h_{c,i}^{(j)}}^{\max_{\mathcal{D}_{rob}}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) + p_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \gamma_i, \Delta t) \tag{4.15}$$

and $p_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \gamma_i, \Delta t)$ *from* (3.11) *fulfills*

$$\Phi_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}}, \gamma_i) \leq \Phi_{rob,i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{rob}, \gamma_i) \quad \forall \boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}_{rob} \tag{4.16}$$

with the invariance function $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}}, \gamma_i)$ *from* (3.10) *and* $\boldsymbol{x}_{\boldsymbol{\eta}} = \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_{\boldsymbol{\eta}}$.

The proof is provided in the appendix. This theorem provides a robust invariance function that takes a non-negative value if for any instance of the uncertainty, the invariance function (3.10) is non-negative. As corrective action should be taken if there is only the slightest chance of a constraint violation, the set of active robust constraints is introduced based on the robust invariance function (4.14) to indicate constraints requiring corrective action.

Definition 4.3. The ***set of active robust constraints*** contains the indices of all constraints requiring robust satisfaction for which the corresponding robust invariance function (4.14) is non-negative.

$$\mathcal{B}_{\text{rob,act}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{\text{rob}}, \boldsymbol{\gamma}) = \{i \in \mathcal{B}_{\text{rob}} \mid \Phi_{\text{rob},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{\text{rob}}, \gamma_i) \geq 0\}$$

For the determination of corrective control, the following theorem generalizes Theorem 3.1 to the case when bounded uncertainties influence the constraints.

Theorem 4.2. *Let the system and constraints be given by* (2.1) *and* (2.11)*. Let Assumptions 2.1–2.6, 4.1 hold. Then, if the optimization problem*

$$\begin{aligned} \boldsymbol{u}_c = \operatorname*{argmin}_{\boldsymbol{u}} \; & \|\boldsymbol{u} - \boldsymbol{u}_{no}\|_2^2 \\ \textit{s.t.} \; & \boldsymbol{a}_{c,i}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})\boldsymbol{u} + \left(\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{c,i}}\right|^{\max_{\mathcal{D}_{rob}}}\right)^{\mathsf{T}} |\boldsymbol{u}| + b_{c,i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) + \Delta_{b_{c,i}}^{\max_{\mathcal{D}_{rob}}} \leq z_{c,i} \quad \forall\, i \in \mathcal{B}_{rob,act} \end{aligned} \tag{4.17}$$

yields a solution for all $t \geq t_0$*, where* $z_{c,i}$ *is determined from* (3.16) *using the robust invariance function and the elements of* $\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{c,i}}\right|^{\max_{\mathcal{D}_{rob}}}$ *fulfill*

$$\left|\Delta_{\boldsymbol{a}_{c,i}}\right|_k^{\max_{\mathcal{D}_{rob}}} = \max\left(\left|\Delta_{\boldsymbol{a}_{c,i,k}}^{\min_{\mathcal{D}_{rob}}}\right|, \left|\Delta_{\boldsymbol{a}_{c,i,k}}^{\max_{\mathcal{D}_{rob}}}\right|\right)$$

with $1 \leq k \leq m$*, the control input* $\boldsymbol{u} = \boldsymbol{u}_c$ *renders the robust invariant set* $\mathcal{G}_{rob}$ *controlled positively invariant and achieves robust satisfaction of the constraints* $i \in \mathcal{B}_{rob}$.

The proof is provided in the appendix. With this theorem, it is possible to find a control input, which achieves the satisfaction of the constraints without having to consider the infinite range of uncertainties but only a single robust condition per constraint. In addition, the optimization problem (4.17) from Theorem 4.2 has the following property.

Corollary 4.2.1. *The optimization problem* (4.17) *is convex and any local minimum is a global minimum.*

The proof is provided in the appendix. As computationally efficient methods exist to solve such convex optimization problems, this enables the determination of a control input in real time.

4.1.2 Robust CBF-based Control

Using Lemma 4.1 and (4.13), CBFs may be used to achieve robust constraint satisfaction.

Theorem 4.3. *Let the nonlinear system be given by* (2.1) *and the outputs describing the constraints by* (2.11)*. Let further Assumptions 2.1–2.6 and 4.1 hold. Let $B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$ be CBFs corresponding to each constraint function $y_{B,i}$, $i \in \mathcal{B}_{rob}$. Then, the control input $\boldsymbol{u} = \boldsymbol{u}_c$ obtained by solving the minimization*

$$\begin{aligned} \boldsymbol{u}_c &= \underset{\boldsymbol{u}}{\operatorname{argmin}} \left\|\boldsymbol{u} - \boldsymbol{u}_{no}\right\|_2^2 \qquad (4.18)\\ s.t.\ \boldsymbol{a}_{B,i}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})\boldsymbol{u} &+ \left(\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{B,i}}\right|^{\max_{\mathcal{D}_{rob}}}\right)^{\mathsf{T}} |\boldsymbol{u}| + b_{B,i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) + \Delta_{b_{B,i}}^{\max_{\mathcal{D}_{rob}}} \\ &\leq \frac{\mu_i}{B_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) + \Delta_{B_i}^{\max_{\mathcal{D}_{rob}}}} \quad \forall\, i \in \mathcal{B}_{rob}\ , \end{aligned}$$

where the elements of $\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{B,i}}\right|^{\max_{\mathcal{D}_{rob}}}$ *fulfill*

$$\left|\Delta_{\boldsymbol{a}_{B,i}}\right|_k^{\max_{\mathcal{D}_{rob}}} = \max\left(\left|\Delta_{\boldsymbol{a}_{B,i,k}}^{\min_{\mathcal{D}_{rob}}}\right|, \left|\Delta_{\boldsymbol{a}_{B,i,k}}^{\max_{\mathcal{D}_{rob}}}\right|\right)$$

with $1 \leq k \leq m$, renders the interior of the robust admissible set $\mathcal{H}_{rob}$ positively invariant and achieves robust satisfaction of the constraints $i \in \mathcal{B}_{rob}$.

The proof is provided in the appendix. As robust CBF-based control renders the robust admissible set invariant by Theorem 4.3, the approach is able to guarantee robust constraint satisfaction. In addition, the optimization problem (4.18) is convex, which allows the employment of computationally efficient methods to solve the minimization in real time.

Corollary 4.3.1. *The optimization problem* (4.18) *is strictly convex and any local minimum is a global minimum.*

The proof is analogous to the proof of Corollary 4.2.1. Both robust invariance control and robust CBF-based control are able to enforce constraints with bounded uncertainties. If the uncertainty distribution is not bounded or if the uncertainty distribution is known and probabilistic constraint satisfaction suffices, a probabilistic approach may be beneficial and lead to less conservative results.

4.2 Probabilistic Constraint Satisfaction

Probabilistic constraint satisfaction aims at providing a minimum probability p_i of satisfying each constraint $i \in \mathcal{B}_{\text{pr}}$. This requires the derivation of probabilistic uncertainty bounds which fulfill the following assumption.

Assumption 4.2. *The uncertainty* $\boldsymbol{\Delta_\eta} \in \mathcal{D}$ *with* $\mathcal{D}$ *from* (2.18) *is with probability* $p_i < 1$ *within the known bounded set* $\mathcal{D}_{p_i} \subset \mathcal{D}$*, i.e.* $\mathscr{P}(\mathcal{D}_{p_i}) = p_i$*.*

If such a set $\mathcal{D}_{p_i}$ and its bounds are not known, knowledge of the underlying probabilistic distribution of $\mathcal{D}$ is required to be able to determine $\mathcal{D}_{p_i}$. The goal is to map the parametric uncertainty to the constraints to enforce them with a certain probabilistic satisfaction guarantee.

Definition 4.4 (p_i-satisfaction)**.** A control scheme achieves p_i***-satisfaction*** to a constraint $i \in \mathcal{B}_{\mathrm{pr}}$ with uncertain parameters, if the probability of violating that constraint is at most $1 - p_i$.

Similar to the robust constraint satisfaction, we use the formulations (4.1)–(4.12) to model the uncertainty in the constraints, the CBFs and their derivatives. Based on this description and Assumption 4.2 it is possible to derive probabilistic bounds on the respective uncertainties.

Lemma 4.2. *Let the system and constraints be given by* (2.1) *and* (2.11)*, respectively. Let Assumptions 2.1–2.3, 4.2 hold. Let*

$$\mathcal{D}_k(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}, \mathcal{D}_{p_i}) = \left\{ \boldsymbol{\Delta_\eta} \in \mathcal{D} \middle| \, \Delta_k^{\min_{\mathcal{D}_{p_i}}}(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}) \leq \Delta_k(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}, \boldsymbol{\Delta_\eta}) \leq \Delta_k^{\max_{\mathcal{D}_{p_i}}}(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}) \right\}$$

$$\begin{aligned} \textit{where } \Delta_k^{\min_{\mathcal{D}_{p_i}}}(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}) &= \inf_{\boldsymbol{\Delta_\eta} \in \mathcal{D}_{p_i}} \Delta_k(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}, \boldsymbol{\Delta_\eta}) \\ \Delta_k^{\max_{\mathcal{D}_{p_i}}}(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}) &= \sup_{\boldsymbol{\Delta_\eta} \in \mathcal{D}_{p_i}} \Delta_k(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}, \boldsymbol{\Delta_\eta}) \end{aligned}$$

with $k \in \{h_{c,i}, \dot{h}_{c,i}, \dots, h_{c,i}^{(r-1)}, a_{c,i,1}, \dots, a_{c,i,m}, b_{c,i}, B_i, a_{B,i,1}, \dots, a_{B,i,m}, b_{B,i}$ *and* Δ_k *from* (4.4)-(4.12)*. Then,*

$$\mathscr{P}(\mathcal{D}_k) \geq p_i \tag{4.19}$$

holds at each instant of time.

The proof is provided in the appendix. This means that, similar to the robust case, there exist bounds within which the function uncertainties Δ_k lie with a certain probability.

4.2.1 Probabilistic Invariance Control

With the previous results, a probabilistic invariance function is derived.

Theorem 4.4. *Let the system and constraints be given by* (2.1) *and* (2.11)*, respectively. Let Assumptions 2.1–2.3, 4.2 hold. Then, the* p_i*-invariance function*

$$\Phi_{pr,i}(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}, \mathcal{D}_{p_i}, \gamma_i) = \max_{\Delta t \geq 0} p_i^{\max}(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}, \mathcal{D}_{p_i}, \gamma_i, \Delta t) \tag{4.20}$$

with $p_i^{\max}(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}, \mathcal{D}_{p_i}, \gamma_i, \Delta t)$ *from* (4.15) *fulfills*

$$\begin{aligned} \mathscr{P}(\mathcal{T}_i \wedge \mathcal{V}_i) &\leq 1 - p_i && (4.21) \\ \textit{with } \mathcal{T}_i &= \{ \boldsymbol{\Delta_\eta} \in \mathcal{D} | \, \Phi_{pr,i}(\boldsymbol{x}, \overline{\boldsymbol{x}_\eta}, \mathcal{D}_{p_i}, \gamma_i) \leq 0 \} \ , && (4.22) \\ \mathcal{V}_i &= \{ \boldsymbol{\Delta_\eta} \in \mathcal{D} | \, \Phi_i(\boldsymbol{x}, \boldsymbol{x_\eta}, \gamma_i) > 0 \} \ , && (4.23) \end{aligned}$$

the invariance function $\Phi_i(\boldsymbol{x}, \boldsymbol{x_\eta}, \gamma_i)$ *from* (3.10) *and* $\boldsymbol{x_\eta} = \overline{\boldsymbol{x}_\eta} + \boldsymbol{\Delta_\eta}$ *for all instances of the uncertainty* $\boldsymbol{\Delta_\eta}$*.*

The proof is provided in the appendix. An interpretation of this theorem is that the probability of unsafe behavior, i.e. a constraint violation due to undetected danger because of a wrong negative value of the probabilistic invariance function, is at most $1 - p_i$. The derived p_i-invariance function is used to define the set of active probabilistic constraints.

Definition 4.5. The ***set of active probabilistic constraints*** contains the indices of all constraints requiring p_i-satisfaction for which the corresponding p_i-invariance function (4.20) is non-negative.

$$\mathcal{B}_{\mathrm{pr,act}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{p_i}, \boldsymbol{\gamma}) = \{i \in \mathcal{B}_{\mathrm{pr}} | \Phi_{\mathrm{pr},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{p_i}, \gamma_i) \geq 0\}$$

Using these considerations, a corrective control input, which achieves p_i-satisfaction of each constraint, is determined.

Theorem 4.5. *Let the system and constraints be given by* (2.1) *and* (2.11)*, respectively. Let Assumptions 2.1–2.6, 4.2 hold. Then, if the optimization problem*

$$\begin{aligned} \boldsymbol{u}_c &= \operatorname*{argmin}_{\boldsymbol{u}} \|\boldsymbol{u} - \boldsymbol{u}_{no}\|_2^2 \qquad (4.24)\\ s.t.\ \boldsymbol{a}_{c,i}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})\boldsymbol{u} &+ \left(\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{c,i}}\right|^{\max_{\mathcal{D}_{p_i}}}\right)^{\mathsf{T}} |\boldsymbol{u}| + b_{c,i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) + \Delta_{b_{c,i}}^{\max_{\mathcal{D}_{p_i}}} \leq z_{c,i} \quad \forall\ i \in \mathcal{B}_{pr,act} \end{aligned}$$

yields a solution for all $t \geq t_0$, where $z_{c,i}$ is determined from (3.16) *using the p_i-invariance function and the elements of $\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{c,i}}\right|^{\max_{\mathcal{D}_{p_i}}}$ fulfill*

$$\left|\Delta_{\boldsymbol{a}_{c,i}}\right|_k^{\max_{\mathcal{D}_{p_i}}} = \max\left(\left|\Delta_{\boldsymbol{a}_{c,i,k}}^{\min_{\mathcal{D}_{p_i}}}\right|, \left|\Delta_{\boldsymbol{a}_{c,i,k}}^{\max_{\mathcal{D}_{p_i}}}\right|\right) ,$$

for $1 \leq k \leq m$, the input $\boldsymbol{u} = \boldsymbol{u}_c$ achieves at least p_i-satisfaction of each constraint $i \in \mathcal{B}_{pr}$.

The proof is provided in the appendix. With this theorem, the control input achieves p_i-satisfaction of the constraints without having to consider the infinite range of uncertainties but only a single condition per constraint. In addition, the optimization problem (4.24) from Theorem 4.5 has the following property.

Corollary 4.5.1. *The optimization problem* (4.24) *is strictly convex and any local minimum is a global minimum.*

The proof is analogous to the proof of Corollary 4.2.1. Hence, similar to robust invariance control, this enables the determination of a control input in real time.

4.2.2 Probabilistic CBF-based Control

Using Lemma 4.2 allows for CBFs being used to achieve probabilistic constraint satisfaction.

Theorem 4.6. *Let the nonlinear system be given by* (2.1) *and the outputs describing the constraints by* (2.11)*. Let further Assumptions 2.1–2.6 and 4.2 hold. Let $B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$ be CBFs* *corresponding to each constraint function* $y_{B,i}$*,* $i \in \mathcal{B}_{pr}$*. Then, the control input* $\boldsymbol{u} = \boldsymbol{u}_c$ *obtained by solving the minimization*

$$\begin{aligned} \boldsymbol{u}_c = \underset{\boldsymbol{u}}{\operatorname{argmin}} \, \|\boldsymbol{u} - \boldsymbol{u}_{no}\|_2^2 \\ s.t. \;\; \boldsymbol{a}_{B,i}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})\boldsymbol{u} + \left(\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{B,i}}\right|^{\max_{\mathcal{D}_{p_i}}}\right)^{\mathsf{T}} |\boldsymbol{u}| + b_{B,i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) + \Delta_{b_{B,i}}^{\max_{\mathcal{D}_{p_i}}} \\ \leq \frac{\mu_i}{B_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) + \Delta_{B_i}^{\max_{\mathcal{D}_{p_i}}}} \quad \forall \; i \in \mathcal{B}_{pr} \end{aligned} \tag{4.25}$$

with the elements of $\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{B,i}}\right|^{\max_{\mathcal{D}_{p_i}}}$ *fulfilling*

$$\left|\Delta_{\boldsymbol{a}_{B,i}}\right|_k^{\max_{\mathcal{D}_{p_i}}} = \max\left(\left|\Delta_{\boldsymbol{a}_{B,i,k}}^{\min_{\mathcal{D}_{p_i}}}\right|, \left|\Delta_{\boldsymbol{a}_{B,i,k}}^{\max_{\mathcal{D}_{p_i}}}\right|\right) \;,$$

for $1 \leq k \leq m$*, achieves at least* p_i*-satisfaction of each constraint* $i \in \mathcal{B}_{pr}$*.*

The proof is provided in the appendix. Similar to probabilistic invariance control, the control input proposed in this theorem achieves p_i-satisfaction of the constraints without having to consider the infinite range of uncertainties but only a single condition per constraint. In addition, the optimization is convex, which allows the employment of computationally efficient methods to solve the minimization in real time.

Corollary 4.6.1. *The optimization problem* (4.25) *is strictly convex and any local minimum is a global minimum.*

The proof is analogous to the proof of Corollary 4.2.1. The presented probabilistic control approaches are able to enforce uncertain constraints, if the uncertainty distribution is known. Otherwise, a sampling-based approach may be advantageous.

4.3 Scenario-based Constraint Satisfaction

The control approach presented in this section is based on the principles of scenario-based optimization [CC05; CG08]. This method provides means to enforce constraints with uncertain parameters in convex optimization problems. In order to make the problem feasible, scenario optimization reduces the number of constraints by drawing samples of the uncertainty at the cost of only satisfying them up to a predefined probability. By using measured data to generate the optimization scenarios, the approach does not require knowledge about the underlying distribution of the uncertainties. The minimal number of required samples is provided in [CG08].

Theorem 4.7 (Theorem 1 in [CGP09])**.** *Consider the following optimization problem*

$$\begin{aligned} \boldsymbol{u}^* = \underset{\boldsymbol{u} \in \mathbb{R}^m}{\operatorname{argmin}} \; J(\boldsymbol{u}) \\ s.t. \;\; f_{opt}(\boldsymbol{u}, \boldsymbol{\delta}_i) \leq 0 \quad \forall \; i = 1, \ldots, N \end{aligned}$$

where cost function $J(\boldsymbol{u})$ *and constraint* $f_{opt}(\boldsymbol{u}, \boldsymbol{\delta}_i)$ *are both convex in* $\boldsymbol{u}$*. Let* $\boldsymbol{\delta}_1, ..., \boldsymbol{\delta}_N$ *be samples of* $\boldsymbol{\delta} \in \mathcal{D}$*, extracted independently from* $\mathcal{D}$ *according to the same probability distribution. Let* N *be chosen to fulfill*

$$N \geq \frac{2}{\epsilon}\left(\log\left(\frac{1}{\beta}\right) + m\right)$$

with the desired maximum violation probability $\epsilon \in (0,1)$ *and the confidence* $\beta \in (0,1)$ *that this probability is achieved. Then, the minimizer* $\boldsymbol{u}^*$*, satisfies* $f_{opt}(\boldsymbol{u}^*, \boldsymbol{\delta}) \leq 0$ *for all* $\delta \in \mathcal{D}$ *except for a fraction of probability smaller than or equal to* ϵ *with confidence* $1 - \beta$*.*

In other words, by choosing sufficiently many samples of the uncertainty, the probability of the actual uncertainty being covered is at least $1-\epsilon$. However, as the samples are generated randomly, either from the known probability distribution or from measurement data, there is a chance that they are chosen badly, thus covering less than the expected amount of uncertainty values. The probability of this happening is at most β.

Remark 4.2. In [CG08], convex optimization problems with the cost function $J(\boldsymbol{u}) = \boldsymbol{c}^\intercal \boldsymbol{u}$ are considered. However, Theorem 4.7 holds for any convex cost function $J(\boldsymbol{u})$ which can be reduced to a linear form by means of an epigraphic reformulation.

Based on the results, which scenario optimization is able to achieve, we define the following notion of probabilistic constraint satisfaction.

Definition 4.6. A control scheme achieves (p_i, β_i)***-satisfaction*** to a constraint with uncertain parameters, if p_i-satisfaction of that constraint is achieved with at least a probability of $1 - \beta_i$.

Scenario-based constraint satisfaction aims at achieving (p_i, β_i)-satisfaction for all constraints $i \in \mathcal{B}_{\text{sc}}$. In order to allow for this, we make the following assumption.

Assumption 4.3. *The set of uncertainty samples*

$$\mathcal{D}_{sc} = \{\boldsymbol{x_\eta} \in \mathcal{D} | \boldsymbol{x_\eta} = \boldsymbol{x}_{\boldsymbol{\eta}_s}\ , \text{s} \in \{1 \dots N\}\}$$

with $\mathcal{D}$ *from* (2.18) *contains sufficiently many scenarios of the parameters* $\boldsymbol{x_\eta} = \overline{\boldsymbol{x_\eta}} + \boldsymbol{\Delta}_\eta$*, i.e.* $N > N_i$ *holds for all* $i \in \mathcal{B}_{sc}$*, where* N_i *is the number of scenarios required for enforcing constraint* i*.*

Inserting the scenarios $\boldsymbol{x}_{\boldsymbol{\eta}_s}$ into the output functions (2.11), the CBFs defined in Def. 3.11, the derivatives and the linearization (2.14) yields the scenario constraints $h_{\text{c},i}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})$, the scenario CBFs $B_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})$, their derivatives $h_{\text{c},i}^{(r)}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})$ with $r \in \{1 \dots r_i\}$, $\dot{B}_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s}, \boldsymbol{u})$ and the linearization scenarios $\boldsymbol{a}_{\text{c},i}^\intercal(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})$, $b_{\text{c},i}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})$, $\boldsymbol{a}_{\text{B},i}^\intercal(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})$, $b_{\text{B},i}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})$.

4.3.1 Scenario Invariance Control

Using the scenario constraints, the scenario invariance functions $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s}, \gamma_i)$ are derived from (3.10) and the set of active scenario constraints is defined.

Definition 4.7. The ***set of active scenario constraints*** consists of the tuples of those constraints i and uncertainty samples s, for which the scenario invariance function has a non-negative value.

$$\mathcal{B}_{\text{sc,act}}(\boldsymbol{x}, \mathcal{D}_{\text{sc}}, \boldsymbol{\gamma}) = \{(i, s) \mid \Phi_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s}, \gamma_i) \geq 0, \quad \forall\ s \in \{1 \dots N_i\}, i \in \mathcal{B}_{\text{sc}}\}$$

Based on the set of active constraints and the scenarios of the linearization $\boldsymbol{a}_{\text{c},i}^\intercal(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})$ and $b_{\text{c},i}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})$, corrective control is determined.

Theorem 4.8. *Let the system and constraints be given by* (2.1) *and* (2.11)*, respectively. Let Assumptions 2.1–2.6, 4.3 hold. Then, the optimization problem*

$$\begin{aligned} \boldsymbol{u}_c = \underset{\boldsymbol{u}}{\operatorname{argmin}} \, \|\boldsymbol{u} - \boldsymbol{u}_{no}\|_2^2 \\ \text{s.t. } \boldsymbol{a}_{c,i}^\intercal(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})\boldsymbol{u} + b_{c,i}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s}) \leq z_{c,i} \quad \forall (i,s) \in \mathcal{B}_{sc,act} \end{aligned} \tag{4.26}$$

where $z_{c,i}$ *is determined from* (3.16) *using* $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s}, \gamma_i)$ *and*

$$N_i = \left\lceil \frac{2}{1-p_i} \left(\ln\left(\frac{1}{\beta_i}\right) + m \right) \right\rceil \ ,$$

is convex and if it yields a solution for all $t \geq t_0$*, the control input* $\boldsymbol{u} = \boldsymbol{u}_c$ *achieves at least* (p_i, β_i)*-satisfaction of each constraint* $i \in \mathcal{B}_{sc}$*.*

The proof is provided in the appendix.

4.3.2 Scenario CBF-based Control

Based on the scenarios of the CBFs $B_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})$ and the I/O-linearization, corrective control is determined.

Theorem 4.9. *Let the system and constraints be given by* (2.1) *and* (2.11)*, respectively. Let Assumptions 2.1–2.6, 4.3 hold. Let* $B_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{x}_\eta})$ *be CBFs corresponding to each constraint function* $y_{B,i}$*,* $i \in \mathcal{B}_{sc}$*. Then, the optimization problem*

$$\begin{aligned} \boldsymbol{u}_c = \underset{\boldsymbol{u}}{\operatorname{argmin}} \, \|\boldsymbol{u} - \boldsymbol{u}_{no}\|_2^2 \\ \text{s.t. } \boldsymbol{a}_{B,i}^\intercal(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})\boldsymbol{u} + b_{B,i}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s}) \leq \frac{\mu_i}{B_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})} \quad \forall (i,s) \in \mathcal{B}_{sc} \times \{1 \dots N_i\} \end{aligned} \tag{4.27}$$

with

$$N_i = \left\lceil \frac{2}{1-p_i} \left(\ln\left(\frac{1}{\beta_i}\right) + m \right) \right\rceil$$

is convex and, if it yields a solution for all $t \geq t_0$*, the control input* $\boldsymbol{u} = \boldsymbol{u}_c$ *achieves at least* (p_i, β_i)*-satisfaction of each constraint* $i \in \mathcal{B}_{sc}$*.*

The proof is provided in the appendix. Both scenario invariance control and scenario CBF-based control determine a control input, which gives probabilistic guarantees for constraint enforcement by generating enough samples of the uncertainties. This means that the approaches may even be applied if the uncertainty distribution is not explicitly known but measurement data is available.

4.4 Combination of Approaches

Sometimes it may be necessary or desired to combine robust, probabilistic and scenario-based approaches, either because the constraint does not meet the requirements for one approach or another approach yields more advantageous result for the specific constraint. In that case, the following theorem applies.

Theorem 4.10. *Let the system and constraints be given by* (2.1) *and* (2.11)*, respectively. Let $B_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{x}_\eta})$ be CBFs corresponding to each constraint function $y_{B,i}$ if the goal is CBF-based control. Let the related parameter uncertainties be such that for constraints $i \in \mathcal{B}_{rob}$, $j \in \mathcal{B}_{pr}$ and $k \in \mathcal{B}_{sc}$, the uncertainties fulfill Assumption 4.1, 4.2 and 4.3, respectively. Let Assumptions 2.1–2.6 hold. Let the optimization problem*

$$\boldsymbol{u}_c = \underset{\boldsymbol{u}}{\operatorname{argmin}} \|\boldsymbol{u} - \boldsymbol{u}_{no}\|_2^2 \tag{4.28}$$

subject to

$$\begin{aligned} \boldsymbol{a}_i^\intercal(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})\boldsymbol{u} + (|\boldsymbol{\Delta}_{\boldsymbol{a}_i}|^{\max_{\mathcal{D}_{rob}}})^\intercal\, |\boldsymbol{u}| + b_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) + \Delta_{b_i}^{\max_{\mathcal{D}_{rob}}} &\le d_i \quad \forall\, i \in \tilde{\mathcal{B}}_{rob}\ , \\ \boldsymbol{a}_j^\intercal(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})\boldsymbol{u} + \left(\left|\boldsymbol{\Delta}_{\boldsymbol{a}_j}\right|^{\max_{\mathcal{D}_{p_j}}}\right)^\intercal |\boldsymbol{u}| + b_j(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) + \Delta_{b_j}^{\max_{\mathcal{D}_{p_j}}} &\le d_j \quad \forall\, j \in \tilde{\mathcal{B}}_{pr}\ , \\ \boldsymbol{a}_k^\intercal(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})\boldsymbol{u} + b_k(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s}) &\le d_k \quad \forall\, (k,s) \in \tilde{\mathcal{B}}_{sc} \end{aligned}$$

where

$$\begin{gathered} \boldsymbol{a}_r = \boldsymbol{a}_{c,r}\ , \quad b_r = b_{c,r}\ , \quad d_r = z_{c,r} \quad \textit{with } r \in \{i,j,k\} \\ \tilde{\mathcal{B}}_{rob} = \mathcal{B}_{rob,act}\ , \quad \tilde{\mathcal{B}}_{pr} = \mathcal{B}_{pr,act}\ , \quad \tilde{\mathcal{B}}_{sc} = \mathcal{B}_{sc,act} \end{gathered}$$

with $z_{c,r}$ according to (3.16) *for invariance control and*

$$\begin{aligned} \boldsymbol{a}_r &= \boldsymbol{a}_{B,r}\ , \quad b_r = b_{B,r} \quad \textit{with } r \in \{i,j,k\} \\ d_r &= \frac{\mu_r}{B_r(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) + \Delta_{B_r}^{\max_{\mathcal{D}_{rob}}}} \quad \textit{with } r \in \{i,j\} \\ d_k &= \frac{\mu_k}{B_k(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s})} \\ \tilde{\mathcal{B}}_{rob} &= \mathcal{B}_{rob}\ , \quad \tilde{\mathcal{B}}_{pr} = \mathcal{B}_{pr}\ , \quad \tilde{\mathcal{B}}_{sc} = \mathcal{B}_{sc} \times \{1 \dots N_k\} \end{aligned}$$

for CBF-based control be given. Let the elements of $|\boldsymbol{\Delta}_{\boldsymbol{a}_r}|_q^{\max_{\mathcal{D}_R}}$ be given by

$$|\Delta_{\boldsymbol{a}_r}|_q^{\max_{\mathcal{D}_R}} = \max\left(\left|\Delta_{\boldsymbol{a}_{r,q}}^{\min_{\mathcal{D}_R}}\right|, \left|\Delta_{\boldsymbol{a}_{r,q}}^{\max_{\mathcal{D}_R}}\right|\right)\ , \quad 1 \le q \le m$$

with the sets $\mathcal{D}_R = \mathcal{D}_{rob}$ for $r = i$, $\mathcal{D}_R = \mathcal{D}_{p_j}$ for $r = j$ and let the number of scenarios be

$$N_k = \left\lceil \frac{2}{1 - p_k}\left(\ln\left(\frac{1}{\beta_k}\right) + m\right)\right\rceil\ .$$

Then the optimization is convex and, if it yields a solution for all $t \ge t_0$, the minimizing input $\boldsymbol{u} = \boldsymbol{u}_c$ achieves robust satisfaction of the constraints $i \in \mathcal{B}_{rob}$, at least p_j-satisfaction of each constraint $j \in \mathcal{B}_{pr}$ and at least (p_k, β_k)-satisfaction of each constraint $k \in \mathcal{B}_{sc}$.

The proof is provided in the appendix. The possibility of combining the different control approaches allows the application in a variety of setups and environments. The choice depends on the required properties of the outcome und depends on the characteristics of the different methods, which are illustrated in the following numerical example. Note that since invariance control and CBF-based control result in similar optimization conditions, the approaches may even be combined as mentioned in Remark 3.15.

Remark 4.3. In Theorems 4.2–4.10, the cost function $\|\boldsymbol{u}-\boldsymbol{u}_{\text{no}}\|_2^2$ is used to derive corrective control. Nevertheless, as the proofs showing constraint adherence are independent from the choice of cost function $C(\boldsymbol{u},\boldsymbol{u}_{\text{no}})$, any convex function may be chosen as long as $C(\boldsymbol{u},\boldsymbol{u}_{\text{no}})$ fulfills the requirements introduced in (3.31) and Remark 4.2.

Finally, note that the boundedness of the controlled system state is not influenced by uncertainties in the constraint.

Remark 4.4. As the guaranteed boundedness of the controlled system does not depend on the number of constraints but on adding a Lyapunov-based boundedness condition as proposed in Theorem 3.3, the added uncertainties do not change the general behavior as they solely increase the number of optimization constraints.

4.5 Numerical Example

The numerical example is designed such that it illustrates the characteristics of the three introduced approaches for the satisfaction of uncertain constraints. The calculations are executed for invariance control and the setup is kept simple enough so the probability bounds are analytically deducible to verify the outcome.

4.5.1 System and Constraints

We consider a system of the form

$$\ddot{\boldsymbol{p}} = \boldsymbol{u}$$

with $\boldsymbol{p},\boldsymbol{u}\in\mathbb{R}^2$, the state $\boldsymbol{x}=[\boldsymbol{p}^\intercal\ \dot{\boldsymbol{p}}^\intercal]^\intercal\in\mathbb{R}^4$ and nominal PD tracking control

$$\boldsymbol{u}_{\text{no}} = \ddot{\boldsymbol{p}}_{\text{des}} + \boldsymbol{K_P}(\boldsymbol{p}_{\text{des}}-\boldsymbol{p}) + \boldsymbol{K_D}(\dot{\boldsymbol{p}}_{\text{des}}-\dot{\boldsymbol{p}})\ ,$$

with the positive definite matrices $\boldsymbol{K_P},\boldsymbol{K_D}\in\mathbb{R}^{2\times2}$, which achieves stable tracking of the desired trajectory $\boldsymbol{p}_{\text{des}}(t)\in\mathbb{R}^2$. Two constraints are introduced and both system and constraints are modeled by spheres resulting in the following constraint descriptions

$$h_{\text{c},i} = (r_x + r_i) - \|\boldsymbol{p}-\boldsymbol{c_i}\|_2 \quad i\in 1,2 \tag{4.29}$$

where r_x and r_i, $i\in\{1,2\}$ are the radii of system and constraints, respectively. The center positions of the constraints $\boldsymbol{c_i}\in\mathbb{R}^2$ are stochastically distributed around an undisturbed position $\bar{\boldsymbol{c}}_{\boldsymbol{i}}\in\mathbb{R}^2$

$$\boldsymbol{c_i} = \bar{\boldsymbol{c}}_{\boldsymbol{i}} + \boldsymbol{\Delta}_{\boldsymbol{c_i}}\ . \tag{4.30}$$

The distribution of the constant uncertainty $\boldsymbol{\Delta}_{\boldsymbol{c_i}}\in\mathbb{R}^2$ is given by

$$\boldsymbol{\Delta}_{\boldsymbol{c_i}} = \Delta_c\cdot\begin{bmatrix}\cos(\rho)\\ \sin(\rho)\end{bmatrix} \tag{4.31}$$

$$\Delta_c = \Delta_r^\alpha\cdot r_{\max}\ ,\ \Delta_r\in[0,1],\rho\in[0,2\pi[\ . \tag{4.32}$$

The parameters ρ and Δ_r are uniformly distributed, whereas $\alpha\in\mathbb{N}$ and $r_{\max}\in\mathbb{R}^+$ are constants. Since Δ_r is uniformly distributed between 0 and 1,

$$\mathscr{P}(\Delta_r<\epsilon)=\epsilon$$

for $\epsilon \in [0,1]$ holds, which implies

$$\mathscr{P}(\Delta_r^\alpha < \epsilon) = \mathscr{P}\Big(\Delta_r < \sqrt[\alpha]{\epsilon}\Big) = \sqrt[\alpha]{\epsilon} \ .$$

With $\Delta_c = \Delta_r^\alpha r_{\max}$ and $r_{\text{pr}} = \epsilon r_{\max}$,

$$\mathscr{P}(\Delta_c \leq r_{\text{pr}}) = \sqrt[\alpha]{\epsilon} := p_{\text{pr}} \tag{4.33}$$

holds and therefore with probability p_{pr}, the position uncertainty lies within a circle of radius

$$r_{\text{pr}} = p_{\text{pr}}^\alpha r_{\max} \ . \tag{4.34}$$

Note that Assumptions 4.1 and 4.2 hold, since $\boldsymbol{\Delta_\eta} = [\Delta_r \ , \ \rho]^\intercal$ fulfills

$$\begin{aligned} \boldsymbol{\Delta_\eta} \in \mathcal{D}_{\text{rob}} \ , \ \mathcal{D}_{\text{rob}} &= \Big\{\Delta_r \in [0,1], \rho \in [0, 2\pi[\Big\} \\ \mathscr{P}\Big(\mathcal{D}_{p_{\text{pr}}}\Big) \leq p_{\text{pr}} \ , \ \mathcal{D}_{p_{\text{pr}}} &= \Big\{\Delta_r \in [0,1], \rho \in [0, 2\pi[\Big| \, \Delta_c \leq r_{\text{pr}}\Big\} \ . \end{aligned}$$

4.5.2 Control Derivation

In order to determine the constraint enforcing control, the output functions (4.29) are derived with respect to time.

$$\dot{h}_{\text{c},i} = -\frac{(\boldsymbol{p}-\boldsymbol{c_i})^\intercal}{\|\boldsymbol{p}-\boldsymbol{c_i}\|_2}(\dot{\boldsymbol{p}}-\dot{\boldsymbol{c}}_{\boldsymbol{i}}) = \boldsymbol{a}_{\text{c},i}^\intercal(\dot{\boldsymbol{p}}-\dot{\boldsymbol{c}}_{\boldsymbol{i}}) \tag{4.35}$$

$$\boldsymbol{a}_{\text{c},i}^\intercal = -\frac{(\boldsymbol{p}-\boldsymbol{c_i})^\intercal}{\|\boldsymbol{p}-\boldsymbol{c_i}\|_2} \tag{4.36}$$

$$\begin{aligned} b_{\text{c},i} &= \frac{(\boldsymbol{p}-\boldsymbol{c_i})^\intercal}{\|\boldsymbol{p}-\boldsymbol{c_i}\|_2}\ddot{\boldsymbol{c}}_{\boldsymbol{i}} - \frac{\|\dot{\boldsymbol{p}}-\dot{\boldsymbol{c}}_{\boldsymbol{i}}\|_2^2}{\|\boldsymbol{p}-\boldsymbol{c_i}\|_2} - \frac{((\boldsymbol{p}-\boldsymbol{c_i})^\intercal(\dot{\boldsymbol{p}}-\dot{\boldsymbol{c}}_{\boldsymbol{i}}))^2}{\|\boldsymbol{p}-\boldsymbol{c_i}\|_2^3} \\ &= -\boldsymbol{a}_{\text{c},i}^\intercal\ddot{\boldsymbol{c}}_{\boldsymbol{i}} - \frac{\|\dot{\boldsymbol{p}}-\dot{\boldsymbol{c}}_{\boldsymbol{i}}\|_2^2}{\|\boldsymbol{p}-\boldsymbol{c_i}\|_2} - \frac{(\boldsymbol{a}_{\text{c},i}^\intercal(\dot{\boldsymbol{p}}-\dot{\boldsymbol{c}}_{\boldsymbol{i}}))^2}{\|\boldsymbol{p}-\boldsymbol{c_i}\|_2} \end{aligned} \tag{4.37}$$

For robust and probabilistic invariance control, it is necessary to explicitly determine the function uncertainties (4.4)-(4.7). The uncertainty bounds of the constraints are given by

$$|\Delta_{h_{\text{c},i}}| = |\, \|\boldsymbol{p}-\bar{\boldsymbol{c}}_{\boldsymbol{i}}\|_2 - \|\boldsymbol{p}-\bar{\boldsymbol{c}}_{\boldsymbol{i}}-\boldsymbol{\Delta_{c_i}}\|_2 \,| \leq \|\boldsymbol{p}-\bar{\boldsymbol{c}}_{\boldsymbol{i}} - (\boldsymbol{p}-\bar{\boldsymbol{c}}_{\boldsymbol{i}}-\boldsymbol{\Delta_{c_i}})\|_2 = \|\boldsymbol{\Delta_{c_i}}\|_2 = \Delta_c \ .$$

For $\Delta_{\dot{h}_{\text{c},i}}$, $\Delta_{\boldsymbol{a}_{\text{c},i}}$ and $\Delta_{b_{\text{c},i}}$, a geometric approach is taken. As $\boldsymbol{a}_{\text{c},i}$ from (4.36) is a unit vector, which means that

$$\|\boldsymbol{a}_{\text{c},i}\|_2 = \|\bar{\boldsymbol{a}}_{\text{c},i}\|_2 = 1 \tag{4.38}$$

holds, an uncertainty in the constraint position $\boldsymbol{c_i}$ does not change the length but only the orientation. Therefore $\boldsymbol{a}_{\text{c},i}$ may be represented as $\bar{\boldsymbol{a}}_{\text{c},i}$ rotated by $\phi(\Delta_{c_i})$

$$\boldsymbol{a}_{\text{c},i} = \begin{bmatrix} \cos(\phi(\Delta_{c_i})) & -\sin(\phi(\Delta_{c_i})) \\ \sin(\phi(\Delta_{c_i})) & \cos(\phi(\Delta_{c_i})) \end{bmatrix} \bar{\boldsymbol{a}}_{\text{c},i} \tag{4.39}$$

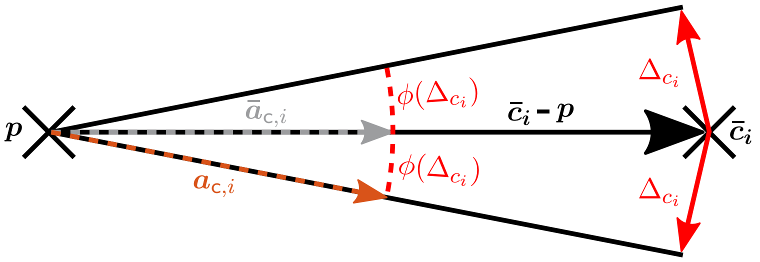

$\boldsymbol{p}$
$\bar{\boldsymbol{a}}_{\mathrm{c},i}$
$\boldsymbol{a}_{\mathrm{c},i}$
$\phi(\Delta_{c_i})$
$\phi(\Delta_{c_i})$
$\bar{\boldsymbol{c}}_{\boldsymbol{i}}$ - $\boldsymbol{p}$
Δ_{c_i}
Δ_{c_i}
$\bar{\boldsymbol{c}}_{\boldsymbol{i}}$

Figure 4.1.: Change in I/O-linearizing vector $\boldsymbol{a}_{\mathrm{c},i}$ from its mean value $\bar{\boldsymbol{a}}_{\mathrm{c},i} = \boldsymbol{a}_{\mathrm{c},i}(\boldsymbol{p}, \bar{\boldsymbol{c}}_{\boldsymbol{i}})$ caused by a bounded disturbance Δ_{c_i}.

and $\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}$ is given by

$$\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}} = \boldsymbol{a}_{\mathrm{c},i} - \bar{\boldsymbol{a}}_{\mathrm{c},i} = \begin{bmatrix} \cos(\phi(\Delta_{c_i})) - 1 & -\sin(\phi(\Delta_{c_i})) \\ \sin(\phi(\Delta_{c_i})) & \cos(\phi(\Delta_{c_i})) - 1 \end{bmatrix} \bar{\boldsymbol{a}}_{\mathrm{c},i} \ .$$

The maximum change in orientation occurs if the direction of the displacement is perpendicular to $\boldsymbol{a}_{\mathrm{c},i}$ as depicted in Fig. 4.1. Due to the symmetry in both directions, the change in orientation $\phi(\Delta_{c_i})$ for a given Δ_{c_i} fulfills

$$|\phi(\Delta_{c_i})| \leq \operatorname{asin}\left(\frac{\Delta_{c_i}}{\|\boldsymbol{p} - \bar{\boldsymbol{c}}_{\boldsymbol{i}}\|_2}\right) := \phi^{\max}(\Delta_{c_i}) \ . \tag{4.40}$$

Hence, the bounds on $\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}$ are given by

$$\begin{aligned} \left\|\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}\right\|_2 &\leq \left\|\begin{bmatrix} \cos(\phi(\Delta_{c_i})) - 1 & -\sin(\phi(\Delta_{c_i})) \\ \sin(\phi(\Delta_{c_i})) & \cos(\phi(\Delta_{c_i})) - 1 \end{bmatrix}\right\|_2 \|\bar{\boldsymbol{a}}_{\mathrm{c},i}\|_2 = \sqrt{2(1 - \cos(\phi(\Delta_{c_i})))} \\ &\leq \sqrt{2(1 - \cos(\phi^{\max}(\Delta_{c_i})))} \\ |\Delta_{a_{\mathrm{c},i,1}}| &= \left|\begin{bmatrix}1 & 0\end{bmatrix} \boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}\right| \leq \left\|\begin{bmatrix}1 & 0\end{bmatrix}\right\|_2 \left\|\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}\right\|_2 \leq \sqrt{2(1 - \cos(\phi^{\max}(\Delta_{c_i})))} := |\Delta_{a_{\mathrm{c},i,1}}|^{\max}(\Delta_{c_i}) \\ |\Delta_{a_{\mathrm{c},i,2}}| &= \left|\begin{bmatrix}0 & 1\end{bmatrix} \boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}\right| \leq \left\|\begin{bmatrix}0 & 1\end{bmatrix}\right\|_2 \left\|\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}\right\|_2 \leq \sqrt{2(1 - \cos(\phi^{\max}(\Delta_{c_i})))} := |\Delta_{a_{\mathrm{c},i,2}}|^{\max}(\Delta_{c_i}). \end{aligned}$$

Furthermore, using (4.38), (4.39) and the fact that $\boldsymbol{\Delta}_{\boldsymbol{c}_{\boldsymbol{i}}}$ is constant, i.e. $\dot{\boldsymbol{c}} = \dot{\bar{\boldsymbol{c}}}$ and $\ddot{\boldsymbol{c}} = \ddot{\bar{\boldsymbol{c}}}$, yields

$$\begin{aligned} \left|\Delta_{\dot{h}_{\mathrm{c},i}}\right| &= \left|\dot{h}_{\mathrm{c},i} - \dot{\bar{h}}_{\mathrm{c},i}\right| = \left|(\boldsymbol{a}^{\mathsf{T}}_{\mathrm{c},i} - \bar{\boldsymbol{a}}^{\mathsf{T}}_{\mathrm{c},i})(\dot{\boldsymbol{p}} - \dot{\bar{\boldsymbol{c}}})\right| \leq \left\|\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}\right\|_2 \left\|\dot{\boldsymbol{p}} - \dot{\bar{\boldsymbol{c}}}\right\|_2 \\ &\leq \sqrt{2(1 - \cos(\phi^{\max}(\Delta_{c_i})))} \ \left\|\dot{\boldsymbol{p}} - \dot{\bar{\boldsymbol{c}}}\right\|_2 := \left|\Delta_{\dot{h}_i}\right|^{\max}(\Delta_{c_i}) \\ |b_{\mathrm{c},i}| &\leq \left\|\ddot{\bar{\boldsymbol{c}}}_{\boldsymbol{i}}\right\|_2 + \frac{\left\|\dot{\boldsymbol{p}} - \dot{\bar{\boldsymbol{c}}}_{\boldsymbol{i}}\right\|_2^2}{\|\boldsymbol{p} - \bar{\boldsymbol{c}}_{\boldsymbol{i}} - \boldsymbol{\Delta}_{\boldsymbol{c}_{\boldsymbol{i}}}\|_2} + \frac{\left\|\dot{\boldsymbol{p}} - \dot{\bar{\boldsymbol{c}}}_{\boldsymbol{i}}\right\|_2^2}{\|\boldsymbol{p} - \bar{\boldsymbol{c}}_{\boldsymbol{i}} - \boldsymbol{\Delta}_{\boldsymbol{c}_{\boldsymbol{i}}}\|_2} \leq \left\|\ddot{\bar{\boldsymbol{c}}}_{\boldsymbol{i}}\right\|_2 + 2\frac{\left\|\dot{\boldsymbol{p}} - \dot{\bar{\boldsymbol{c}}}_{\boldsymbol{i}}\right\|_2^2}{\|\boldsymbol{p} - \bar{\boldsymbol{c}}_{\boldsymbol{i}}\|_2 - \Delta_{c_i}} \\ \Delta_{b_{\mathrm{c},i}} &= b_{\mathrm{c},i} - \bar{b}_{\mathrm{c},i} \leq |b_{\mathrm{c},i}| - \bar{b}_{\mathrm{c},i} \leq \left\|\ddot{\bar{\boldsymbol{c}}}_{\boldsymbol{i}}\right\|_2 + 2\frac{\left\|\dot{\boldsymbol{p}} - \dot{\bar{\boldsymbol{c}}}_{\boldsymbol{i}}\right\|_2^2}{\|\boldsymbol{p} - \bar{\boldsymbol{c}}_{\boldsymbol{i}}\|_2 - \Delta_{c_i}} - \bar{b}_{\mathrm{c},i} := \Delta_{b_{\mathrm{c},i}}(\Delta_{c_i}) \end{aligned}$$

with $\bar{b}_{\mathrm{c},i} = b_{\mathrm{c},i}(\boldsymbol{p}, \dot{\boldsymbol{p}}, \bar{\boldsymbol{c}}, \dot{\bar{\boldsymbol{c}}})$. Hence, the function bounds required for robust invariance control are

$$\begin{aligned}
\Delta_{h_{\mathrm{c},i}}^{\max_{\mathcal{D}_{\mathrm{rob}}}} &= r_{\max} \\
\Delta_{\dot{h}_{\mathrm{c},i}}^{\max_{\mathcal{D}_{\mathrm{rob}}}} &= |\Delta_{\dot{h}_{\mathrm{c},i}}|^{\max}(r_{\max}) \\
\left|\Delta_{\boldsymbol{a}_{\mathrm{c},i}}\right|_1^{\max_{\mathcal{D}_{\mathrm{rob}}}} &= |\Delta_{a_{\mathrm{c},i,1}}|^{\max}(r_{\max}) \\
\left|\Delta_{\boldsymbol{a}_{\mathrm{c},i}}\right|_2^{\max_{\mathcal{D}_{\mathrm{rob}}}} &= |\Delta_{a_{\mathrm{c},i,2}}|^{\max}(r_{\max}) \\
\Delta_{b_{\mathrm{c},i}}^{\max_{\mathcal{D}_{\mathrm{rob}}}} &= \Delta_{b_{\mathrm{c},i}}(r_{\max})
\end{aligned}$$

whereas the bounds for probabilistic invariance control are

$$\begin{aligned}
\Delta_{h_{\mathrm{c},i}}^{\max_{\mathcal{D}_{p_{\mathrm{pr}}}}} &= r_{\mathrm{pr}} \\
\Delta_{\dot{h}_{\mathrm{c},i}}^{\max_{\mathcal{D}_{p_{\mathrm{pr}}}}} &= |\Delta_{\dot{h}_{\mathrm{c},i}}|^{\max}(r_{\mathrm{pr}}) \\
\left|\Delta_{\boldsymbol{a}_{\mathrm{c},i}}\right|_1^{\max_{\mathcal{D}_{p_{\mathrm{pr}}}}} &= |\Delta_{a_{\mathrm{c},i,1}}|^{\max}(r_{\mathrm{pr}}) \\
\left|\Delta_{\boldsymbol{a}_{\mathrm{c},i}}\right|_2^{\max_{\mathcal{D}_{p_{\mathrm{pr}}}}} &= |\Delta_{a_{\mathrm{c},i,2}}|^{\max}(r_{\mathrm{pr}}) \\
\Delta_{b_{\mathrm{c},i}}^{\max_{\mathcal{D}_{p_{\mathrm{pr}}}}} &= \Delta_{b_{\mathrm{c},i}}(r_{\mathrm{pr}}) \ .
\end{aligned}$$

Corrective control is then determined using Theorem 4.2 or Theorem 4.5.

For scenario invariance control, the required amount of instances of the uncertainty are randomly generated according to the probability distribution of the uncertainties Δ_r and ρ in (4.32). Each instance is used to generate one constraint and corrective control is then determined using Theorem 4.8.

4.5.3 Implementation

The simulation model is implemented in *Matlab/Simulink*. The elements of the ideal and undisturbed trajectories $\boldsymbol{p}_{\mathrm{des}}(t), \bar{\boldsymbol{c}}_{\mathbf{1}}(t), \bar{\boldsymbol{c}}_{\mathbf{2}}(t) \in \mathbb{R}^2$ are splines, which are determined using

$$q\,(t) = (1 - d_3)\, q_{0,i} + d_3 q_f + \left(d_3 - d_3^2\right)(d_1\,(1 - d_3) + d_2 d_3)$$

$$d_1 = t_T \dot{q}_0 - (q_f - q_0) \ , \qquad d_2 = -t_T \dot{q}_f + (q_f - q_0) \ , \qquad d_3 = \frac{t}{t_T}$$

with $q \in \{p_{\mathrm{des}}, \bar{c}_1, \bar{c}_2\}$, the duration of the motion t_T and 0, f indicating initial and final values, respectively. The results are generated using the parameters in Table 4.1. With the chosen parameters, scenario invariance control generates $N_i = 87$ conditions per constraint to derive corrective control according to Theorem 4.8.

Each control scheme is tested for two values of the distribution parameter α and for 1000 different randomly generated constraint trajectories for each constraint. In order to verify the confidence parameter β for scenario invariance control, 200 different sets of scenarios are randomly generated and each is tested with the 1000 constraint trajectories.

Table 4.1.: Model and control parameters for the numerical evaluation of invariance control with uncertain constraints.

General		
Step size	T_A	$0.001\,\mathrm{s}$
Simulation time	t_{end}	$50\,\mathrm{s}$
Nominal control		
Stiffness	$\boldsymbol{K_P}$	$100 \cdot \boldsymbol{I_2} \cdot 1/\mathrm{s}^2$
Damping	$\boldsymbol{K_D}$	$20 \cdot \boldsymbol{I_2} \cdot 1/\mathrm{s}$
System		
Size	n, m	4, 2
Radius	r_x	$1\,\mathrm{m}$
Initial, final position	$\boldsymbol{p_0}$, $\boldsymbol{p_f}$	$[0,\ 0]^\intercal\,\mathrm{m}$, $[10,\ 10]^\intercal\,\mathrm{m}$
Initial, final velocity	$\dot{\boldsymbol{p}}_{\boldsymbol{0}}$, $\dot{\boldsymbol{p}}_{\boldsymbol{f}}$	$[0,\ 0]^\intercal\,\mathrm{m/s}$, $[0,\ 0]^\intercal\,\mathrm{m/s}$
Constraints		
Radius	r_1, r_2	$1\,\mathrm{m}$, $1\,\mathrm{m}$
Initial position	$\bar{\boldsymbol{c}}_{\boldsymbol{10}}$, $\bar{\boldsymbol{c}}_{\boldsymbol{20}}$	$[5,\ -1]^\intercal\,\mathrm{m}$, $[10,\ 2]^\intercal\,\mathrm{m}$
Final position	$\bar{\boldsymbol{c}}_{\boldsymbol{1f}}$, $\bar{\boldsymbol{c}}_{\boldsymbol{2f}}$	$[0,\ 10]^\intercal\,\mathrm{m}$, $[6,\ 10]^\intercal\,\mathrm{m}$
Initial velocity	$\dot{\bar{\boldsymbol{c}}}_{\boldsymbol{10}}$, $\dot{\bar{\boldsymbol{c}}}_{\boldsymbol{20}}$	$[0,\ 0]^\intercal\,\mathrm{m/s}$, $[0,\ 0]^\intercal\,\mathrm{m/s}$
Final velocity	$\dot{\bar{\boldsymbol{c}}}_{\boldsymbol{1f}}$, $\dot{\bar{\boldsymbol{c}}}_{\boldsymbol{2f}}$	$[0,\ 0]^\intercal\,\mathrm{m/s}$, $[0,\ 0]^\intercal\,\mathrm{m/s}$
Motion duration	t_T	$40\,\mathrm{s}$
Position uncertainty	$r_{\max}$	$0.5\,\mathrm{m}$
	α	1 or 5
Invariance control		
Parameters	γ_1, γ_2	$-50 \cdot 1/\mathrm{s}^2$, $-50 \cdot 1/\mathrm{s}^2$
Satisfaction probability	p_{pr}	0.9
Confidence parameter	β	0.1

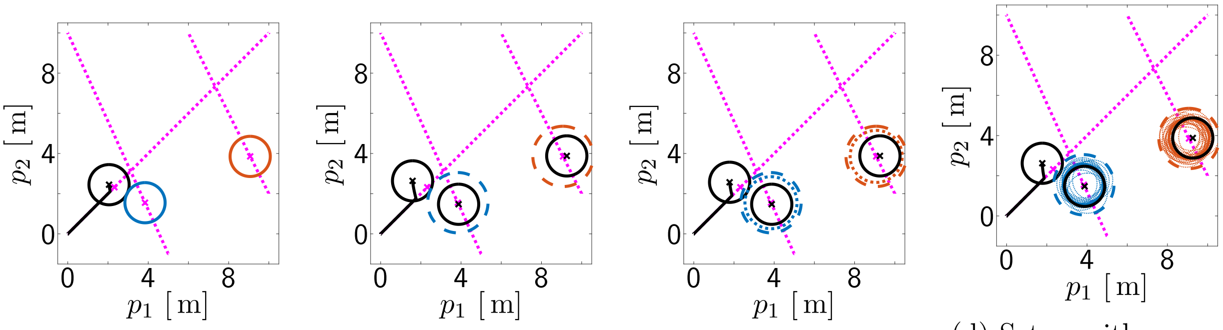
4.5.4 Results

The computation times of the individual approaches implemented in *Matlab/Simulink* and executed on a single core of a commercially available laptop computer with an Intel core i5 processor and 8.00 GB RAM are provided in Table 4.2. For all three approaches, the computation time is less than the simulation time t_{end}, hence the real-time requirement is met. In addition it may be noticed that robust and probabilistic invariance control have considerably shorter computation times than scenario invariance control. This is owed to the fact that the scenario-based approach with the chosen parameters includes N_i times more constraints than the other two approaches.

The configuration of system and constraints at $t = 12.5\,\mathrm{s}$ is depicted in Fig. 4.2. While, for comparison, Fig. 4.2a shows the behavior of system and constraints in the absence of uncertainties, i.e. if it is possible to apply invariance control from Section 3.1, Fig. 4.2b depicts the result of robust invariance control, where the systems avoids the violation of constraints

Table 4.2.: Computation time.

Robust	$< 5\,\mathrm{s}$
Probabilistic	$< 5\,\mathrm{s}$
Scenario	$< 40\,\mathrm{s}$



(a) Disturbance-free setup.

(b) Robust setup.

(c) Setup with p_{pr}-bounds ······, ······.

(d) Setup with scenarios ······, ······.

Figure 4.2.: Setup of system —— and random constraint trajectories (constraint 1 —— and constraint 2 ——) for $\alpha = 5$ at $t = 12.5\,\mathrm{s}$ with the respective undisturbed center positions ×, undisturbed trajectories ······ and maximum displacements (robust bound) - - - -, - - - -.

for all possible uncertainty values. This means it stays out of the sphere containing all possible constraints depicted by the dashed lines. In Fig. 4.2c and Fig. 4.2d, the behavior for probabilistic and scenario invariance control is illustrated including the probabilistic bound containing 90% of the uncertain constraints and the scenarios used for control derivation. It may be observed that these approaches do not cover all possible uncertainties and therefore may lead to violations but also to a less restrictive behavior.

Table 4.3 provides an insight into the results generated by the different control schemes. It shows the percentage of positive output functions, i.e. violated constraints, and positive invariance functions for 1000 random constraint trajectories for robust, probabilistic and scenario invariance control. As scenario invariance control is evaluated with 200 different sets of scenarios, minimum, maximum and average violation percentages are provided. Note that the percentage of violations of the invariance function Φ_i may be slightly higher than that for the output function h_i. This is due to the fact that, by design, the set of states for which $h_{\mathrm{c},i} \leq 0$ holds is a subset of $\Phi_i \leq 0$ with Φ_i according to (3.10).

For robust invariance control, no violations of the constraints are expected, i.e. all output functions should keep non-positive values, since all instances of the uncertainty are considered. For $\alpha = 1$, these expectations are fulfilled. However, there is a single trial for $\alpha = 5$, in which $h_{\mathrm{c},2}$ and Φ_2 take positive values. As the theory is derived for continuous systems and the simulation is carried out with small but discrete time steps, chattering occurs. In this specific trial, the constraint is generated with almost the maximum possible uncertainty thus being close to the robust bound. As a result, it is affected by chattering which leads to negligible violations of less than 1 mm. This effect may be avoided by applying the chatter-

Table 4.3.: Percentage of positive outputs (4.29) and positive invariance functions (3.10) for robust and probabilistic constraint satisfaction with 1000 constraint trajectories.

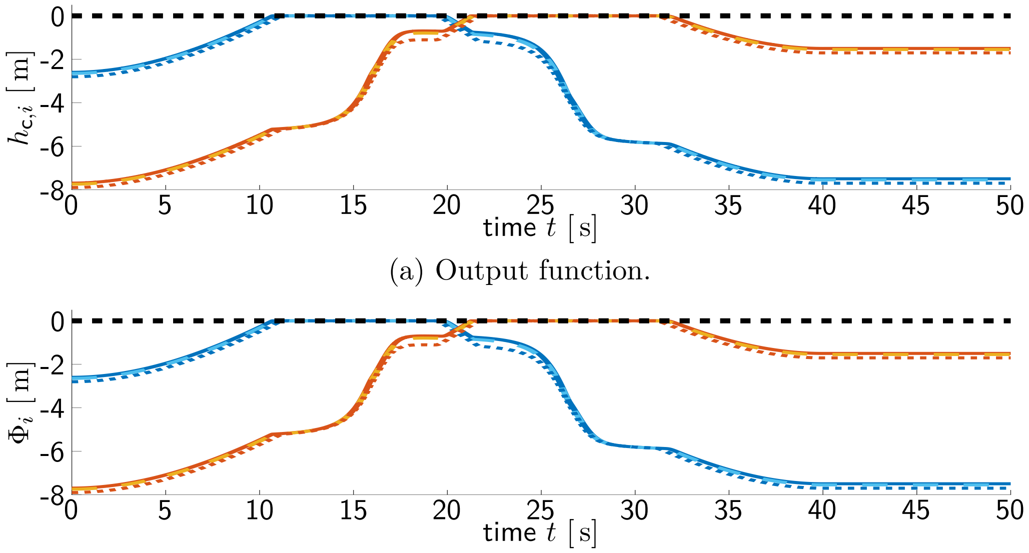
		$h_{c,1} > 0$	$\Phi_1 > 0$	$h_{c,2} > 0$	$\Phi_2 > 0$
Robust					
$\alpha = 1$		0.0 %	0.0 %	0.0 %	0.0 %
$\alpha = 5$		0.0 %	0.0 %	0.1 %	0.1 %
Probabilistic					
$\alpha = 1$		6.2 %	6.3 %	4.9 %	4.9 %
$\alpha = 5$		5.3 %	5.3 %	5.3 %	5.3 %
Scenario					
$\alpha = 1$	minimum	3.0 %	3.0 %	3.2 %	3.3 %
	maximum	14.5 %	14.5 %	14.4 %	14.4 %
	average	7.6 %	7.6 %	6.7 %	6.7 %
$\alpha = 5$	minimum	0.6 %	0.6 %	1.6 %	1.6 %
	maximum	10.0 %	10.0 %	11.2 %	11.2 %
	average	3.8 %	3.8 %	4.4 %	4.4 %

ing reduction method proposed in Sec. 3.2.2 or by slightly increasing the boundary values of the disturbance for robust invariance control.

For probabilistic invariance control, less than 10% of the constraints are violated, i.e. satisfaction is achieved in more than 90% of the cases, which corresponds to the chosen satisfaction probability $p_{\mathrm{pr}} = 0.9$ for both constraints.

These results are supported by Fig. 4.3, which depicts the output and invariance function of an enlarged constraint with center position $\bar{\boldsymbol{c}}_{\boldsymbol{i}}$ and radius $r_{\mathrm{rob}} = r_i + r_{\mathrm{max}}$ or $r_p = r_i + r_{\mathrm{pr}}$ with r_{pr} from (4.34). Due to the design of the constraints and the uncertainty, all instances of the uncertain constraint lie within the radius r_{rob} and with probability p_{pr} within the radius r_p. Since the output and constraint functions in Fig. 4.3 take non-positive values, except for small violations $< 1\,\mathrm{mm}$, which are not visible in the figure and are due to the previously mentioned chattering, the desired robust and probabilistic satisfaction properties are achieved. It is also noticeable that for probabilistic invariance, the functions take smaller values for increasing α, which is caused by the decrease in the radius r_p. For robust invariance, the values do not change with α since the radius r_{rob} is independent from the actual distribution, but solely depends on the bounds thus being more restrictive than probabilistic invariance control.

For scenario invariance control, the minimum and average violation percentages in Tab. 4.3 are less than 10%, which fits the chosen satisfaction probability $p_{\mathrm{pr}} = 0.9$ for both constraints. However, some of the maximum violation percentages are higher than 10%. In these cases, the randomly generated scenario is not representative enough to avoid violations to the desired extent. This is covered by the confidence parameter β, i.e. the probability of generating such bad scenarios, which should be less than β. The evaluation of all 200 scenarios in Tab. 4.4 shows that the percentage of scenarios, for which more than 10% of the output or invariance functions are violated, is less than 10% thus corresponding to $\beta = 0.1$.



(a) Output function.

(b) Invariance function.

Figure 4.3.: Value of the (a) constraint outputs and (b) invariance functions for —— robust satisfaction of $h_{\mathrm{c},1}$, —— robust satisfaction of $h_{\mathrm{c},2}$, - - - - p_{pr}-satisfaction of $h_{\mathrm{c},1}$ with $\alpha = 1$, - - - - p_{pr}-satisfaction of $h_{\mathrm{c},2}$ with $\alpha = 1$, ······ p_{pr}-satisfaction of $h_{\mathrm{c},1}$ with $\alpha = 5$ and ······ p_{pr}-satisfaction of $h_{\mathrm{c},2}$ with $\alpha = 5$.

Table 4.4.: Percentage of scenarios violating the desired satisfaction probability p_{pr} with scenario invariance control for 1000 constraint trajectories and 200 scenarios.

	$\mathscr{P}(h_{\mathrm{c},i} > 0) > 1 - p_{\mathrm{pr}}$ or $\mathscr{P}(\Phi_i > 0) > 1 - p_{\mathrm{pr}}$			
	$h_{\mathrm{c},1}$	Φ_1	$h_{\mathrm{c},2}$	Φ_2
$\alpha = 1$	2.7 %	2.7 %	0.4 %	0.4 %
$\alpha = 5$	0.0 %	0.0 %	0.1 %	0.1 %

Hence, (p_{pr}, β)-satisfaction is achieved for both constraints and both α-distributions.

4.6 Discussion

Concluding this chapter, we provide a short summary of the capabilities of the proposed control approaches for the satisfaction of constraints with uncertain parameters. Robust, probabilistic and scenario-based constraint satisfaction may be achieved by using both invariance control or CBF-based control. As the methods may be combined, this enables the choice of a different satisfaction and control type for each constraint, which makes it applicable in a variety of applications.

Each of the three methods has its individual requirements and advantages. Therefore, the choice of the satisfaction type for each constraint depends on the application, the desired outcome, the available knowledge about the uncertainties and the available computational power to solve the optimization. While robust constraint satisfaction enforces the constraints for all possible uncertainties, it may turn out to be rather conservative for flat distributions.

Table 4.5.: Characteristics of the three satisfaction types for l uncertain state constraints.

	Robust	Probabilistic	Scenario
Guaranteed constraint satisfaction	robust	p_i	(p_i, β)
Restrictiveness for flat uncertainty distributions	high (conservativeness)	medium (overestimation)	medium - low (depending on scenario configuration)
Knowledge of uncertainty distribution	max/min values	required	helpful (data available) / required (no data)
Usage with unbounded distributions	no	yes	yes
Data-driven control	no guarantees	no guarantees	possible
Preliminary computations	min/max of Δ_k (Lemma 4.1)	min/max of Δ_k (Lemma 4.2)	scenarios, if derived from probability distribution
Number of optimization conditions...	at most l	at most l	$\sum_{i=1}^{l} N_i$ (Theorem 4.8)
... scales with m	no	no	linearly
... scales with p_i	no	no	linearly with $1/(1-p_i)$
... scales with β_i	no	no	logarithmically with $1/\beta_i$

Here the probabilistic approaches have an advantage as they provide narrower bounds, then of course with probabilistic guarantees only. Note that as probabilistic constraint satisfaction is based on an estimation of probabilistic function bounds, the resulting behavior may, depending on the form of the uncertainties, still be more conservative than the results from the scenario approach. On the other hand the number of samples required for scenario-based constraint satisfaction depends strongly on the choice of probabilities, while the robust and probabilistic approaches do not change the number of constraints. A detailed comparison of the characteristics of the individual approaches is provided in Table 4.5.

5

Sharing Control in Multi-Agent Systems

The simultaneous employment of multiple robotic systems to relieve and aid humans proves challenging as each robot is simultaneously required to execute a task that is connected with a certain role while limiting its influence on other robots and ensuring the safe interaction with other robots, humans, and the environment. As the tasks carried out by the agents may be of different importance, a prioritization scheme for multiple agents should be applied to share evasive actions between agents.

Related Work and Open Problems

Using established methods such as MPC by [MRR+00] or the methods introduced in Chapter 3 to enforce constraints on multi-agent systems, generally, do not consider the option of sharing evasive actions between agents in multi-agent systems according to given priorities. Approaches designed specifically for collision avoidance in multi-agent systems such as presented in [Ros96; CYZ+07; LJX15], are either unable to guarantee constraint satisfaction, do not allow for a dynamic prioritization of the agents or may result in agents blocking the path.

The required priorities may, for example, be determined by a heuristic approach, which changes the priorities according to the circumstances and aims at mimicking human behavior [BCL16]. Other prioritization schemes are found in traffic management [AMP+11] and message scheduling for multi-agent systems [HPS+09], where the priorities are derived from the message importance and throughput. If the goal is to grant the agents access to a shared resource, the possibilities range from deterministic weighted round robin scheduling [KSC91] to probabilistic approaches like the lottery approach [TCG16]. However, these methods do not ensure a prioritization which guarantees constraint satisfaction and avoids trapping agents with low priorities.

Based on the findings on shared invariance control [KPW+ed], this chapter introduces shared constraint enforcement for the I/O-linearization-based control approaches from Chapter 3. Using priority-based weightings in the minimization (3.87) to divide the control effort between the agents allows for a distributed implementation by generating independent optimization problems. The agent priorities are determined by a two-stage prioritization scheme, which allots the priorities such that trap situations are resolved and a variety of factors may be included in the prioritization. The approach enables agents with high priorities to carry on undisturbed from other agents, while agents with low priority never end up in a situation where they are trapped by other agents, unable to avoid constraint violations.

5.1 Shared Constraint Satisfaction

Naturally, the control schemes introduced in Chapter 3 may be implemented on multiple agents simultaneously to enforce environmental and inter-agent constraints by using a dynamic model of the other agents in the constraint parametrization. This approach leads, however, to a rather conservative behavior of the agents as each one tries to avoid constraint violations at any cost which is especially disadvantageous for robots in narrow environments. Alternatively, the constraint may be enforced only by a subset of the involved agents, which may lead to the optimization being infeasible if the evading agent is unable to act, e.g. due to being trapped between constraints.

These drawbacks are resolved in this section by introducing a control approach for multi-agent systems based on the following assumption.

Assumption 5.1. *The I/O-linearization of each constraint $y_{c,i}$, $i \in \mathcal{B}$ from* (2.11) *with respect to the input $\boldsymbol{u}_j$ of each agent $j \in \mathcal{N}_{ag}$ yields the same relative degree r_i for all agents j.*

This assumption is imposed for convenience of notation and poses no additional restrictions on the systems, since the relative degree may be increased, if necessary, by augmented invariance control as introduced in Sec. 3.2.1. This also applies to CBF-based control as discussed in Remark 3.14. Note that for robotic systems with position constraints, for example, the assumption is naturally fulfilled as the relative degree equals two for all agents.

Building on the naturally shared actions achieved by a centralized implementation, shared constraint satisfaction allows for a distributed implementation based on given priorities.

5.1.1 Centralized Implementation

Concatenating states and inputs of the n_{ag} systems with the respective dynamics given in (2.3) yields the centralized system description

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}$$

in accordance with the dynamics of a single system (2.1), where

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_{n_{\text{ag}}} \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{n_{\text{ag}}} \end{bmatrix}, \quad \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{f}_{n_{\text{ag}}}(\boldsymbol{x}_{n_{\text{ag}}}) \end{bmatrix},$$

$$\boldsymbol{G}(\boldsymbol{x}) = \operatorname{diag}\left(\boldsymbol{G}_i(\boldsymbol{x}_i)|_{\forall\ i \in \mathcal{N}_{\text{ag}}}\right) = \begin{bmatrix} \boldsymbol{G}_1(\boldsymbol{x}_1) & \boldsymbol{0}_{n_1 \times m_2} & \dots & \boldsymbol{0}_{n_1 \times m_{n_{\text{ag}}}} \\ \boldsymbol{0}_{n_2 \times m_1} & \ddots & & \vdots \\ \vdots & & \ddots & \boldsymbol{0}_{n_{n_{\text{ag}}-1} \times m_{n_{\text{ag}}}} \\ \boldsymbol{0}_{n_{n_{\text{ag}}} \times m_1} & \dots & \boldsymbol{0}_{n_{n_{\text{ag}}} \times m_{n_{\text{ag}}-1}} & \boldsymbol{G}_{n_{\text{ag}}}(\boldsymbol{x}_{n_{\text{ag}}}) \end{bmatrix}. \tag{5.1}$$

Thus, the centralized system may be considered as a single agent, allowing control to be derived using invariance control or CBF-based control as introduced in Chapter 3. This requires the involved agents to be controlled by a single, centralized controller, which has access to the agent states, dynamic information and constraint parameters. The structure for two agents is illustrated in Fig. 5.1. The control input is generated using (3.87) and

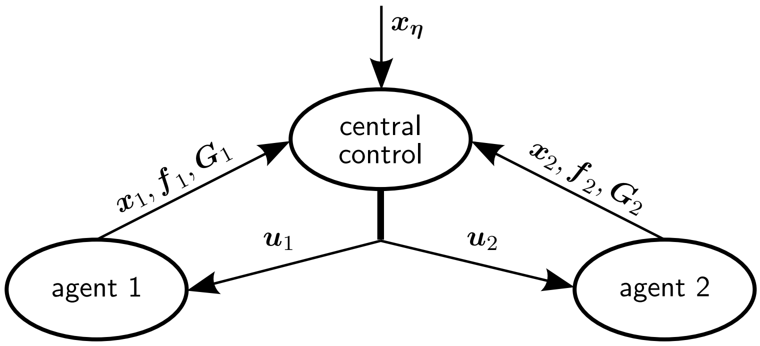

$\boldsymbol{x}_\eta$
central control
$\boldsymbol{x}_1, \boldsymbol{f}_1, \boldsymbol{G}_1$
$\boldsymbol{x}_2, \boldsymbol{f}_2, \boldsymbol{G}_2$
$\boldsymbol{u}_1$
$\boldsymbol{u}_2$
agent 1
agent 2

Figure 5.1.: Centralized control architecture for inter-agent constraints of two agents.

the centralized system description. The control action is automatically shared between the agents as the left side of the optimization condition

$$\boldsymbol{a}_i^\intercal \boldsymbol{u} + b_i = \begin{bmatrix} \boldsymbol{\alpha}_{i,1}^\intercal \\ \vdots \\ \boldsymbol{\alpha}_{i,n_{\text{ag}}}^\intercal \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{n_{\text{ag}}} \end{bmatrix} + b_i = \sum_{j \in \mathcal{N}_{\text{ag}}} \boldsymbol{\alpha}_{i,j}^\intercal \boldsymbol{u}_j + b_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}}) \tag{5.2}$$

and $\boldsymbol{a}_i$ and b_i as defined in (3.87) includes the input $\boldsymbol{u}_j$ of each agent system affected by the constraint due to Assumption 5.1. Hence, all agents react to the constraint, the evasive effort is partitioned based on the agent dynamics and constraint satisfaction is guaranteed by Theorem 3.1 or Theorem 3.7.

In this rather straightforward approach, the agents have to be controlled by a common central control instance even if no inter-agent constraints are active. For more independence of the agents, a distributed implementation is preferable. Furthermore, any prioritization of the agents is lost in the centralized implementation, since the priorities have no influence on the allocation of the evasive effort.

5.1.2 Distributed Implementation

Introducing shared constraint satisfaction, we aim at a distributed implementation as depicted for two agents in Fig. 5.2, where each agent has its own control loop of system dynamics and control law. Each agent is assumed to have knowledge of the constraint parameters $\boldsymbol{x}_\eta$ and the dynamic information of the other agents either through explicit communication or via observation. In addition each agent is assigned a priority.

Definition 5.1. The ***priority*** c_i of an agent $i \in \mathcal{N}_{\text{ag}}$ fulfills

$$c_i \in \left]0,\ c_{\max}\right]$$

with a maximum priority value $c_{\max} \in \mathbb{R}^+$.

A framework how to assign such priorities is introduced in the following section. For now, the priorities are assumed to be given.

The goal of shared constraint satisfaction is to share the evasive control action between the agents according to their priorities, i.e. high effort for low-priority agents and low or no effort for high-priority agents. This is achieved by partitioning the agents into different groups based on their priority.

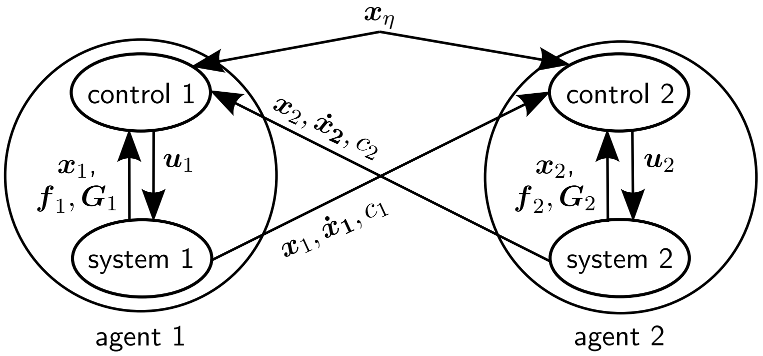


Figure 5.2.: Distributed control architecture of shared constraint satisfaction for inter-agent constraints of two agents.

Definition 5.2. An ***agent priority community*** $\mathcal{P}_j$ consists of the agents, whose priorities c_i fulfill $s_j < c_i \leq s_{j+1}$ with positive constants $s_j,\ s_{j+1} > 0$ and is defined as

$$\mathcal{P}_j = \{i \in \mathcal{N}_{\mathrm{ag}} \mid s_j < c_i \leq s_{j+1},\ s_j, s_{j+1} > 0\}.$$

Note that the agent priority communities have to be chosen such that each agent belongs to exactly one community. As only the agents with the lowest priority should actively enforce the constraints and share the effort, it is necessary to find these agents and the related priority community.

Definition 5.3. The ***minimal active priority community*** $\mathcal{P}_{\mathrm{min},i}$ for constraint i denotes the lowest priority community, which contains agents that are affected by the constraint. The community $\mathcal{P}_{\mathrm{min},i} = \mathcal{P}_{j_{\mathrm{min}}}$ fulfills the following two properties.

$$\begin{aligned} &\exists\ j \in \mathcal{P}_{\mathrm{min},i} :\ \boldsymbol{\alpha}_{i,j}^{\mathsf{T}} \neq \mathbf{0}_{1\times m_j}\ , \\ &\forall\ k < j_{\mathrm{min}},\ j \in \mathcal{P}_k :\ \boldsymbol{\alpha}_{i,j}^{\mathsf{T}} = \mathbf{0}_{1\times m_j}\ . \end{aligned}$$

Using the minimal agent priority community, it is possible to find those agents, which should actively pursue constraint enforcement, i.e. those agents with the lowest priorities, which are affected by a constraint.

Definition 5.4. The ***set of active agents*** $\mathcal{A}_i$ for constraint i denotes those agents in the minimal agent priority community, which are affected by the constraint

$$\mathcal{A}_i = \{j \in \mathcal{P}_{\mathrm{min},i} \mid \boldsymbol{\alpha}_{i,j}^{\mathsf{T}} \neq \mathbf{0}_{1\times m_j}\}\ .$$

The remaining agents, which are not in the set of active agents, are either not affected by the constraint at all or have high priorities and should therefore not carry out any evasive action. These agents are collected in the set of inactive agents.

Definition 5.5. The ***set of inactive agents*** $\mathcal{J}_i$ for constraint i denotes the complimentary set to the set of active agents

$$\mathcal{J}_i = \mathcal{N}_{\text{ag}} \setminus \mathcal{A}_i \ .$$

Using these considerations, it is possible to introduce shared constraint satisfaction for multi-agent systems.

Theorem 5.1. *Let the agent dynamics and constraints be given by* (2.3) *and* (2.11)*, respectively. Let Assumptions 2.1, 2.2, 2.6, 5.1 hold. Let $B_i(\boldsymbol{x}, \boldsymbol{x}_\eta)$ be CBFs corresponding to each constraint function $y_{B,i}$, $i \in \mathcal{B}$. Then, if the optimization*

$$\boldsymbol{u}_{c,j} = \underset{\boldsymbol{u}_j}{\operatorname{argmin}} \ ||\boldsymbol{u}_j - \boldsymbol{u}_{no,j}||_2^2 \tag{5.3}$$

$$s.t. \ \boldsymbol{\alpha}_{i,j}^\intercal \boldsymbol{u}_j + w_{i,j}(\boldsymbol{c}) \left(\sum_{k \in \mathcal{J}_i} (\boldsymbol{\alpha}_{i,k}^\intercal \boldsymbol{u}_k) + b_i \right) \leq w_{i,j}(\boldsymbol{c}) \ d_i \quad \forall \ i \in \tilde{\mathcal{B}}, \ j \in \mathcal{A}_i \tag{5.4}$$

with $\boldsymbol{\alpha}_{i,j}^\intercal$ according to (5.2)*, b_i, d_i, $\tilde{\mathcal{B}}$ as in* (3.87)*, the priorities $\boldsymbol{c} = [c_1, \ \ldots, \ c_{n_{ag}}]^\intercal$ and weights $w_{i,j}(\boldsymbol{c}) : [0, \ c_{\max}]^{n_{ag}} \to [0, 1]$ fulfilling*

$$\sum_{j \in \mathcal{A}_i} w_{i,j}(\boldsymbol{c}) = 1 \tag{5.5}$$

yields a solution for each agent $j \in \mathcal{N}_{ag}$ and all $t \geq t_0$, the control inputs $\boldsymbol{u}_j = \boldsymbol{u}_{c,j}$ avoid any constraint violation.

The proof is provided in the appendix. The theorem provides means to share the evasive control actions between the lowest priority agents. A solution exists if the constraints on each agent are non-contradicting. Furthermore, for robotic systems it is essential to avoid singularities as they result in a loss of controllability. Both conditions may be ensured by including these criteria in the determination of priorities.

Remark 5.1. The way of dividing the optimization condition between the agents is not unique. In general the optimization conditions may be arbitrarily shared among the agents as long as the the weighted conditions (5.4) sum up to the centralized condition (5.2). This is especially useful to reduce the communication between agents if the condition is divided such that parts of the dynamics do not need to be known by the other agents.

As the control actions of the agents just suffice to fulfill the optimization condition, this avoids overly restrictive reactions to the constraints, which would occur if the constraints are fully enforced by all agents. Therefore, the approach lends itself to the implementation on robotic systems acting in narrow environments.

Remark 5.2. As environmental constraints act on each agent separately, each agent is always the sole active agent for an environmental constraint. Therefore, it carries out the entire evasive action and the satisfaction is not affected by the shared implementation.

Theorem 5.1 does not give any indication as to how the weighting factors $w_{i,j}(\boldsymbol{c})$ are determined as a function of the priority vector $\boldsymbol{c} = [c_1, \ldots, c_{n_{\mathrm{ag}}}]^\intercal$. The most straightforward definition of the weight function is given by

$$w_{i,j}(\boldsymbol{c}) = \frac{\frac{1}{c_j}}{\sum_{j \in \mathcal{A}_i} \frac{1}{c_j}} \,. \tag{5.6}$$

By inverting the priorities the agents with the lowest priority in the set of active agents are allocated the highest weighting factor, which means that they are affected most by the constraint. Normalizing the inverted priority ensures that condition (5.5) is fulfilled. Naturally, other weighting functions, possibly with a saturation may be used as well. Note that shared constraint satisfaction exhibits some useful properties.

Corollary 5.1.1. *The optimization* (5.3) *is strictly convex.*

The proof is provided in the appendix. Due to the convexity of the optimization problem, efficient solvers may be used thus allowing the application on real-time systems with fast sampling times.

Corollary 5.1.2. *Let the state values of all agents and the inputs of the agents in the set of inactive agents $j \in \mathcal{I}_i$ be known to the active agents $j \in \mathcal{A}_i$. Then, the constrained optimization problems* (5.3) *for shared constraint satisfaction may be solved in a distributed manner.*

The proof is provided in the appendix. As the optimization may be solved in distributed fashion, this allows for the desired control structure illustrated in Fig. 5.2, where each agent determines the control input based on its priority.

5.2 Agent Prioritization

Theorem 5.1 does not ensure that the constrained optimization (5.3) is feasible for all agents. Therefore, the prioritization scheme has to be designed to account for the feasibility, since there might occur situations in which the agents trap each other leading to infeasibility. Figure 5.3 shows an exemplary situation with four spherical agents with fixed priorities in two dimensional space, which may represent mobile robotic systems. The innermost agent holds a static position, while the outer agents move in the directions indicated by the solid arrows. In order to avoid collisions, each pair of agents defines one inter-agent constraint. Since they are in different agent priority communities, the lower prioritized agent takes the entire evasive action as indicated by the dotted arrows. The inner agent is in the lowest priority community and has to avoid collisions with all other agents. However, the outer agents leave no space to escape this trap situation which eventually leads to them crashing into the inner agent.

5.2.1 Two-stage Prioritization Scheme

In order to resolve such trap situations, we propose a prioritization scheme comprised of two stages as depicted in Fig. 5.4. The first stage assigns a priority community according

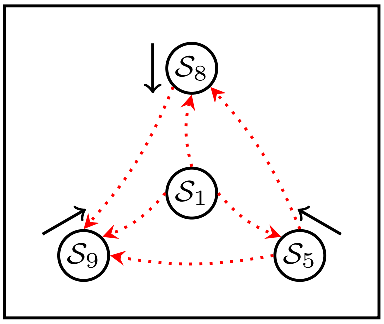


Figure 5.3.: Four agent scenario with different fixed priorities with ⟶ the direction of motion and ⇢ the distribution of the shared inter-agent constraints.

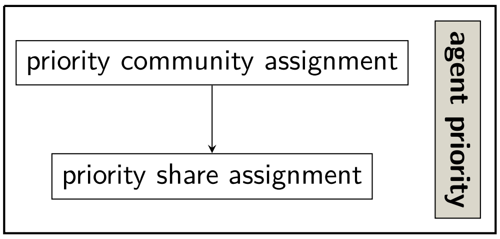


Figure 5.4.: Two-stage prioritization scheme.

to Def. 5.2. The priority community decides which agents actively enforce constraints. This stage ensures that the optimization problem (5.3) remains feasible for each agent. Note that as the situation may change over time, the priority assignment needs to be monitored constantly. The second stage takes care of the priority fine-tuning. Taking individual factors into account the priorities of agents within one priority community are assigned to share the effort accordingly.

5.2.2 Priority Community Assignment

The goal of assigning agents to a priority community is to ensure the feasibility of the optimization in Theorem 5.1. This requires the detection of trap situations leading to infeasibility and a trap handling scheme to adjust the priorities.

Trap Detection

Before introducing the approach for trap detection, we define the meaning of an agent being trapped.

Definition 5.6. An agent is ***trapped*** if the optimization (5.3) in Theorem 5.1 is infeasible and ***free*** otherwise.

More formally, if the set

$$\mathcal{M}_j = \left\{ \boldsymbol{u}_j \;\middle|\; \boldsymbol{\alpha}_{i,j}^{\intercal} \boldsymbol{u}_j \leq \beta_{i,j} \;\forall\; i \in \mathcal{B}_{\text{act,ag}}(j) \right\} , \tag{5.7}$$

$$\text{with } \beta_{i,j} = w_{i,j}(\boldsymbol{c}) \left(d_i - \sum_{j \in \mathcal{J}_i} \boldsymbol{\alpha}_{i,j}^{\intercal} \boldsymbol{u}_j + b_i \right) ,$$

$$\mathcal{B}_{\text{act,ag}}(j) = \{ i \mid i \in \tilde{\mathcal{B}}, j \in \mathcal{A}_k \} \tag{5.8}$$

and $\tilde{\mathcal{B}}$, d_i, b_i as defined in (3.87), which considers all relevant constraints for agent j, is empty, the optimization problem is infeasible and the agent is trapped. Stacking the individual inequations in (5.7) yields the element-wise matrix inequality $\boldsymbol{A}_j \boldsymbol{u}_j \preccurlyeq \boldsymbol{\beta}_j$, which enables the use of Farkas' lemma to determine whether $\mathcal{M}_j = \emptyset$.

Theorem 5.2 (Farkas' Lemma [Van01, p.167])**.** *The element-wise inequality $\boldsymbol{A}_j \boldsymbol{u}_j \preccurlyeq \boldsymbol{\beta}_j$ has no solution if and only if there exists a vector $\boldsymbol{z}$ fulfilling $\boldsymbol{z} \succcurlyeq \boldsymbol{0}_{m_j \times 1}$ element-wise such that both $\boldsymbol{A}_j^\intercal \boldsymbol{z} = \boldsymbol{0}_{|\mathcal{B}_{act,ag}(j)| \times 1}$ and $\boldsymbol{\beta}_j^\intercal \boldsymbol{z} < 0$ hold.*

The conditions introduced in Farkas' Lemma may be evaluated by the minimization

$$\min_{\boldsymbol{z}} \boldsymbol{\beta}_j^\intercal \boldsymbol{z}$$
$$\text{s.t. } \boldsymbol{A}_j^\intercal \boldsymbol{z} = \boldsymbol{0}_{|\mathcal{B}_{\mathrm{act,ag}}(j)| \times 1} \ , \quad \boldsymbol{z} \succcurlyeq \boldsymbol{0}_{m_j \times 1} \ .$$

This minimization has a solution as $\boldsymbol{z} = \boldsymbol{0}_{m_j \times 1}$ fulfills both minimization conditions. By applying computationally efficient algorithms from convex linear programming the minimization is solved. If a positive $\boldsymbol{z}$ exists for which the found minimum is negative, the conditions from Farkas' Lemma are fulfilled and agent j is trapped. Otherwise, the minimum is $\boldsymbol{\beta}_j^\intercal \boldsymbol{z} = 0$ and the agent is free.

Naturally, trap situations may only be resolved if reducing the number of constraints is possible, i.e. if they are caused by inter-agent constraints, thus motivating the following assumption.

Assumption 5.2. *The uncontrollable environmental constraints are such that an agent is at no time trapped solely by the environment.*

Trap Handling

Once all trapped agents are identified, the agents taking part in or being close to a trap situation are determined. For robotic systems with position constraints, this corresponds to finding all robots that are physically close or moving fast towards the trapped robot. More generally speaking, the invariance functions provide a measure for evaluating which agents may be considered close.

Definition 5.7. A ***nearly active constraint*** has a non-positive invariance function value fulfilling

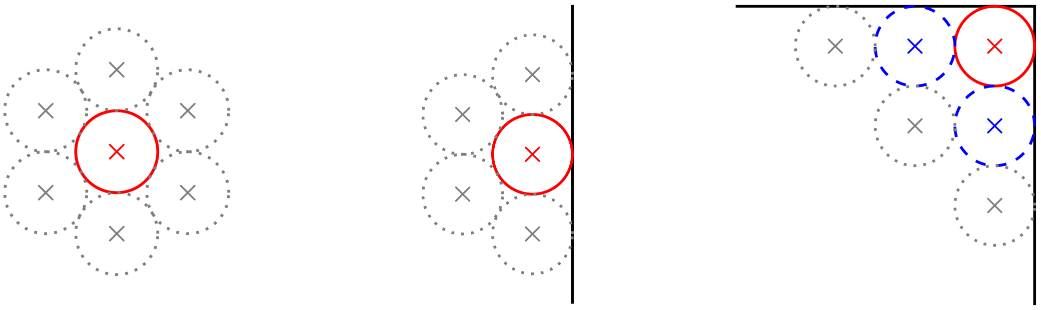
$$0 \geq \Phi_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}}, \gamma_i) \geq \Phi_{\mathrm{th}}$$

with a constant negative threshold $\Phi_{\mathrm{th}} \in \mathbb{R}^-$.

Using the nearly active constraints, an agent graph is established. The agents are nodes and those affected by the same nearly active constraints are connected via edges.

Definition 5.8. An ***emergency community*** $\mathcal{E}$ is a set of agents connected by common nearly active inter-agent constraints, in which at least one of the agents is trapped.

Emergency communities are determined via graph search starting from trapped agents. Within an emergency community, priorities are reassigned to reinstate the feasibility of the optimization (5.3). Naturally, there are various solutions for reassigning priorities to resolve trap situations. We propose a priority community reassignment based on two criteria:



(a) No walls.

(b) Single wall.

(c) Two walls.

Figure 5.5.: Trap situations of different severity with — the most trapped agent restricted by (a) solely free agents ⋯⋯, (b) free agents and a single wall — and (c) two walls and other trapped agents - - - -, which are in turn trapped by free agents and a single wall.

- the severity of the trapping situation depending on the involved constraints,
- the proximity to free agents.

The ***severity*** criterion is determined by the number of involved environmental constraints. In Fig. 5.5 the severity increases from left (solely agents) to right (multiple environmentally constrained dimensions). The more severely an agent is trapped, the higher should be its priority community.

The ***proximity*** criterion is determined by the distance to free agents. Using the agent graph of the emergency community and starting at the free agents, which are sorted into the lowest priority community, the priority community increases for each step along an edge until all agents are assigned.

The trap situation is then resolved by Algorithm 2. By assigning free agents to low priority communities, it removes constraints from the other agents, thus freeing more agents. These are in turn assigned to the higher priority communities, which is repeated until all agents are free. Assuming that there are no constraints that mutually trap two or more agents, i.e. that would need to be removed entirely to free the agents, and that an agent is never trapped solely by environmental constraints, there will always be at least one free agent at each step, thus leading to convergence of the algorithm. In addition, the algorithm never needs more steps than there are agents in the emergency community, but in general it needs considerably less. The process is illustrated in the following example.

Example 5.1. Consider the exemplary trap situation with 8 spherical agents with double integrator dynamics and one environmental constraint in two-dimensional space depicted in Fig. 5.6a. Initially, there are two trapped agents as the surrounding agents are in higher priority communities. Applying Algorithm 2 (lines 1–5), the lowest priority community containing a trapped agent is $\mathcal{P}_1$, meaning that all agents except the one in $\mathcal{P}_0$, are reassigned to $\mathcal{P}_1$. As a result, there are now three trapped agents as depicted in Fig. 5.6b. After finishing the first run of the while-loop, the free agents remain in $\mathcal{P}_1$, while the previously trapped agents are assigned to $\mathcal{P}_2$ as all are restricted by one environmental constraint and are one step from a free agent. Consequently, there remains one trapped agent as illustrated in Fig. 5.6c, since it is not able to evade the outer two agents moving in. After a second run of the while-loop, the outer two agents, which are now free remain in $\mathcal{P}_2$, while the middle agent is reassigned to $\mathcal{P}_3$. In this final configuration, shown in Fig. 5.6d, all agents are free and the priority community assignment for the eight agents is completed after two steps.

All agents that are not part of an emergency community may be assigned either to arbitrary priority communities or preferably based on their priority shares.

5.2.3 Priority Share Assignment

In situations, which are not critical for the feasibility of the optimization problem, the priorities may be assigned freely. In this case, each agent j has a priority c_j which is calculated using $n_{\text{pc}} \in \mathbb{N}$ priority criteria. The criteria coefficients $c_{j,i} \in \mathbb{R}^+$ with $i \in \{1, \ldots, n_{\text{pc}}\}$ should be determined such that they take high values if the criterion is important or highly restrictive, such as for example low manipulability of a robotic manipulator, and low values otherwise. The individual coefficients $c_{j,i}$ are then weighted depending on which criteria should be the most influential and combined to the agent priority

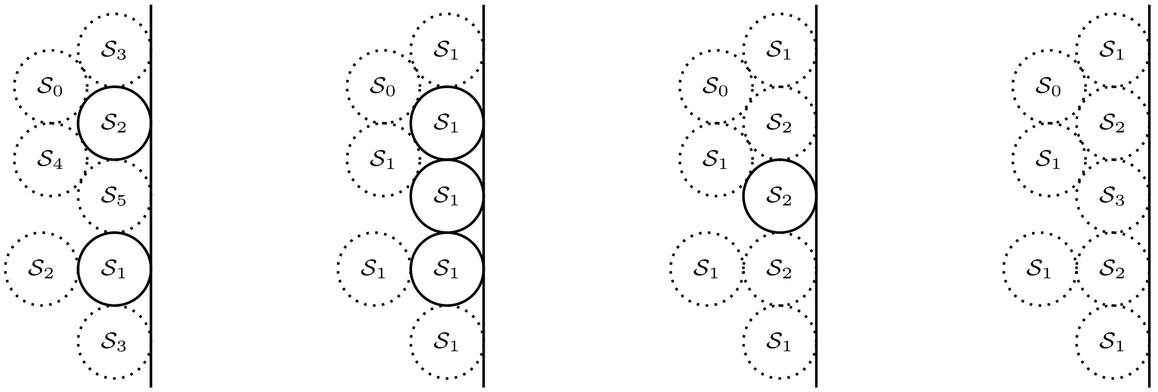
$$c_j = \sum_{i=1}^{n_{\text{pc}}} \omega_{j,i} \cdot c_{j,i} \tag{5.9}$$

with non-negative scalar weights $\omega_{j,i} \in \mathbb{R}^+$.

The proposed priority share calculation framework offers several advantages. It is flexible, scalable and allows the calculation of the agent priorities in a distributed manner. Furthermore, adding new criteria for priority calculation is trivial.

Algorithm 2 Priority Community (PC) Assignment.

```
Determine all trapped agents
Determine emergency community $\mathcal{E}$
while $\exists$ trapped agent do
    Determine $\mathcal{P}_m$: lowest PC containing trapped agent
    Set agent $i \in \mathcal{P}_m \; \forall \; i \in (\mathcal{E} \cap \mathcal{P}_p)$, $p > m$
    for trapped agents do
        Determine $\mathcal{P}_{si}, \mathcal{P}_{sj}$ using severity $\forall \; i, j \in \mathcal{P}_m$
        Determine $\mathcal{P}_{ci}, \mathcal{P}_{cj}$ using proximity $\forall \; i, j \in \mathcal{P}_m$
        Reassign trapped agents $i, j$ to $\mathcal{P}_{pi}, \mathcal{P}_{pj}$ fulfilling
         $pi > m$, $pj > m$ and
        if $si = sj \land ci = cj$ then
            $pi = pj$,
        else if $si > sj$ then
            $pi > pj$,
        else if $si = sj \land ci > cj$ then
            $pi > pj$,
        else
            $pi < pj$
    Determine all trapped agents
```


$\mathcal{S}_3$ $\mathcal{S}_0$ $\mathcal{S}_2$ $\mathcal{S}_4$ $\mathcal{S}_5$ $\mathcal{S}_2$ $\mathcal{S}_1$ $\mathcal{S}_3$
$\mathcal{S}_1$ $\mathcal{S}_0$ $\mathcal{S}_1$ $\mathcal{S}_1$ $\mathcal{S}_1$ $\mathcal{S}_1$ $\mathcal{S}_1$ $\mathcal{S}_1$
$\mathcal{S}_1$ $\mathcal{S}_0$ $\mathcal{S}_2$ $\mathcal{S}_1$ $\mathcal{S}_2$ $\mathcal{S}_1$ $\mathcal{S}_2$ $\mathcal{S}_1$
$\mathcal{S}_1$ $\mathcal{S}_0$ $\mathcal{S}_2$ $\mathcal{S}_1$ $\mathcal{S}_3$ $\mathcal{S}_1$ $\mathcal{S}_2$ $\mathcal{S}_1$

(a) Initial. (b) Reassign. (c) Loop 1. (d) Loop 2.

Figure 5.6.: Trap situation with ⋯⋯ free and —— trapped agents with the respective priority communities in different stages of Algorithm 2.

Remark 5.3. While the priority share only depends on the individual agent characteristics and may therefore be computed in a distributed manner, the priority community assignment is preferably determined by a central computing unit as otherwise extensive communication between the agents would be required.

In the following, the capabilities of the proposed prioritization framework are illustrated in numerical examples and experiments.

5.3 Evaluation

The proposed trap handling and prioritization scheme for shared constraint satisfaction is tested using invariance control, i.e. shared invariance control, in different scenarios in simulation as well as in experiments on robotic manipulators. For both the numerical examples and the experiments, control is implemented in *Matlab/Simulink*.

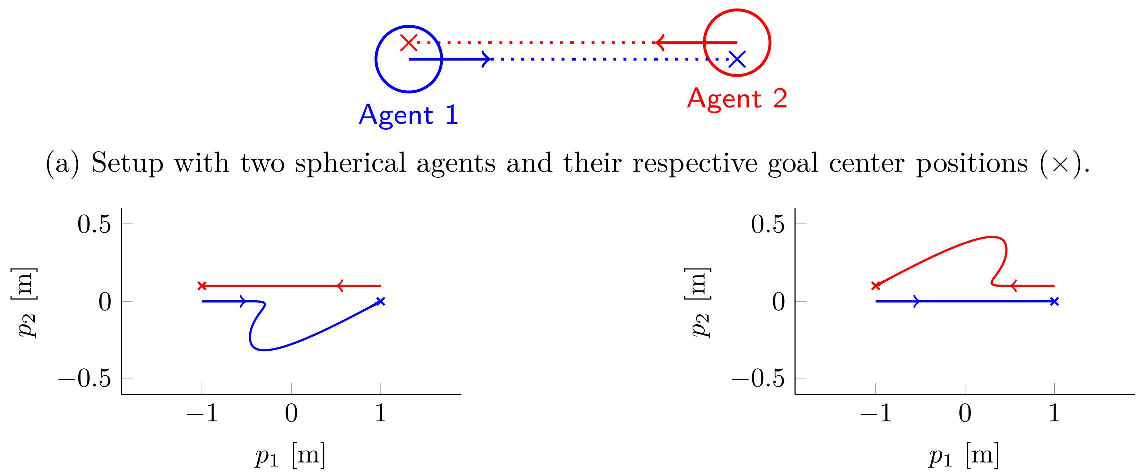
5.3.1 Agents in Different Priority Communities

Starting with a rather straightforward example, we examine the case of two spherical agents with radius $r_{\mathrm{ag}} \in \mathbb{R}^{+}$ and double-integrator dynamics in different priority communities. Initially, the agents are arranged as depicted in Fig. 5.7a. Both agents are nominally controlled by a PD control law to reach a goal position and the inter-agent constraint is given by

$$y_{\mathrm{c}} = 2r_{\mathrm{ag}} - \left\lVert \boldsymbol{p}_{\mathrm{ag1}} - \boldsymbol{p}_{\mathrm{ag2}} \right\rVert_2 , \tag{5.10}$$

where $\boldsymbol{p}_{\mathrm{ag1}}, \boldsymbol{p}_{\mathrm{ag2}} \in \mathbb{R}^2$ are the positions of the agents. The model parameters are provided in Table 5.1.

Without the constraint, both agents would follow a straight line towards their desired position. With the constraint and shared invariance control derived using Theorem 5.1, the agents adjust their actions to avoid constraint violations. Figure 5.7b depicts the trajectories of the agents if the second agent is in a higher priority community. As expected, the second agent follows a straight trajectory towards its desired position whereas the first agent carries



(a) Setup with two spherical agents and their respective goal center positions (×).

(b) Agent 2 with higher priority.

(c) Agent 1 with higher priority.

Figure 5.7.: Setup and trajectories for two invariance controlled agents in different priority communities.

Table 5.1.: Parameters for numerical evaluation of shared constraint satisfaction using invariance control.

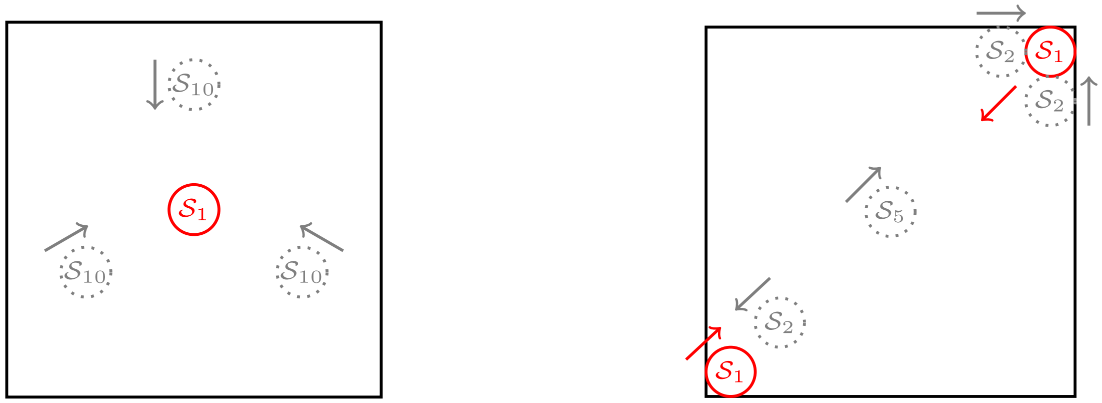
Sampling time	T_A	0.001 s
Agent radii	r_{ag}	0.2 m
Initial positions	$\boldsymbol{p}_{\text{ag},0}$	$[-1,\ 0]^\intercal$ m, $[1,\ 0.1]^\intercal$ m
Desired positions	$\boldsymbol{p}_{\text{ag,des}}$	$[1,\ 0]^\intercal$ m, $[-1,\ 0.1]^\intercal$ m
PD control	$\boldsymbol{K}_{\text{P}},\ \boldsymbol{K}_{\text{P}}$	$10 \cdot \boldsymbol{I}_2 \cdot 1/\text{s}^2$, $7 \cdot \boldsymbol{I}_2 \cdot 1/\text{s}$
Corrective control	γ	-20 m/s^2

out an evasive motion. If the priority communities are reversed, the first agent approaches the goal in a straight line while the second agent executes the evasive movement as shown in Fig. 5.7c. The successful assignment of evasive actions to agents in lower priority communities is the basis for resolving trap situations.

5.3.2 Evaluation of Trap Handling

For the evaluation of the proposed approach for trap detection and handling, we consider the two setups illustrated in Fig 5.8. Each setup contains multiple spherical agents with radius $r_{\text{ag}} \in \mathbb{R}^+$ and double-integrator dynamics. The agents are pair-wise constrained by inter-agent constraints according to (5.10). Nominal control is given by constant accelerations in the directions of the arrows shown in Fig. 5.8. Shared invariance control is implemented in accordance with Theorem 5.1. The parameters used in the simulations are provided in Table 5.2.

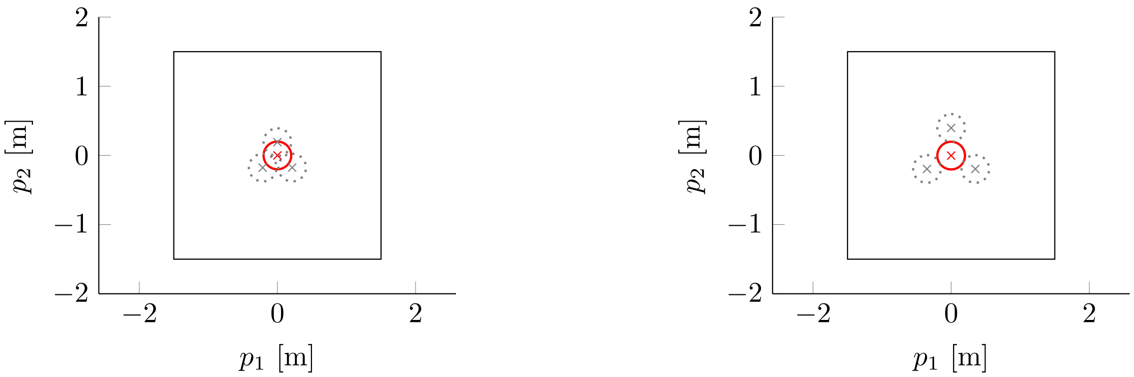
Figure 5.9a shows what happens in the setup of Fig. 5.8a without trap handling. With the inner agent being stationary and the outer agents moving towards it, at one point, the inner agent is trapped. As it is in the lowest priority community, it has to take over all evasive actions, which is not possible. Being in higher priority communities, the outer agents do not



(a) Single trapped agent.

(b) Multiple trapped agents.

Figure 5.8.: Initialization of different test scenarios with free ······ and trapped —— agents as well as static environmental constraints —— with the arrows indicating the movement direction of each agent.



(a) Violations without trap detection.

(b) Adherence with trap detection.

Figure 5.9.: Behavior of ······ free and —— trapped agents (a) without trap detection and (b) with trap detection in the setup of Fig. 5.8a.

respect the constraint and which leads to violations.

If, on the other hand, trap detection is used the agents form an emergency community and redistribute their priorities. As a result, the outer agents have to respect the constraints and violations are avoided as depicted in Fig. 5.9b.

The results of trap handling are further illustrated by the setup introduced in Fig. 5.8b. In this case, the agents move such that two emergency communities form around the agents in the opposing corners. Figure 5.10 depicts the final configuration of the agents, when all agents become stationary as movement in the desired directions is no longer possible. It may be observed that no constraint violations occur. Furthermore, the agent, which starts in the middle and moves towards the upper right corner pushes its way in between the two agents trapping the agent in the corner. This is owed to the fact that this agent is in a higher priority community, leading to evasive movements of the other two.

With reliable trap detection and handling, we now turn to the experimental evaluation of shared invariance control.

Table 5.2.: Simulation parameters for the evaluation of trap handling.

General		
Agent radii	r_{ag}	0.2 m
Threshold	Φ_{th}	0.05 m
Setup Fig. 5.8a		
Sampling time	T_A	0.001 s
Initial positions	$\boldsymbol{p}_{\text{ag},0}$	$[0,\ 0]^\intercal$, $[0,\ 1]^\intercal$, $[-\frac{\sqrt{3}}{2},\ -0.5]^\intercal$, $[\frac{\sqrt{3}}{2},\ -0.5]^\intercal$ m
Accelerations	$\boldsymbol{u}_{\text{no}}$	$[0,\ 0]^\intercal$, $[0,\ -2]^\intercal$, $[\sqrt{3},\ 1]^\intercal$, $[-\sqrt{3},\ 1]^\intercal$ m/s^2
Corrective control	γ	-20 m/s^2
Setup Fig. 5.8b		
Sampling time	T_A	0.0001 s
Initial positions	$\boldsymbol{p}_{\text{ag},0}$	$[1.3,\ 1.3]^\intercal$, $[1.3,\ 0.9]^\intercal$, $[0.9,\ 1.3]^\intercal$, $[-1.3,\ -1.3]^\intercal$, $[-0.9,\ -0.9]^\intercal$, $[0,\ 0]^\intercal$ m
Accelerations	$\boldsymbol{u}_{\text{no}}$	$[-1,\ -1]^\intercal$, $[0,\ 2]^\intercal$, $[2,\ 0]^\intercal$, $[2,\ 2]^\intercal$, $[-1,\ -1]^\intercal$, $[3,\ 3]^\intercal$ m/s^2
Corrective control	γ	$-10 \cdot \sqrt{18}$ m/s^2

5.3.3 Experimental Evaluation

The experimental evaluation is executed on the same robotic system [Sta06] as the experiments in Chapter 3. The two redundant position-controlled manipulators $j \in \{\mathrm{l}, \mathrm{r}\}$ are controlled to follow a desired trajectory while complying to external forces and satisfying inter-agent constraints for collision avoidance between the end effectors. The same control structure as shown in Fig. 3.11 is used.

Nominal Control

The trajectory for the position-controlled manipulators is generated by an admittance-type control law derived from a model of the robotic manipulators and an impedance control law. The gravity and Coriolis effects are assumed to be compensated leading to the general joint-dynamics model of the manipulators

$$\boldsymbol{M}_{j,\mathrm{q}} \ddot{\boldsymbol{q}}_{j,\mathrm{des}} = \boldsymbol{\tau}_j \ , \tag{5.11}$$

with the joint positions $\boldsymbol{q}_{j,\mathrm{des}} \in \mathbb{R}^7$, a positive definite mass matrix $\boldsymbol{M}_{j,\mathrm{q}} \in \mathbb{R}^{7\times 7}$ and the input torque $\boldsymbol{\tau}_j \in \mathbb{R}^7$. The Cartesian impedance control law as introduced by [AOF+03]

$$\boldsymbol{\tau}_{j,\mathrm{imp}} = \boldsymbol{\tau}_{j,\mathrm{ext}} + \boldsymbol{J}_j^\intercal \left(\boldsymbol{M}_{j,\boldsymbol{p}}\, \ddot{\boldsymbol{p}}_{j,\mathrm{ref}} + \boldsymbol{D}_{j,\boldsymbol{p}} \left(\dot{\boldsymbol{p}}_{j,\mathrm{ref}} - \dot{\boldsymbol{p}}_{j,\mathrm{des}} \right) + \boldsymbol{K}_{j,\boldsymbol{p}} \left(\boldsymbol{p}_{j,\mathrm{ref}} - \boldsymbol{p}_{j,\mathrm{des}} \right) \right) ,$$

with the external torques $\boldsymbol{\tau}_{j,\mathrm{ext}} = \boldsymbol{J}_j^\intercal \boldsymbol{f}_{j,\mathrm{ext}} \in \mathbb{R}^7$, derived from the Cartesian forces $\boldsymbol{f}_{j,\mathrm{ext}} \in \mathbb{R}^3$ using the Jacobian $\boldsymbol{J}_j \in \mathbb{R}^{3\times 7}$, the desired Cartesian position $\boldsymbol{p}_{j,\mathrm{des}} \in \mathbb{R}^3$, the Cartesian reference trajectory $\boldsymbol{p}_{j,\mathrm{ref}} \in \mathbb{R}^3$, the Cartesian mass matrix $\boldsymbol{M}_{j,\boldsymbol{p}} \in \mathbb{R}^{3\times 3}$ and positive definite stiffness $\boldsymbol{K}_{j,\boldsymbol{p}} \in \mathbb{R}^{3\times 3}$ and damping $\boldsymbol{D}_{j,\boldsymbol{p}} \in \mathbb{R}^{3\times 3}$ matrices for both manipulators $j \in \{\mathrm{l}, \mathrm{r}\}$. The Cartesian impedance is complemented by the desired null-space behavior

$$\boldsymbol{\tau}_{j,\mathrm{N}} = -\boldsymbol{K}_{j,\mathrm{N}} \left(\boldsymbol{q}_{j,\mathrm{des}} - \boldsymbol{q}_{j,\mathrm{N}} \right) - \boldsymbol{D}_{j,\mathrm{N}} \dot{\boldsymbol{q}}_{j,\mathrm{N}},$$

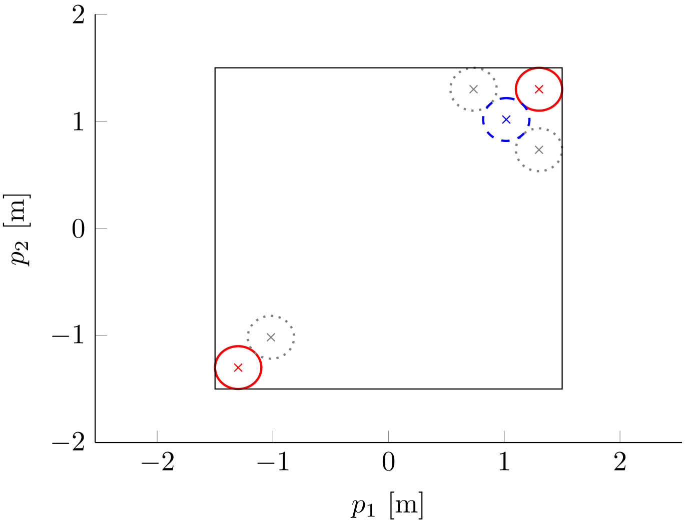


Figure 5.10.: Two emergency communities with ······ free and ——, - - - - trapped agents forming in opposing corners in the setup shown in Fig. 5.8b.

with the desired joint trajectory $\boldsymbol{q}_{j,\mathrm{N}}$ and the positive definite matrices $\boldsymbol{K}_{j,\mathrm{N}},\ \boldsymbol{D}_{j,\mathrm{N}} \in \mathbb{R}^7$. This leads to the nominal control law

$$\boldsymbol{\tau}_{j,\mathrm{no}} = \boldsymbol{\tau}_{j,\mathrm{imp}} + (\boldsymbol{I_7} - \boldsymbol{J}_j^\intercal (\boldsymbol{J}_j^\#)^\intercal)\boldsymbol{\tau}_{j,\mathrm{N}}$$

with the generalized inverse $\boldsymbol{J}_j^\# \in \mathbb{R}^{3\times 3}$, which achieves the desired force behavior in Cartesian task space. Invariance control is included in between nominal control and the dynamics model (5.11) as shown in Fig. 3.11 thus generating a constraint admissible trajectory for the actual robotic system.

Constraints

An inter-agent constraint is chosen to model both end effectors being enclosed by spheres with constant radii, which should never overlap. The constraint function is given by

$$y_\mathrm{c} = -\left\|\boldsymbol{p}_{\mathrm{l,des}} - \boldsymbol{p}_{\mathrm{r,des}}\right\|_2 + r_\mathrm{sph} \tag{5.12}$$

with a constant radius $r_\mathrm{sph} = r_\mathrm{l} + r_\mathrm{r} \in \mathbb{R}^+$, which is the sum of the radii of the spheres around both end effectors. Additionally, static box constraints similar to Sec. 3.4.1 are implemented. Naturally, more inter-agent constraints may be added for full-body collision avoidance, but are omitted for improved clarity of the results.

Corrective Control

Differentiation of (5.12)

$$\begin{aligned}
\dot{y}_\mathrm{c} &= \frac{\partial y_\mathrm{c}}{\partial \boldsymbol{p}_{\mathrm{l,des}}} \frac{\partial \boldsymbol{p}_{\mathrm{l,des}}}{\partial \boldsymbol{q}_{\mathrm{l,des}}} \dot{\boldsymbol{q}}_{\mathrm{l,des}} + \frac{\partial y_\mathrm{c}}{\partial \boldsymbol{p}_{\mathrm{r,des}}} \frac{\partial \boldsymbol{p}_{\mathrm{r,des}}}{\partial \boldsymbol{q}_{\mathrm{r,des}}} \dot{\boldsymbol{q}}_{\mathrm{r,des}} = \frac{\partial y_\mathrm{c}}{\partial \boldsymbol{p}_{\mathrm{l,des}}} \boldsymbol{J}_\mathrm{l} \dot{\boldsymbol{q}}_{\mathrm{l,des}} + \frac{\partial y_\mathrm{c}}{\partial \boldsymbol{p}_{\mathrm{r,des}}} \boldsymbol{J}_\mathrm{r} \dot{\boldsymbol{q}}_{\mathrm{r,des}}, \\
\ddot{y}_\mathrm{c} &= \frac{\partial \dot{y}_\mathrm{c}}{\partial \boldsymbol{p}_{\mathrm{l,des}}} \boldsymbol{J}_\mathrm{l} \dot{\boldsymbol{q}}_{\mathrm{l,des}} + \frac{\partial \dot{y}_\mathrm{c}}{\partial \boldsymbol{p}_{\mathrm{r,des}}} \boldsymbol{J}_\mathrm{r} \dot{\boldsymbol{q}}_{\mathrm{r,des}} + \frac{\partial y_\mathrm{c}}{\partial \boldsymbol{p}_{\mathrm{l,des}}} \left(\dot{\boldsymbol{J}}_\mathrm{l} \dot{\boldsymbol{q}}_{\mathrm{l,des}} + \boldsymbol{J}_\mathrm{l} \ddot{\boldsymbol{q}}_{\mathrm{l,des}}\right) \\
&\quad + \frac{\partial y_\mathrm{c}}{\partial \boldsymbol{p}_{\mathrm{r,des}}} \left(\dot{\boldsymbol{J}}_\mathrm{r} \dot{\boldsymbol{q}}_{\mathrm{r,des}} + \boldsymbol{J}_\mathrm{r} \ddot{\boldsymbol{q}}_{\mathrm{r,des}}\right) .
\end{aligned}$$

and substitution of $\ddot{\boldsymbol{q}}_{j,\mathrm{des}}$ from (5.11) yields the relative degree $r = 2$. The optimization condition is given by (5.4) with

$$\begin{aligned}\boldsymbol{\alpha}_j &= \frac{\partial y_\mathrm{c}}{\partial \boldsymbol{p}_{j,\mathrm{des}}} \boldsymbol{J}_j \boldsymbol{M}_{j,\mathrm{q}}^{-1}\,, \quad j \in \{\mathrm{l}, \mathrm{r}\} \\ b &= \sum_{j \in \{l,r\}} \frac{\partial \dot{y}_\mathrm{c}}{\partial \boldsymbol{p}_{j,\mathrm{des}}} \boldsymbol{J}_j \dot{\boldsymbol{q}}_{j,\mathrm{des}} + \frac{\partial y_\mathrm{c}}{\partial \boldsymbol{p}_{j,\mathrm{des}}} \dot{\boldsymbol{J}}_j \boldsymbol{q}_{j,\mathrm{des}}\,.\end{aligned}$$

Corrective control is derived according to Theorem 5.1 with the optimization conditions corresponding to the static box constraints from Sec. 3.4.1 included in the optimization of each agent according to Remark 5.2.

Priority Assignment

In the experiment, the agent priorities are assigned according to multiple criteria:

- Each agent has a static ***task priority*** c_tsk.
- The ***number of nearly active constraints*** according to Def. 5.7 yield the criterion

  $$c_\Phi = |\mathcal{D}|\,,\ \mathcal{D} = \{i \in \mathcal{B} | \Phi_i \geq \Phi_\mathrm{th}\}\,.\,,$$

  which assigns a higher priority to agents with a more constrained environment.
- The ***external force*** criterion assigns higher priorities to agents with high external forces.

  $$c_{\boldsymbol{f}_\mathrm{ext}} = \|\boldsymbol{f}_\mathrm{ext}\|_2$$

- The ***joint limit*** criterion

  $$c_\mathrm{jl} = \frac{1}{\min\left(\boldsymbol{q}_\mathrm{max} - \boldsymbol{q}, \boldsymbol{q} - \boldsymbol{q}_\mathrm{min}\right)}$$

  assigns higher priorities if agents are close to a joint limit.
- The ***manipulability measure*** as introduced by [Yos85] is based on the Jacobian $\boldsymbol{J}$ and is used to give a higher priority to agents with lower manipulability in task space.

  $$c_{\boldsymbol{J}} = \frac{1}{\sqrt{\det \boldsymbol{J}\boldsymbol{J}^\mathsf{T}}}.$$

A weighted sum with positive weights over the desired criteria derives the final agent priorities.

Implementation

The derived invariance control law including nominal control and the constraints is implemented in the *Real-Time Workshop* of *Matlab/Simulink*. The used solver is a discrete time Euler solver with the sampling frequency of 1 kHz. The Cartesian reference trajectories are given by

$$\boldsymbol{p}_{\mathrm{l,ref}} = \int_{t_0}^{t} 0.1 \begin{bmatrix} 0.2\sin\left(2\frac{\pi}{25}\psi + \frac{\pi}{2}\right) \\ -0.1\sin\left(\frac{\pi}{25}\psi\right) \\ -0.1\sin\left(\frac{\pi}{25}\psi + \frac{\pi}{2}\right) \end{bmatrix} \mathrm{d}\psi,$$
$$\boldsymbol{p}_{\mathrm{r,ref}} = \int_{t_0}^{t} 0.1 \begin{bmatrix} 0.2\sin\left(2\frac{\pi}{25}\psi + \frac{\pi}{2}\right) \\ 0.05\sin\left(\frac{\pi}{25}\psi\right) \\ -0.05\sin\left(\frac{\pi}{25}\psi + \frac{\pi}{2}\right) \end{bmatrix} \mathrm{d}\psi\ .$$

The parameters for the experiments are provided in Tab. 5.3. If not denoted differently, the parameters for both manipulators are equal.

Results

Shared constraint satisfaction is validated in two experiments. First, solely static priorities are used to illustrate the capabilities in the presence of environmental box constraints and external forces, which are measured by a *JR3 sensor* sensing forces and torques with 6 degrees of freedom. Then, solely the inter-agent constraint is considered and the static priorities are replaced by a weighted sum of the dynamic priority criteria with the weights ω_{Φ}, $\omega_{\boldsymbol{f}_{\mathrm{ext}}}$, ω_{jl}, $\omega_{\boldsymbol{J}} \in \mathbb{R}^{+}$ corresponding to the criteria as provided in Tab. 5.3. This second experiment evaluates the shared control approach for multi-agent systems with dynamic priority assignment.

Experiment with Static Priorities In this experiment, the end effectors follow the desired trajectories while satisfying the inter-agent and box constraints. Additionally, the external forces depicted in Fig. 5.11 are applied and generate a compliant reaction due to the used admittance control.

Despite the forces, the box constraints are met as shown in Fig. 5.12. The inter-agent constraint is satisfied as depicted by Fig. 5.13a which shows that the distance between the end effector positions is at least $r_{\mathrm{sph}} = 0.2\,\mathrm{m}$. In addition, the maxima of the invariance functions corresponding to the box constraints depicted in Fig. 5.13b remain at non-positive values. The relative position deviation from the trajectory without invariance control

$$\boldsymbol{e}_{\mathrm{rel},i} = \frac{\boldsymbol{e}_{\boldsymbol{p},i}}{\boldsymbol{e}_{\boldsymbol{p},\mathrm{l}} + \boldsymbol{e}_{\boldsymbol{p},\mathrm{r}}},\ \text{with } i \in \{\mathrm{l},\ \mathrm{r}\} \tag{5.13}$$

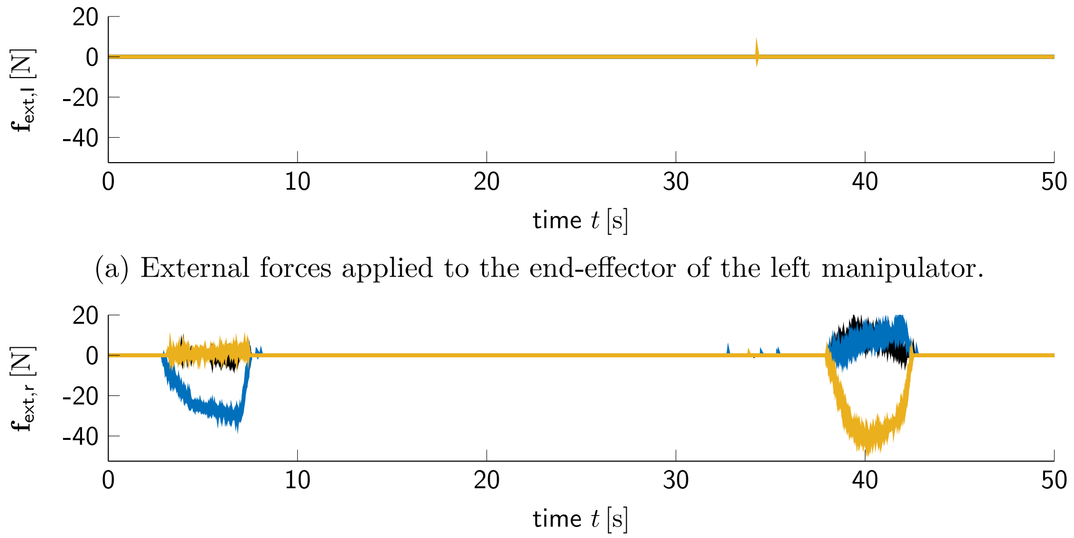
with $\boldsymbol{e}_{\boldsymbol{p},i} = \boldsymbol{p}_{\mathrm{ref},i} - \boldsymbol{p}_i$ is shown in Fig. 5.13c. Note that the high oscillations when no constraint is active are due to the small position deviations in this case and therefore tiny changes have a large impact on the relative position deviation. More importantly, however, if solely the inter-agent constraint is active, e.g. at $t \in [129, 132]\,\mathrm{s}$, the relative position deviation is partitioned according to the weights $w_{\mathrm{l}} = \frac{1}{3}$ and $w_{\mathrm{r}} = \frac{2}{3}$ resulting from the static priorities and (5.6). Hence, the experimental results validate shared invariance control for

Table 5.3.: Parameters for the experiments to validate shared constraint satisfaction.

General		
Sampling time	T_A	0.001 s
Initial joint position	$\boldsymbol{q}_{0,\mathrm{l}}$	$[0.27, 2.22, 0, -1.49, 1.44, 1.32, -0.73]^\intercal$ rad
	$\boldsymbol{q}_{0,\mathrm{r}}$	$[0.61, 0.39, 0, 1.5, 0.79, 0.93, -1.35]^\intercal$ rad
Initial Cartesian position	$\boldsymbol{p}_{\mathrm{l},0}$	$[0.5, 0.15, -0.441]^\intercal$ m
	$\boldsymbol{p}_{\mathrm{r},0}$	$\begin{bmatrix}0.5 & -0.15 & -0.341\end{bmatrix}^\intercal$ m
Nominal control		
Desired compliance behavior	$\boldsymbol{M_p}$	$12 \cdot \boldsymbol{I}_3 \cdot \mathrm{kg}$
	$\boldsymbol{K_p}$	$200 \cdot \boldsymbol{I}_3 \cdot \mathrm{N/m}$
	$\boldsymbol{D_p}$	$80 \cdot \boldsymbol{I}_3 \cdot \mathrm{N\,s/m}$
Desired null-space behavior	$\boldsymbol{K}_\mathrm{N}$	$\mathrm{diag}\,(360 \cdot \boldsymbol{I}_4,\ 180 \cdot \boldsymbol{I}_3)$ N m/rad
	$\boldsymbol{D}_\mathrm{N}$	$\mathrm{diag}\,(25 \cdot \boldsymbol{I}_4,\ 10 \cdot \boldsymbol{I}_3)$ N m s/rad
Constraints		
Box constraints	$\boldsymbol{p}_\mathrm{min}$	$[0.3, -0.3, -0.48]$ m
	$\boldsymbol{p}_\mathrm{max}$	$[0.57, 0.3, -0.28]$ m
Inter-agent constraint	r_sph	0.2 m
Shared invariance control		
Control parameter	γ	-1.8 m/s^2
Static priorities	$c_{\mathrm{tsk,l}}$, $c_{\mathrm{tsk,r}}$	2, 1
Dynamic priority assignment	Φ_th	$-0.025\,\mathrm{m}$
	ω_Φ	$0.5\,\mathrm{m}$
	$\omega_{\boldsymbol{f}_\mathrm{ext}}$	0.1 1/N
	ω_jl	0.05 rad
	$\omega_{\boldsymbol{J}}$	$0.4\,\mathrm{m}$

multi-agent systems as both manipulators satisfy the static box constraints and share the evasive effort for the inter-agent constraint according to their priority values.

Experiment with Dynamic Priorities In the second experiment the priorities for shared invariance control are determined using the dynamic criteria. In this experiment solely the inter-agent constraint is considered. As the applied external forces, depicted in Fig. 5.14, are used in the force-based priority assignment criterion $c_{\boldsymbol{f}_\mathrm{ext}}$ to determine the agent priorities, they have a direct impact on the priorities, shown in Fig. 5.15b. Instead of being static, the priority values of the agents now change over time according to the dynamic criteria. Nevertheless, the manipulators generally satisfy the inter-agent constraint keeping at least a distance of $r_\mathrm{sph} = 0.2\,\mathrm{m}$ as depicted in Fig. 5.15a. The minor violations are caused by the sampled time implementation, a small value of γ and the system not being able to counter-



(a) External forces applied to the end-effector of the left manipulator.

(b) External forces applied to the end-effector of the right manipulator.

Figure 5.11.: External forces — $\mathrm{f}_{p1,\text{ext}}$, — $\mathrm{f}_{p2,\text{ext}}$, — $\mathrm{f}_{p3,\text{ext}}$ in the experiment with static priorities for the left and right manipulator.

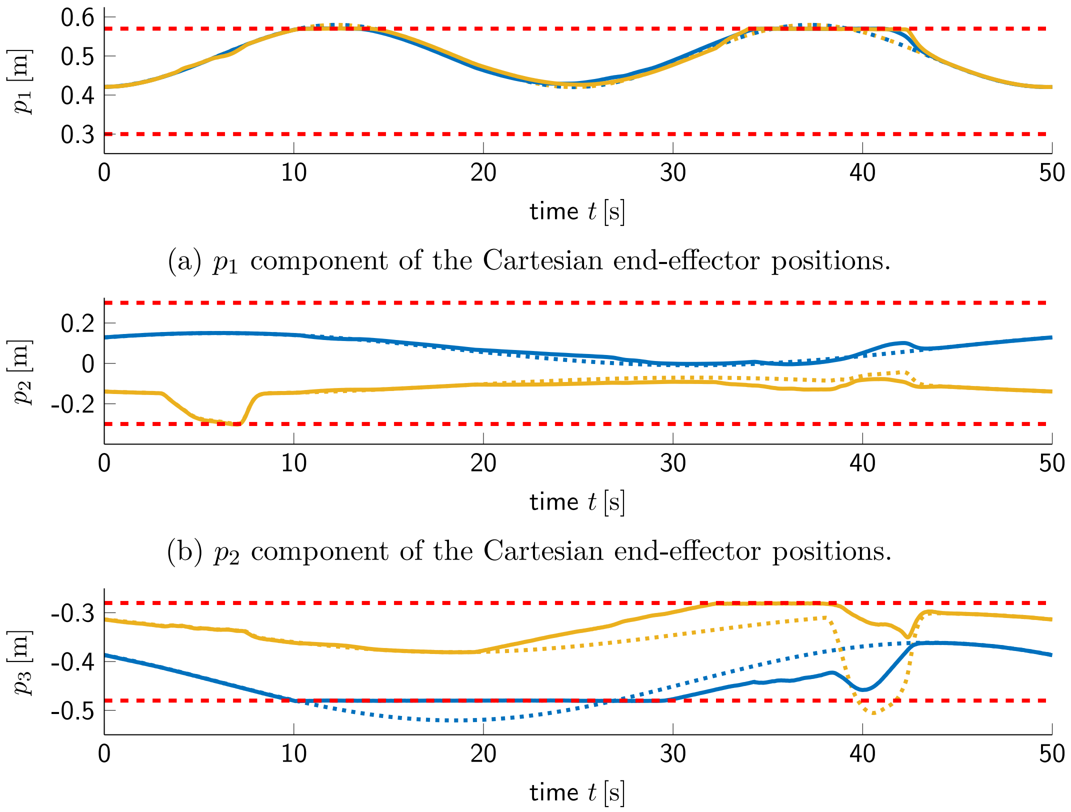
act arbitrarily large external forces. By increasing γ and applying methods for chattering reduction from Sec. 3.2.2, this effect may be overcome.

Since the force sensors are prone to measurement noise, the priorities of the agents as depicted in Fig. 5.15b and the desired partition in Fig. 5.15c are subject to noise as well. It may, however be observed that the relative position deviation of the manipulators follows the trend of the desired weights while the constraint is active, which results in the manipulator with the higher priority experiencing the smaller position deviation. Furthermore, the robotic system acts like a low-pass filter, smoothing the desired signal. Naturally, if the constraint is inactive, the weights differ from the desired weight as no evasive action and therefore no weighting is necessary.

5.4 Discussion

Concluding this chapter, we summarize the capabilities of the proposed control approach for shared constraint satisfaction and the associated prioritization scheme. Sharing the effort necessary to ensure constraint satisfaction in multi-agent systems allows for agents with important tasks to execute them with priority while other agents take the major part in achieving constraint satisfaction. The control input is generated by a minimization problem and may be designed using invariance control as well as CBF-based control, such that the approach which is more advantageous for the respective application may be chosen. As the approach does not change the general idea of I/O-linearization-based control approaches, which generate a condition on the input for each constraint, shared constraint satisfaction is compatible with the approaches introduced in the previous chapters.

The presented prioritization scheme is comprised of two stages. The first stage assigns the agents to priority communities to ensure that the optimization for the control input remains feasible for all agents. The second stage then distributes the priorities according to



(a) p_1 component of the Cartesian end-effector positions.

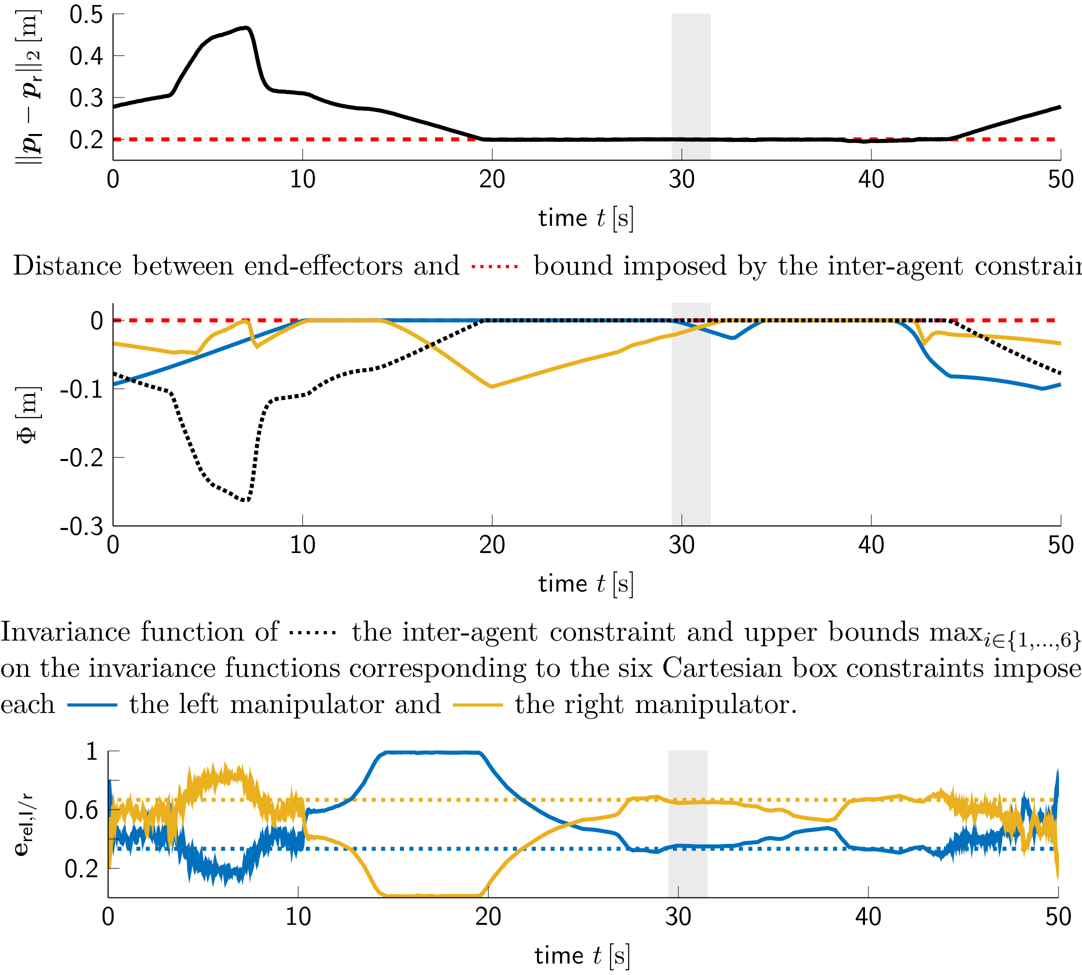
(b) p_2 component of the Cartesian end-effector positions.

(c) p_3 component of the Cartesian end-effector positions.

Figure 5.12.: Cartesian position $\boldsymbol{p}_{\mathrm{l/r}}$ of both end effectors in the experiment with static priorities and - - - - box constraints for the —— left and the —— right manipulator with the respective reference $\boldsymbol{p}_{\mathrm{ref,l/r}}$ ⋯⋯, ⋯⋯ without invariance control.

a variety of criteria, thus enabling the agents to share evasive actions required for constraint satisfaction.

The combination of control and prioritization scheme is validated in numerical examples and experiments, which encourage the use of shared constraint satisfaction in setups with multiple agents including physical human-robot interaction with inter-agent constraints as well as environmental constraints and dynamic priorities.

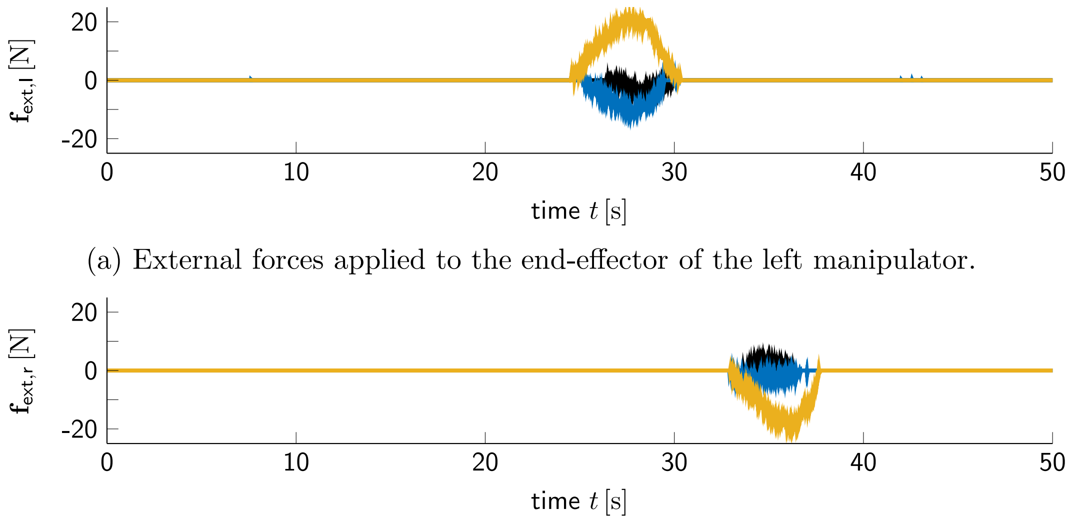


(a) Distance between end-effectors and ······ bound imposed by the inter-agent constraint.

(b) Invariance function of ······ the inter-agent constraint and upper bounds $\max_{i\in\{1,\ldots,6\}}(\Phi_i)$ on the invariance functions corresponding to the six Cartesian box constraints imposed on each —— the left manipulator and —— the right manipulator.

(c) Relative position deviation for —— the left and —— the right manipulator with the respective desired values ······ , ······ .

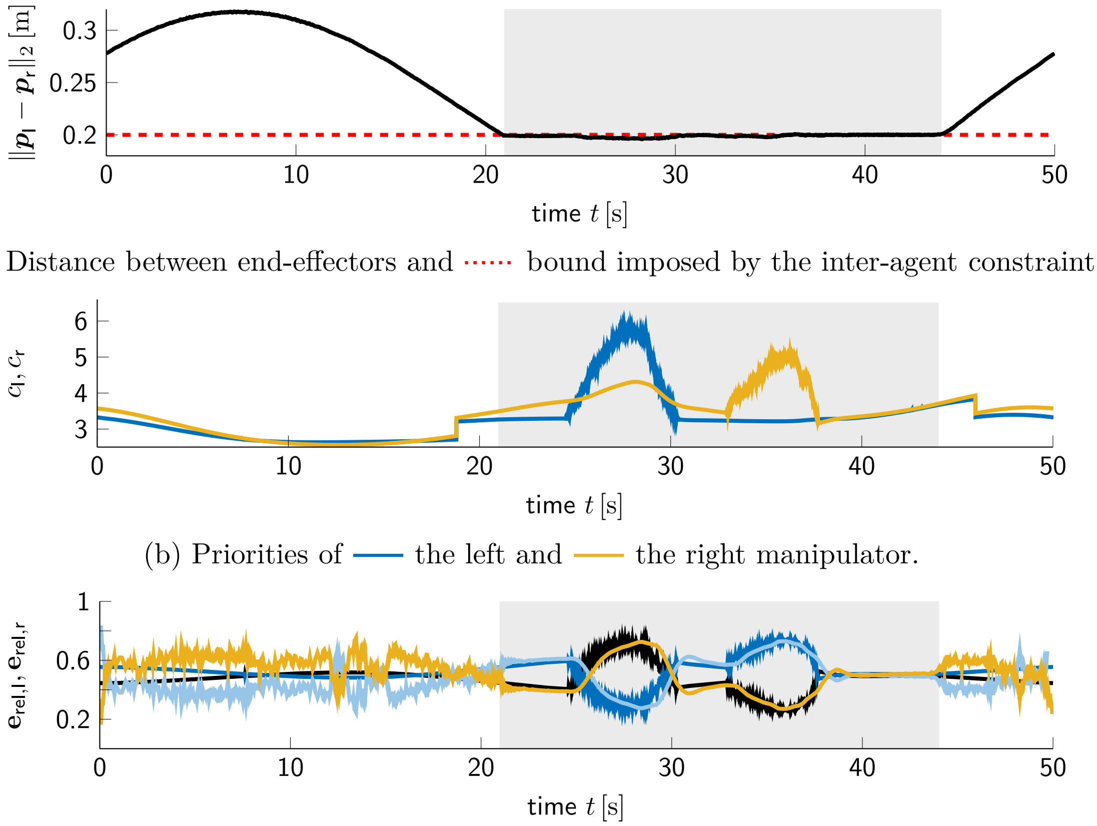
Figure 5.13.: Inter-agent constraint, invariance functions and relative position deviation for the experiment with static priorities emphasizing ▨ the time during which solely the inter-agent constraint is active.



(a) External forces applied to the end-effector of the left manipulator.

(b) External forces applied to the end-effector of the right manipulator.

Figure 5.14.: External forces — $\mathrm{f}_{p1,\mathrm{ext}}$, — $\mathrm{f}_{p2,\mathrm{ext}}$, — $\mathrm{f}_{p3,\mathrm{ext}}$ in the experiment with dynamic priorities for the left and right manipulator.



(a) Distance between end-effectors and ⋯⋯ bound imposed by the inter-agent constraint.

(b) Priorities of — the left and — the right manipulator.

(c) Relative position deviation to the unconstrained position for — the left and — the right manipulator with the respective desired values —, —.

Figure 5.15.: Inter-agent constraint, dynamic priorities and weights for shared invariance control with dynamic priorities with emphasis on ▨ the time during which the inter-agent constraint is active.

6

Conclusion and Future Directions

Robotic systems sharing their workspace with humans need the ability to interact and cooperate closely as well as physically with human partners. Examples of such scenarios are found in industrial manufacturing, health care applications such as robot-aided surgery or rehabilitation, elderly care and household assistants. In all these applications, the safe interaction is an essential and indispensable factor. Safety is characterized by physical and psychological factors, which means that interacting agents, be it humans or robots, should not be harmed and any interaction should be comfortable and stress-free. Hence, any control framework implemented on a robot engaging in human-robot interaction should be dependable and guarantee safety under real-time conditions.

This thesis aims at developing a control framework that includes the important features required for a safe human-robot interaction.

Summary of the Contributions

We present invariance control and CBF-based control as promising candidates for safe human-robot interaction. Both are added to existing control loops with task-oriented nominal control. The control input is derived by solving a convex minimization, which is subject to conditions on the input. By designing the control input to render a subset of the state space positively invariant, adherence to static and dynamic state and output constraints is guaranteed, resolving the need for the integrity of the human partner. The need for comfortable and stress-free interaction is addressed by augmenting the systems, which allows to adjust the dynamic behavior of the system to fit the requirements.

By analyzing the effect of uncertain parameters on the optimization conditions, robust, probabilistic and scenario-based constraint satisfaction are derived, thus allowing for safe interaction even in cases when the dynamics of the constraints are not exactly known, either due to uncertain measurements or due to the model being generated by a learning-based approach. The choice of the satisfaction type depends on the desired outcome as for example robust constraint satisfaction guarantees satisfaction for all possible uncertainties, whereas probabilistic and scenario-based satisfaction yield probabilistic guarantees.

In addition, both approaches allow for shared satisfaction of constraints in multi-agent systems. Using priorities derived from a dual-stage prioritization scheme, weights are calculated and included in the optimization conditions for constraint satisfaction. The prioritization ensures that the optimization remains feasible even if multiple agents are closely interacting and the included weights are responsible for sharing control actions between the agents.

Note that both invariance control and CBF-based control generate optimization conditions with a similar structure. The same holds true for the conditions resulting from uncertain constraints and the weighted conditions for shared control effort. Hence the approaches and the different constraint types are compatible, which means they may be combined arbitrarily allowing to find the best fit to the respective application and constraints.

The presented experimental results encourage the application of invariance control to robotic systems in scenarios involving humans and physical interaction with humans.

Implications

Revisiting the challenges imposed by physical human-robot interaction as introduced in Sec. 1.1, we examine the extent to which the proposed control framework is able to answer these questions.

Challenge 1 is addressed by augmenting the system and applying augmented invariance control. Even though the approach is introduced for invariance control, the concepts may be transferred to CBF-based control. There are little restrictions on the choice of augmentation, which means that it may be used to adjust the resulting dynamics to the expectation of the human partner. Furthermore, as nominal control may by chosen freely, a human-friendly design [ZRK+04] may be utilized.

Challenge 2 is met by invariance control as well as CBF-based control. Both approaches achieve constraint satisfaction in the presence of interaction forces as demonstrated in the experiments. By applying the presented method for chattering reduction, the switching invariance control law also guarantees constraint satisfaction in sampled time, which is achieved naturally by CBF-based control. Furthermore, the control input is derived from convex optimization using efficient existing solvers, which are able to find a solution in real-time even for high numbers of constraints.

Challenge 3 is addressed by minimizing the deviation of corrective control from nominal control. This means that the resulting behavior is close to the desired unconstrained behavior even in the presence of constraints, which is supported by the constraints having little to no influence on the control input if the state is far from the bounds. It should, however, be noted, that the control input being close to nominal control does not necessarily mean that the measured constrained manipulator trajectory is close to the unconstrained trajectory as neither of the proposed control schemes minimizes the deviation from the desired trajectory. Minimization in the input space in contrast has the advantage of preserving the haptic sensation generated by nominal control.

Challenge 4 is concerned with uncertain constraints, an issue that is addressed in Chapter 4. The presented approaches allow the guaranteed satisfaction of constraints subject to bounded uncertainties and a probabilistic guarantee for known uncertainty distributions or sufficiently many data samples. Therefore, if some knowledge about the uncertainty characteristics is available, both invariance control and CBF-based control are able to handle uncertain constraints.

Challenge 5 is addressed by introducing a prioritization of the agents depending on their task and the feasibility of the optimization problem for control derivation. Based on the priorities, each agent takes over most, little or none of the evasive effort required for achieving safe interaction. The approach avoids agents blocking the path of other agents and allows for a redistribution of the evasive effort if the priorities of the agents change, for example as their task becomes more urgent.

Outlook

In this thesis we take a step towards safe physical and close human-robot interaction between humans and robots in the presence of uncertain constraints. Novel control frameworks are presented and supported by formal results. Illustrating numerical examples and experiments on anthropomorphic manipulators verify the theoretical results. Although a number of relevant issues regarding safe human-robot interaction have been resolved in this thesis, several open research problems remain to be studied in future works.

Input constraints

Although the presented control approaches satisfy constraints by generating conditions on the input, both approaches require the available set of input values to be unbounded. As the conditions artificially constrain the input, additional limits on the available input values may be included straightforwardly in the optimization but may lead to infeasibility. Changing the control parameters attenuates that issue but does not change the fact that high input values may be required occasionally to preserve the satisfaction guarantees. As the inclusion of bounds on the input corresponds to a saturation in the system model, one idea might be to assess the feasibility of the approaches with a different class of nonlinear systems.

Uncertain system model

A shortcoming of using I/O-linearization in the control derivation is that an exact model is required for perfect results. If unmodeled effects are present, the satisfaction guarantees are lost. This thesis already addresses the effects of uncertain parameters in the constraint description but assumes the system model to be known. For even wider applicability it will be useful to extend the presented concepts to systems with uncertain dynamics, i.e. due to an unknown load or due to learned dynamics [BUH17].

Learning constraints

In addition to learning the system dynamics or the constraint parameters, the possibility to detect and model constraints during runtime further improves the safety of the interaction. If the system is capable to automatically generate constraints, the necessity of fully modeling the environment becomes obsolete and hence unsafe behavior due to unmodeled events may be avoided. Ideally the system should be able to find an optimal configuration of constraints during runtime that ensures safety while restricting the robot as little as possible. In such applications, however, any related uncertainties have to be included.

Sharing control with human partners

Last but not least, it might be interesting to extend the concepts for shared control in multi-robot systems to teams of robots and humans. Instead of assuming that the robot has to take over the entire effort for keeping harm from the humans, it might be promising to use insights into human-human interaction [FGP+09] to predict how much of the evasive effort will be performed by the human. This would result in less conservative behavior of the robot, which may be advantageous in some applications.

A

Formal Proofs

Proofs from Section 3.1

Proof of Lemma 3.1. For the proof, we define the sets

$$\mathcal{H}_i(\boldsymbol{\eta}) = \{\boldsymbol{x} \,|\, h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta}) \leq 0\}$$
$$\mathcal{G}_i(\boldsymbol{x}_\eta, \gamma_i) = \{\boldsymbol{x} \,|\, \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) \leq 0\} \ .$$

With (2.12), the admissible and invariant set are given by

$$\mathcal{H}(\boldsymbol{\eta}) = \bigcap_{i=1}^{l} \mathcal{H}_i(\boldsymbol{\eta}) \ , \qquad \mathcal{G}(\boldsymbol{x}_\eta, \boldsymbol{\gamma}) = \bigcap_{i=1}^{l} \mathcal{G}_i(\boldsymbol{x}_\eta, \gamma_i) \ .$$

In addition,

$$\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) \geq h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta})$$

holds since the invariance function is defined as the future maximum value of the constraint function. Hence,

$$\begin{aligned} \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) \leq 0 \quad &\Rightarrow \quad h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta}) \leq 0 \\ h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{\eta}) \leq 0 \quad &\not\Rightarrow \quad \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) \leq 0 \end{aligned}$$

holds, which is equivalent to

$$\mathcal{G}_i(\boldsymbol{x}_\eta, \gamma_i) \subseteq \mathcal{H}_i(\boldsymbol{\eta})$$

and as a result

$$\mathcal{G}(\boldsymbol{x}_\eta, \boldsymbol{\gamma}) = \bigcap_{i=1}^{l} \mathcal{G}_i(\boldsymbol{x}_\eta, \gamma_i) \subseteq \bigcap_{i=1}^{l} \mathcal{H}_i(\boldsymbol{\eta}) = \mathcal{H}(\boldsymbol{\eta}) \ .$$

□

Proof of Lemma 3.2. Invariance is achieved if the invariance function never takes a positive value. There is only a danger of leaving the invariant set, if the state is at the bound, i.e. if $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) = 0$ holds. Therefore, in order to show invariance, it suffices to show that the invariance function never increases at this point.

The invariance function $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i)$ represents the maximum future value of the constraint function for a pseudo input $z_i = \gamma_i$. Hence, if $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) = c$ with any constant $c \in \mathbb{R}$ holds at $t = t_0$, setting $z_i = \gamma_i$ results in $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) \leq c$ for all $t \geq t_0$ by design.

Applying these considerations to the cases in (3.16) shows that for $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) = 0$ there are two options for the pseudo input. It is either equal to γ_i, which leads to $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) \leq 0$ for all future times, or it is equal to 0. Since it is equal to zero only if additionally $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, 0) \leq 0$ holds, extending the previous considerations,

$$\Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, \gamma_i) \leq \Phi_i(\boldsymbol{x}, \boldsymbol{x}_\eta, 0) \leq 0$$

holds for all future times, thus leading to invariance. □

Proof of Lemma 3.3. The cost function is assumed to be strictly convex. The constraints are linear in the optimization variable $\boldsymbol{u}$ and therefore affine and convex in $\boldsymbol{u}$. As there are no equality constraints, the optimization is strictly convex and any local minimum is a global minimum [Boy04, p.136ff.]. □

Proof of Theorem 3.1. Corrective control is determined by the constrained minimization problem (3.31). The constraint, which has to be fulfilled by the corrective input $\boldsymbol{u}_\mathrm{c}$, is given by (3.30). Using (3.1) shows that the left side of the condition is the pseudo input of the active constraints.

$$\boldsymbol{z}_i = \boldsymbol{a}_{\mathrm{c},i}^\mathsf{T} \boldsymbol{u}_\mathrm{c} + b_{\mathrm{c},i} \tag{A.1}$$

Therefore, any solution of the minimization fulfills

$$\boldsymbol{z}_i \preccurlyeq \boldsymbol{z}_{\mathrm{c},i} \ . \tag{A.2}$$

An input $z_i \leq z_{\mathrm{c},i}$ for all active constraints renders an integrator chain invariant with respect to the invariant set by Lemma 3.2. Additionally, the remaining inactive constraints are not in danger of violating a constraint and stay within the invariant set for any control action. Therefore, the solution of the constrained minimization problem provides a corrective control input, which achieves positive invariance of the invariant set. □

Proof of Theorem 3.2. The system is outside of the invariant set, if at least one invariance function has a value larger than zero and it enters the invariant set, if the control signal is such that the invariance functions eventually take non-positive values. Therefore, the system is guaranteed to enter the invariant set, if the control signal is such that any state trajectory starting outside of the invariant set is guaranteed to enter the set eventually.

For constraints with relative degree $r_i = 1$, the invariance function is given by (3.12). As the constraint is active, i.e. initially $y_{i,0} = y_i(t_0) > 0$, the input z_i of the linearized integrator chain fulfills $z_i = z_{\mathrm{c},i} \leq \gamma_i < 0$, see (3.21), (3.31). As $\dot{y}_i = z_i$ holds for $r_i = 1$, $\dot{y}_i \leq \gamma_i < 0$ holds as long as the constraint is active, i.e. as long as y_i has a positive value. This means, that y_i and therefore also the value of the invariance function decreases until $y_i = 0$ is reached, at which point the system enters the invariant set of the constraint. Since $\dot{y}_i$ is strictly less than zero, this will happen in a finite time interval $T \leq \frac{y_{i,0}}{\gamma_i}$.

For constraints with relative degree $r_i = 2$, the integrator chain of the linearized system has two states y_i and $\dot{y}_i$. The invariance function is given by (3.13). The areas of the linearized state space, for which the invariance function has a positive value, are given by

$$\begin{aligned}
\mathcal{S}_1(\boldsymbol{x}_\eta, \gamma_i) &= \{\boldsymbol{x} \mid y_i \leq 0 \wedge \dot{y}_i > 0 \wedge y_i > \frac{1}{2\gamma_i} \dot{y}_i^2\} \\
\mathcal{S}_2(\boldsymbol{x}_\eta) &= \{\boldsymbol{x} \mid y_i > 0 \wedge \dot{y}_i \geq 0\} \\
\mathcal{S}_3(\boldsymbol{x}_\eta) &= \{\boldsymbol{x} \mid y_i > 0 \wedge \dot{y}_i < 0\}
\end{aligned}$$

We now show that, starting from any arbitrary point in sets $\mathcal{S}_1$, $\mathcal{S}_2$ or $\mathcal{S}_3$, the system will eventually enter the invariant set.

First, assume that the initial configuration lies within set $\mathcal{S}_1$. In this case, the constraint is active and $\ddot{y}_i = z_i \leq z_{\mathrm{c},i} = \gamma_i < 0$ holds, see (3.22), (3.31). Since $\ddot{y}_i$ is strictly less than zero, $\dot{y}_i$ decreases and reaches zero after a finite time interval T_1. During that time, the output y_i increases its value since $\dot{y}_i > 0$ holds and may even become positive. If, after the time interval T_1, $y_i \leq 0$ holds, the system is within the invariant set, since $\dot{y}_i = 0$ holds. Otherwise, $y_i = 0$ and $\dot{y}_i > 0$ hold after a time interval $\Delta t < T_1$, meaning that the system enters the set $\mathcal{S}_2$.

Now assume that the initial configuration lies within $\mathcal{S}_2$. Again, the constraint is active and in this case, $\ddot{y}_i = z_i \leq z_{\mathrm{c},i} = \gamma_i < 0$ holds, see (3.22), (3.31). Similar to the previous case, $\dot{y}_i$ decreases and will reach a value of zero after a finite time interval T_2. During that time, the output y_i further increases its value since $\dot{y}_i > 0$ holds. After the time interval T_2, $y_i > 0$ and $\dot{y}_i = 0$ holds, while the input is still strictly negative, meaning that the system enters set $\mathcal{S}_3$.

Finally, assume that the initial configuration lies within $\mathcal{S}_3$. The constraint is active and in this case, $\ddot{y}_i = z_i \leq z_{\mathrm{c},i} = 0$, see (3.22), (3.31). The value of y_i decreases, since $\dot{y}_i < 0$ within set $\mathcal{S}_3$. With $\ddot{y}_i \leq 0$, $\dot{y}_i$ either remains constant at its strictly negative value or decreases further. Therefore, after a finite time interval T_3, $y_i = 0$ is reached, while $\dot{y}_i < 0$ holds, meaning that the system enters the invariant set.

Therefore, for constraints with $r_i = 2$, starting from arbitrary states outside of the invariant set, the state trajectory evolves such that the invariant set is entered within a finite time interval $T \leq T_1 + T_2 + T_3$. The trajectory may either evolve from $\mathcal{S}_1$ or $\mathcal{S}_3$ directly into the invariant set, from $\mathcal{S}_1$ over $\mathcal{S}_2$ and $\mathcal{S}_3$ or from $\mathcal{S}_2$ over $\mathcal{S}_3$ into the invariant set.

For constraints with higher relative degrees, the consideration is similar. Applying the pseudo input until all states of the integrator chain are negative, results in entering the set. The duration may be calculated using (3.6)–(3.8) and is finite since all signals are bounded. □

Proof of Theorem 3.3. This proof is conducted in two steps: First the existence of parameters $V_{\max}$ and α is shown, for which the optimization has a unique solution and second the resulting boundedness of the tracking error is proved.

In order to show that the minimization has a solution, it suffices to determine one input $\boldsymbol{u}_\mathrm{c}$ which fulfills both types of conditions: the invariance conditions derived from I/O-linearization, which ensure constraint satisfaction, and the convergence condition, which guarantees boundedness. By Assumption 3.1, there exists a solution to (3.31), i.e. the minimization problem with solely the invariance conditions. This solution is given by (3.38). The convergence condition on the other hand defines an upper bound on

$$\dot{V}(\boldsymbol{e}, \dot{\boldsymbol{e}}) = \frac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}} \underbrace{(\dot{\boldsymbol{x}}_\mathrm{des} - \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}_\mathrm{c})}_{=\dot{\boldsymbol{e}}} .$$

For $\mathcal{I} = \emptyset$ with $\mathcal{I}$ from (3.37), the invariance conditions have no influence on the solution and is fulfilled by $\boldsymbol{u}_\mathrm{c} = \boldsymbol{u}_\mathrm{no}$. In addition, $\boldsymbol{u}_\mathrm{c} = \boldsymbol{u}_\mathrm{no}$ also fulfills the convergence condition since then

$$\dot{V}(\boldsymbol{e}, \dot{\boldsymbol{e}}) = \dot{V}_\mathrm{no}(\boldsymbol{e}, \dot{\boldsymbol{e}}_\mathrm{no}) \leq B_{\dot{V}}$$

with the bound $B_{\dot{V}}$ as defined in Theorem 3.3 holds for any value of $V_{\max}$ and α.

For $\mathcal{I} \neq \emptyset$,

$$\boldsymbol{u}_{\mathrm{c},\mathcal{I}} = \boldsymbol{W}(\boldsymbol{A}_{\mathrm{c},\mathcal{I}}\boldsymbol{W})^{+}(\boldsymbol{z}_{\mathrm{c},\mathcal{I}} - \boldsymbol{b}_{\mathrm{c},\mathcal{I}}) + (\boldsymbol{I} - \boldsymbol{W}(\boldsymbol{A}_{\mathrm{c},\mathcal{I}}\boldsymbol{W})^{+}\boldsymbol{A}_{\mathrm{c},\mathcal{I}})\boldsymbol{u}_{\mathrm{no}}$$

from (3.38) solves the minimization (3.31) with only the invariance conditions, i.e. in matrix-vector notation

$$\boldsymbol{A}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}\boldsymbol{u}_{\mathrm{c},\mathcal{I}} + \boldsymbol{b}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}} \preccurlyeq \boldsymbol{z}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}$$

holds. If by adding the convergence condition,

$$\operatorname{rank}\begin{bmatrix}\boldsymbol{A}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}\\ \dfrac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}}\boldsymbol{G}(\boldsymbol{x})\end{bmatrix} > \operatorname{rank}(\boldsymbol{A}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}})$$

is fulfilled, then

$$\exists\ \boldsymbol{R}_{\mathcal{B}_{\mathrm{act}}}:\ \boldsymbol{R}_{\mathcal{B}_{\mathrm{act}}} \in \ker(\boldsymbol{A}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}),\ \boldsymbol{R}_{\mathcal{B}_{\mathrm{act}}} \notin \ker\left(\frac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}}\boldsymbol{G}(\boldsymbol{x})\right)\ .$$

The input $\boldsymbol{u}_{\mathrm{c}} = \boldsymbol{u}_{\mathrm{c},\mathcal{I}} + \boldsymbol{R}_{\mathcal{B}_{\mathrm{act}}}\boldsymbol{u}_{\mathcal{B}_{\mathrm{act}},\mathrm{ker}}$ fulfills the invariance conditions independently from the value of $\boldsymbol{u}_{\mathcal{B}_{\mathrm{act}},\mathrm{ker}}$ since

$$\begin{aligned}\boldsymbol{A}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}\boldsymbol{u}_{\mathrm{c}} + \boldsymbol{b}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}} &= \boldsymbol{A}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}(\boldsymbol{u}_{\mathrm{c},\mathcal{I}} + \boldsymbol{R}_{\mathcal{B}_{\mathrm{act}}}\boldsymbol{u}_{\mathcal{B}_{\mathrm{act}},\mathrm{ker}}) + \boldsymbol{b}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}\\ &= \boldsymbol{A}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}\boldsymbol{u}_{\mathrm{c},\mathcal{I}} + \underbrace{\boldsymbol{A}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}\boldsymbol{R}_{\mathcal{B}_{\mathrm{act}}}}_{=\boldsymbol{0}_{|\mathcal{B}_{\mathrm{act}}|\times m}}\boldsymbol{u}_{\mathcal{B}_{\mathrm{act}},\mathrm{ker}} + \boldsymbol{b}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}} = \boldsymbol{A}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}\boldsymbol{u}_{\mathrm{c},\mathcal{I}} + \boldsymbol{b}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}\ .\end{aligned}$$

As $\boldsymbol{R}_{\mathcal{B}_{\mathrm{act}}} \notin \ker\left(\frac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}}\boldsymbol{G}(\boldsymbol{x})\right)$, this means that $\frac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}}\boldsymbol{G}(\boldsymbol{x})\boldsymbol{R}_{\mathcal{B}_{\mathrm{act}}} \neq \boldsymbol{0}_{1\times m}$ holds and there exists a $\boldsymbol{u}_{\mathcal{B}_{\mathrm{act}},\mathrm{ker}}$, such that

$$\frac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}}(\dot{\boldsymbol{x}}_{\mathrm{des}} - \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{G}(\boldsymbol{x})(\underbrace{\boldsymbol{u}_{\mathrm{c},\mathcal{I}} + \boldsymbol{R}_{\mathcal{B}_{\mathrm{act}}}\boldsymbol{u}_{\mathcal{B}_{\mathrm{act}},\mathrm{ker}}}_{=\boldsymbol{u}_{\mathrm{c}}})) \leq \alpha(V_{\max} - V(\boldsymbol{e}))$$

is fulfilled for any choice of $V_{\max}$ and α, which means that the convergence condition holds with $\dot{V}(\boldsymbol{e},\dot{\boldsymbol{e}}) \leq \alpha(V_{\max} - V(\boldsymbol{e})) \leq B_{\dot{V}}$. On the other hand, if

$$\operatorname{rank}\begin{bmatrix}\boldsymbol{A}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}}\\ \dfrac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}}\boldsymbol{G}(\boldsymbol{x})\end{bmatrix} = \operatorname{rank}(\boldsymbol{A}_{\mathrm{c},\mathcal{B}_{\mathrm{act}}})$$

holds, the input $\boldsymbol{u}_{\mathrm{c}} = \boldsymbol{u}_{\mathrm{c},\mathcal{I}}$ fulfills the invariance conditions as discussed above, whereas the left hand side of the convergence condition

$$\begin{aligned}&\frac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}}(\dot{\boldsymbol{x}}_{\mathrm{des}} - \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}_{\mathrm{c},\mathcal{I}})\\ &\quad= \underbrace{\dot{V}_{\mathrm{no}}(\boldsymbol{e},\dot{\boldsymbol{e}}_{\mathrm{no}})}_{\leq 0} - \frac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}}\boldsymbol{G}(\boldsymbol{x})\boldsymbol{W}(\boldsymbol{A}_{\mathrm{c},\mathcal{I}}\boldsymbol{W})^{+}(\boldsymbol{z}_{\mathrm{c},\mathcal{I}} - \boldsymbol{b}_{\mathrm{c},\mathcal{I}} - \boldsymbol{A}_{\mathrm{c},\mathcal{I}}\boldsymbol{u}_{\mathrm{no}})\\ &\quad\leq \underbrace{\frac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}}\boldsymbol{G}(\boldsymbol{x})\boldsymbol{W}(\boldsymbol{A}_{\mathrm{c},\mathcal{I}}\boldsymbol{W})^{+}(\overbrace{\boldsymbol{A}_{\mathrm{c},\mathcal{I}}\boldsymbol{u}_{\mathrm{no}} + \boldsymbol{b}_{\mathrm{c},\mathcal{I}}}^{\boldsymbol{z}_{\mathrm{no},\mathcal{I}}} - \boldsymbol{z}_{\mathrm{c},\mathcal{I}})}_{:=LHS(\boldsymbol{x},\boldsymbol{x}_{\mathrm{des}},\boldsymbol{x}_{\eta})}\ .\end{aligned}$$

and the right hand side of the convergence condition

$$B_{\dot{V}} \geq \underbrace{\alpha(V_{\max} - V(\boldsymbol{e}))}_{:=RHS(\boldsymbol{e})}$$

are bounded by functions $LHS(\boldsymbol{x}, \boldsymbol{x}_{\text{des}}, \boldsymbol{x}_\eta)$ and $RHS(\boldsymbol{e})$. The output functions and derivatives are continuous by Assumption 2.2, including $\boldsymbol{A}_{\text{c},\mathcal{I}}$, $\boldsymbol{A}^+_{\text{c},\mathcal{I}}$, and $\boldsymbol{b}_{\text{c},\mathcal{I}}$ and the weighting matrix $\boldsymbol{W}$ is constant. Nominal control is continuous by Assumption 2.4 and $\boldsymbol{G}(\boldsymbol{x})$ and the Lyapunov function are continuous as well. Since $\boldsymbol{z}_{\text{no},\mathcal{I}}$ is an additive and multiplicative concatenation of continuous functions, it is continuous. As the parameters $\boldsymbol{x}_\eta$ are bounded by Assumption 2.2 and the desired trajectory $\boldsymbol{x}_{\text{des}}$ is also assumed to be bounded, $\boldsymbol{z}_{\text{no},\mathcal{I}}$ and as a result $\boldsymbol{z}_{\text{c},\mathcal{I}}$ from (3.16) and the functions $LHS(\boldsymbol{x}, \boldsymbol{x}_{\text{des}}, \boldsymbol{x}_\eta)$ and $RHS(\boldsymbol{e})$ are bounded on any bounded set of states $\boldsymbol{x}$. Therefore, if

$$\mathcal{V} = \left\{ \boldsymbol{x} \,\middle|\, \exists\, t \geq t_0 : \operatorname{rank} \begin{bmatrix} \boldsymbol{A}_{\text{c},\mathcal{B}_{\text{act}}}(\boldsymbol{x}, \boldsymbol{\eta}) \\ \dfrac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}} \boldsymbol{G}(\boldsymbol{x}) \end{bmatrix} = \operatorname{rank}(\boldsymbol{A}_{\text{c},\mathcal{B}_{\text{act}}}(\boldsymbol{x}, \boldsymbol{\eta})) \right\}$$

is bounded, the input $\boldsymbol{u}_{\text{c}} = \boldsymbol{u}_{\text{c},\mathcal{I}}$ is only applied on this bounded set of states $\boldsymbol{x}$ and there exist bounds

$$\begin{aligned} B_{LHS} &= \sup_{\mathcal{V}} \left(LHS(\boldsymbol{x}, \boldsymbol{x}_{\text{des}}, \boldsymbol{x}_\eta) \right) \\ B_{RHS} &= \inf_{\mathcal{V}} (RHS(\boldsymbol{e})) = \alpha(V_{\max} - \sup_{\mathcal{V}}(V(\boldsymbol{e}))) \ . \end{aligned}$$

By choosing the parameters $V_{\max}$ and α such that they fulfill

$$\sup_{\mathcal{V}} \left(LHS(\boldsymbol{x}, \boldsymbol{x}_{\text{des}}, \boldsymbol{x}_\eta) \right) \leq \alpha(V_{\max} - \sup_{\mathcal{V}}(V(\boldsymbol{e}))) \ ,$$

the convergence condition for $\boldsymbol{u}_{\text{c}} = \boldsymbol{u}_{\text{c},\mathcal{I}}$ is met as well since

$$\begin{aligned} \dot{V}(\boldsymbol{e}, \dot{\boldsymbol{e}}) &= \frac{\partial V(\boldsymbol{e})}{\partial \boldsymbol{e}} (\dot{\boldsymbol{x}}_{\text{des}} - \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{G}(\boldsymbol{x}) \boldsymbol{u}_{\text{c},\mathcal{I}}) \\ &\leq LHS(\boldsymbol{x}, \boldsymbol{x}_{\text{des}}, \boldsymbol{x}_\eta) \leq \sup_{\mathcal{V}} \left(LHS(\boldsymbol{x}, \boldsymbol{x}_{\text{des}}, \boldsymbol{x}_\eta) \right) \\ &\leq \alpha(V_{\max} - \sup_{\mathcal{V}}(V(\boldsymbol{e}))) \leq \alpha(V_{\max} - V(\boldsymbol{e})) \leq B_{\dot{V}} \end{aligned}$$

holds for all $\boldsymbol{x} \in \mathcal{V}$. Hence in all cases, there exists at least one $\boldsymbol{u}_{\text{c}}$, which solves the minimization problem but which is not necessarily the optimal solution. Since, however the optimization is convex as the cost function is strictly convex and the optimization conditions are affine in the optimization variable, there exists a unique solution to the problem.

With the existence of a corrective control input established, we turn to the tracking error. The solution of the minimization fulfills

$$\dot{V}(\boldsymbol{e}, \dot{\boldsymbol{e}}) \leq \max(\alpha(V_{\max} - V(\boldsymbol{e})),\ \dot{V}_{\text{no}}(\boldsymbol{e}, \dot{\boldsymbol{e}}_{\text{no}})) \ .$$

Additionally for $V(\boldsymbol{e}) > V_{\max}$, the right hand side is non-positive, as $\alpha(V_{\max} - V(\boldsymbol{e})) < 0$ holds and $\dot{V}_{\text{no}}(\boldsymbol{e}, \dot{\boldsymbol{e}}_{\text{no}}) \leq 0$ by Assumption 2.4. Therefore, $\dot{V}(\boldsymbol{e}, \dot{\boldsymbol{e}}) \leq 0$ holds, which means that the Lyapunov function is bounded with $V(\boldsymbol{e}) \leq \max(V_{\max}, V(\boldsymbol{e}(t_0)))$. As $V(\boldsymbol{e})$ is a Lyapunov function showing global stability of the tracking error, it is continuous and radially unbounded, meaning that if the norm of $\boldsymbol{e}$ goes to infinity, so will $V(\boldsymbol{e})$. Therefore by the converse argument, if $V(\boldsymbol{e})$ is bounded, so is $\boldsymbol{e}$. □

Proofs from Section 3.2

Proof of Lemma 3.4. With Assumption 2.2, the I/O-linearizing transformation of $y_{\mathrm{c},i}$ with respect to the original input $\boldsymbol{u} = \boldsymbol{h}_{\mathrm{c,a}}(\boldsymbol{\chi})$ is given by (3.1). Further derivation of $y_{\mathrm{c},i}$ yields

$$\begin{aligned} y_{\mathrm{c},i}^{(r_i+k_i)} &= \frac{\mathrm{d}^{k_i}}{\mathrm{d}t^{k_i}}\left(\boldsymbol{a}_{\mathrm{c},i}^{\intercal}\boldsymbol{h}_{\mathrm{c,a}}(\boldsymbol{\chi}) + b_{\mathrm{c},i}\right) \\ &= \boldsymbol{a}_{\mathrm{c},i}^{\intercal}\boldsymbol{h}_{\mathrm{c,a}}^{(k_i)}(\boldsymbol{\chi}) + R_{i,k_i}(\boldsymbol{x},\boldsymbol{\chi},\boldsymbol{x}_{\eta}) \end{aligned}$$

which consists of one term depending on the k_i-th derivative of $\boldsymbol{h}_{\mathrm{c,a}}(\boldsymbol{\chi})$, i.e. $\boldsymbol{u}$, and a résidu $R_{i,k_i}(\boldsymbol{x},\boldsymbol{\chi},\boldsymbol{x}_{\eta})$ depending on the lower order derivatives $\boldsymbol{u}^{(j)}$, $j \in \{0, 1, ..., k_i - 1\}$. As a result from Assumption 3.3, the I/O-linearization of $\boldsymbol{u}$ with respect to $\boldsymbol{u}_{\mathrm{a}}$ is given by (3.41). This means that $\boldsymbol{u}^{(v)}$ and therefore also $h_{\mathrm{c},i}^{(r_i+v)}(\tilde{\boldsymbol{x}})$ are the first time derivatives, which are directly influenced by $\boldsymbol{u}_{\mathrm{a}}$. This yields

$$y_{\mathrm{c},i}^{(r_i+v)} = \boldsymbol{a}_{\mathrm{c},i}^{\intercal}\mathscr{L}_{\boldsymbol{\chi}\boldsymbol{G}_{\mathrm{a}}}\mathscr{L}_{\boldsymbol{\chi}\boldsymbol{f}_{\mathrm{a}}}^{v-1}\boldsymbol{h}_{\mathrm{c,a}}(\boldsymbol{\chi})\boldsymbol{u}_{\mathrm{a}} + R_{i,v}(\boldsymbol{x},\boldsymbol{\chi},\boldsymbol{x}_{\eta}) \ ,$$

where $R_{i,v}(\boldsymbol{x},\boldsymbol{\chi},\boldsymbol{x}_{\eta}) = \tilde{b}_i$. Assumption 3.3, implies that the matrix $\boldsymbol{A}_{\mathrm{a}} = \mathscr{L}_{\boldsymbol{\chi}\boldsymbol{G}_{\mathrm{a}}}\mathscr{L}_{\boldsymbol{\chi}\boldsymbol{f}_{\mathrm{a}}}^{v-1}\boldsymbol{h}_{\mathrm{c,a}}(\boldsymbol{\chi})$ is invertible and Assumption 2.2 ensures that $\boldsymbol{a}_{\mathrm{c},i}^{\intercal}$ has at least one non-zero element. Therefore $\tilde{\boldsymbol{a}}_i^{\intercal} = \boldsymbol{a}_{\mathrm{c},i}^{\intercal}\boldsymbol{A}_{\mathrm{a}}$ holds and has at least one non-zero element, thus implying the well-defined augmented relative degree $r_{\mathrm{a},i} = r_i + v$. □

Proof of Lemma 3.5. Substitution of (3.47) in (3.41) yields

$$\boldsymbol{u}^{(v)} = \boldsymbol{u}_{\mathrm{no}}^{(v)} - \sum_{j=0}^{v-1} k_j(\boldsymbol{u}^{(j)} - \boldsymbol{u}_{\mathrm{no}}^{(j)})$$

and the decoupled error dynamics

$$\boldsymbol{e}_{\mathrm{u},i}^{(v)} = -\sum_{j=0}^{v-1} k_j e_{\mathrm{u},i}^{(j)} \tag{A.3}$$

for $i \in \{1, 2, ..., m\}$, which represent linear time-invariant dynamics. By the Routh-Hurwitz stability criterion, the error dynamics are uniformly exponentially stable if the characteristic polynomial $d(s) = s^v + \sum_{j=0}^{v-1} k_j s^j$ is a Hurwitz polynomial. □

Proof of Theorem 3.4. As a result from Lemma 3.5, $\boldsymbol{u} = \boldsymbol{u}_{\mathrm{no}} + \boldsymbol{e}_{\mathrm{no}}$ is fulfilled and $\lim_{t\to\infty} \boldsymbol{e}_{\mathrm{no}} = \boldsymbol{0}_{m\times 1}$. Insertion into (2.1) yields

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})(\boldsymbol{u}_{\mathrm{no}} + \boldsymbol{e}_{\mathrm{no}}) \ ,$$

which is a representation of the nominally controlled dynamics with the additional input $\boldsymbol{e}_{\mathrm{no}}$. With [Kha96, Lemma 4.6] and under Assumption 2.1, the nominally controlled system (2.1) is ISS with respect to the additional input, if its tracking error is uniformly exponentially stable for $\boldsymbol{e}_{\mathrm{no}} = \boldsymbol{0}_{m\times 1}$. As $\boldsymbol{e}_{\mathrm{no}}$ is the output of the error dynamics (A.3), which are uniformly exponentially stable, the tracking error of the cascade connection of the error dynamics and (2.1) under nominal control is uniformly asymptotically stable according to [Kha96, Lemma 4.7] if the nominally controlled system is ISS, i.e. if it has an exponentially stable tracking error. □

Proof of Lemma 3.6. The proof is based on the continuity of the output functions, their derivatives and the system stated in Assumptions 2.1 and 2.2. For each instant of time, the state $\boldsymbol{x}$, the uncertainties $\boldsymbol{x}_\eta$ and the input error $\boldsymbol{e}_\mathrm{u}$ have a well-defined, fixed value, which means that $h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta)$, $\dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_\eta)$, …, $h_{\mathrm{c},i}^{(r_i-1)}(\boldsymbol{x}, \boldsymbol{x}_\eta)$ have a well-defined, fixed value as well. In addition, continuous functions have a bounded output on every bounded input interval. Therefore, as $t \in [t_k, t_{k+1}]$ is bounded and the input error is bounded by Assumption 2.6, the constraint output functions and the derivatives are bounded on the time interval and there exists an upper bound $h_{\mathrm{c},i\;[k]}^{(j)\;\max}(\boldsymbol{x}[k], \boldsymbol{x}_\eta[k], z_i[k])$, $\forall 0 \leq j \leq r_i - 1$, the value of which depends on the pseudo input $z_i[k]$ at the sampling instant as this determines the control input $\boldsymbol{u}[k]$. Furthermore, the invariance function as defined in (3.10)–(3.11) is a concatenation of continuous and bounded functions evaluated on a bounded time interval and hence there exists an upper bound $\Phi_{i\;[k]}^{\max}(\boldsymbol{x}[k], \boldsymbol{x}_\eta[k], z_i[k])$, the value of which depends on the pseudo input $z_i[k]$. □

Proof of Lemma 3.7. The proof is based on the continuity of the linearization as stated in Assumptions 2.1 and 2.2. For each instant of time, $\boldsymbol{x}$, $\boldsymbol{x}_\eta$ and $\boldsymbol{e}_\mathrm{u}$ have a well-defined, fixed value, which means that $\boldsymbol{a}_{\mathrm{c},i}^\intercal$ and $b_{\mathrm{c},i}$ have a well-defined, fixed value as well. In addition, both the elements of $\boldsymbol{a}_{\mathrm{c},i}^\intercal$ and $b_{\mathrm{c},i}$ are continuous functions and hence have a bounded output on every bounded input interval. Therefore, as $t \in [t_k, t_{k+1}]$ is bounded and the input error is always bounded by Assumption 2.6, the elements of $\boldsymbol{a}_{\mathrm{c},i}^\intercal$ and $b_{\mathrm{c},i}$ are within bounded sets. □

Proof of Theorem 3.5. Algorithm 1 determines a set $\mathcal{B}_\max$ of all constraints that are currently active at t_k and all constraints that have the potential of becoming active in the interval $t \in [t_k, t_{k+1}]$. The constraints that are not in $\mathcal{B}_\max$ are not in danger of violation for the corrective control input determined by Algorithm 1. This means that their respective invariance functions are currently negative and also the predicted maximum of the invariance function $\Phi_{i\;[k]}^{\max}(\boldsymbol{x}[k], \boldsymbol{x}_\eta[k], z_i[k])$ is negative.

As each constraint in $\mathcal{B}_\max$ is included in the determination of corrective control, their respective constraint satisfaction conditions are included in the optimization 3.57. Furthermore the optimization condition considers all possible future values of $\boldsymbol{a}_{\mathrm{c},i}^\intercal$ and $b_{\mathrm{c},i}$, thus making sure that the determined corrective control value fulfills the invariance condition for the entire interval $t \in [t_k, t_{k+1}]$. Therefore, the control input ensures that the pseudo inputs corresponding to active constraints fulfill $z_i \leq z_{\mathrm{c},i}$ for all $t \in [t_k, t_{k+1}]$ with $z_{\mathrm{c},i}$ from (3.53), which is designed to keep the associated invariance function at non-positive values.

Therefore, all invariance functions remain non-positive within $t \in [t_k, t_{k+1}]$. As the steps are repeated for all following time intervals, invariance is achieved by induction. □

Proofs from Section 3.3

Proof of Theorem 3.6. Each constraint $y_{\mathrm{B},i}$ with $i \in \mathcal{B}$ is associated with a CBF B_i and the sets $\mathcal{H}_i$, $\mathcal{M}_{B_i}$ from Lemma 3.8. By Assumption 2.2, $\mathcal{H}$ is not empty and therefore the constraints do not conflict. In addition, Assumption 2.2 ensures that each constraint has a well-defined relative degree and therefore the sets $\mathcal{M}_{B_i}$ are not empty as $\boldsymbol{a}_{\mathrm{B},i} \neq \boldsymbol{0}_{m\times 1}$ if B_i is a CBF.

The set $\mathcal{M}_B(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\mu})$ is the intersection of the individual $\mathcal{M}_{B_i}$.

$$\mathcal{M}_B = \bigcap_{i\in\mathcal{B}} \mathcal{M}_{B_i}$$

Hence if $\boldsymbol{u}$ lies within $\mathcal{M}_B(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\mu})$, it lies also within all $\mathcal{M}_{B_i}$. Therefore, by Lemma 3.8, $\boldsymbol{u}$ renders all $\mathrm{Int}(\mathcal{H}_i)$, $i \in \mathcal{B}$, positively invariant. As $\mathcal{H}$ is the intersection of all $\mathcal{H}_i$, $\mathrm{Int}(\mathcal{H})$ is rendered invariant by $\boldsymbol{u} \in \mathcal{M}_B(\boldsymbol{x}, \boldsymbol{x}_\eta, \boldsymbol{\mu})$. □

Proof of Theorem 3.7.
The proof is conducted in two steps. First, uniqueness and Lipschitz continuity of the solution is proved. In a second step the invariance of $\mathrm{Int}(\mathcal{H})$ is shown.

Step 1: Uniqueness and Lipschitz continuity:
Let $\boldsymbol{u}_\mathrm{c}$ be a solution of (3.74) at a point $\boldsymbol{\xi} = [\boldsymbol{x}^\intercal,\ \boldsymbol{x}_\eta^\intercal]^\intercal$. According to [MPA13, Theorem 1], if the conditions

(*i*) $w^*(\boldsymbol{\xi}) > 0$, where $w^*(\boldsymbol{\xi})$ is the solution of the linear program

$$w^*(\boldsymbol{\xi}) = \max_{(\boldsymbol{u}, w) \in \mathbb{R}^{m+1}} w \tag{A.4}$$

$$\text{s.t.} \qquad \begin{bmatrix} \boldsymbol{A}(\boldsymbol{\xi}) & \boldsymbol{1}_{l \times 1} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ w \end{bmatrix} \leq \boldsymbol{c}(\boldsymbol{\xi}) \ , \tag{A.5}$$

(*ii*) $\boldsymbol{A}(\boldsymbol{\xi})$ and $\boldsymbol{b}(\boldsymbol{\xi})$ are Lipschitz continuous at $\boldsymbol{\xi}$,

(*iii*) $\boldsymbol{u}_\mathrm{no}(\boldsymbol{\xi})$ is Lipschitz continuous at $\boldsymbol{\xi}$,

with $\boldsymbol{A}(\boldsymbol{\xi}) = [\boldsymbol{a}_{\mathrm{B},1},\ \dots,\ \boldsymbol{a}_{\mathrm{B},l}]^\intercal$, $\boldsymbol{c}(\boldsymbol{\xi}) = [\frac{\mu_1}{B_1} - b_{\mathrm{B},1},\ \dots,\ \frac{\mu_l}{B_l} - b_{\mathrm{B},l}]^\intercal$ and $\boldsymbol{a}_{\mathrm{B},i}$, $b_{\mathrm{B},i}$ from (3.70) hold at $\boldsymbol{\xi}$, then the solution $\boldsymbol{u}_\mathrm{c}$ of the QP (3.74) is unique and Lipschitz continuous at $\boldsymbol{\xi}$.

Condition (i) is based on the Mangasarian-Fromovitz constraint qualification [MF67]. If the constraint qualification holds, i.e. if $w^*(\boldsymbol{\xi}) > 0$ holds, the solution of the QP (3.74) is unique at $\boldsymbol{\xi}$. Conditions (ii) and (iii) are necessary for the Lipschitz continuity of the solution of (3.74).

As the optimization 3.74 is stricty convex, there exists a unique solution by Lemma 3.9. Hence the verification of (i) is redundant. For a formal verification of (i), the interested reader is referred to [RKH16].

Condition (ii) holds for all $\boldsymbol{x} \in \mathcal{H}$ as local Lipschitz continuity of $\boldsymbol{A}$ and $\boldsymbol{b}$ follows from Definition 3.11, which demands that a CBF B_i and its Lie derivatives $\mathscr{L}_{\boldsymbol{x}\boldsymbol{f}} B_i$ and $\mathscr{L}_{\boldsymbol{x}\boldsymbol{G}} B_i$ are locally Lipschitz. As the nominal control signal $\boldsymbol{u}_\mathrm{no}$ is assumed to be locally Lipschitz, Condition (iii) holds as well. Therefore, all three conditions hold for all $\boldsymbol{x} \in \mathcal{H}$ and the control value $\boldsymbol{u}_\mathrm{c}$ obtained by solving the QP (3.74) is unique and Lipschitz for all $\boldsymbol{x} \in \mathcal{H}$.

Step 2: Invariance of $\mathrm{Int}(\mathcal{H})$*:*
The optimization condition of (3.74) ensures that $\boldsymbol{u}_\mathrm{c}$ lies in $\mathcal{M}_B$. *Step 1* shows that the solution $\boldsymbol{u}_\mathrm{c}$ is Lipschitz continuous. With Assumptions 2.1, 2.2 and 3.7, Theorem 3.6 is applicable and $\boldsymbol{u}_\mathrm{c}$ renders the interior of the set $\mathcal{H}$ positively invariant. □

Proofs from Section 4.1

Proof of Lemma 4.1. The proof is based on the continuity of the output functions and their derivatives from Assumption 2.2. For each instant of time t, $\boldsymbol{x}$ and $\overline{\boldsymbol{x}_\eta}$ have a well-defined, fixed value, which means due to the assumed continuity $h_{\mathrm{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})$, $\dot{h}_{\mathrm{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})$, …, $\boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})$ and $b_{\mathrm{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})$ have a well-defined, fixed value as well. Furthermore, $B_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}})$, $\boldsymbol{a}_{\mathrm{B},i}^{\mathsf{T}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})$ and $b_{\mathrm{B},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}})$ are Lipschitz continuous as the CBFs are designed to fulfill Def. 3.11. In addition, continuous functions have a bounded output on every bounded input interval. As the parameter variation $\boldsymbol{\Delta}_{\boldsymbol{\eta}}$ is bounded by Assumption 4.1, $h_{\mathrm{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}+\boldsymbol{\Delta}_{\boldsymbol{\eta}})$, …, $b_{\mathrm{B},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}+\boldsymbol{\Delta}_{\boldsymbol{\eta}})$ are bounded and therefore the uncertainties described by the differences $\Delta_k(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}})$ with $k \in \{h_{\mathrm{c},i}, \dot{h}_{\mathrm{c},i}, \dots, h_{\mathrm{c},i}^{(r-1)}, a_{\mathrm{c},i,1}, \dots, a_{\mathrm{c},i,m}, b_{\mathrm{c},i}, B_i, a_{\mathrm{B},i,1}, \dots, a_{\mathrm{B},i,m}, b_{\mathrm{B},i}\}$ as defined in (4.4)–(4.12) are bounded. □

Proof of Theorem 4.1. The function $p_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}}, \gamma_i, \Delta t)$ from (3.11), which is used to determine the invariance function (3.10), is a concatenation of continuous functions and therefore also continuous. This means that it is bounded on $\mathcal{D}_{\mathrm{rob}}$ for each fixed value of $\boldsymbol{x}$ and the upper bound is determined by

$$\begin{aligned}
p_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_{\boldsymbol{\eta}}, \gamma_i, \Delta t) &= \frac{\Delta t^{r_i}}{r_i!}\gamma_i + \sum_{j=0}^{r_i-1} \frac{\Delta t^j}{j!} h_{\mathrm{c},i}^{(j)}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_{\boldsymbol{\eta}}) \\
&\leq \sup_{\boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}_{\mathrm{rob}}} \left(\frac{\Delta t^{r_i}}{r_i!}\gamma_i + \sum_{j=0}^{r_i-1} \frac{\Delta t^j}{j!} \left(h_{\mathrm{c},i}^{(j)}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) + \Delta_{h_{\mathrm{c},i}^{(j)}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}}) \right) \right) \\
&= p_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \gamma_i, \Delta t) + \sup_{\boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}_{\mathrm{rob}}} \left(\sum_{j=0}^{r_i-1} \frac{\Delta t^j}{j!} \Delta_{h_{\mathrm{c},i}^{(j)}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \boldsymbol{\Delta}_{\boldsymbol{\eta}}) \right) \\
&= p_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \gamma_i, \Delta t) + \sum_{j=0}^{r_i-1} \frac{\Delta t^j}{j!} \Delta_{h_{\mathrm{c},i}^{(j)}}^{\max_{\mathcal{D}_{\mathrm{rob}}}}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}) \\
&:= p_i^{\max}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{\mathrm{rob}}, \gamma_i, \Delta t)
\end{aligned}$$

and therefore,

$$\max_{\Delta t \geq 0} p_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}}, \gamma_i, \Delta t) \leq \max_{\Delta t \geq 0} p_i^{\max}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{\mathrm{rob}}, \gamma_i, \Delta t) \quad \forall \boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}_{\mathrm{rob}}$$

holds, which means that the robust invariance function (4.14) fulfills (4.16). □

Proof of Theorem 4.2. The inactive constraints $i \in \mathcal{B}_{\mathrm{rob}} \cap \neg \mathcal{B}_{\mathrm{rob,act}}$ are automatically satisfied for any control input as they are not in immediate danger of violation. Their invariance function is non-positive for any instance of the uncertainty by definition as their robust invariance function $\Phi_{\mathrm{rob},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}}, \mathcal{D}_{\mathrm{rob}}, \gamma_i)$ is non-positive and Theorem 4.1 holds.

The active robust constraints $i \in \mathcal{B}_{\mathrm{rob,act}}$ are included in the determination of the corrective control input. In order to enforce them independently from the current value of the uncertainties, an independent upper bound on the invariance condition (3.30) from Theorem 3.1 is determined. Using Lemma 4.1 yields

$$\begin{aligned}
\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}^{\mathsf{T}} \boldsymbol{u} \leq \left| \boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}^{\mathsf{T}} \boldsymbol{u} \right| \leq \sum_{k=1}^{m} \left| \Delta_{\boldsymbol{a}_{\mathrm{c},i,k}} \right| |u_k| &\leq \sum_{k=1}^{m} \underbrace{\max\left(\left| \Delta_{\boldsymbol{a}_{\mathrm{c},i,k}}^{\min_{\mathcal{D}_{\mathrm{rob}}}} \right|, \left| \Delta_{\boldsymbol{a}_{\mathrm{c},i,k}}^{\max_{\mathcal{D}_{\mathrm{rob}}}} \right| \right)}_{\left| \Delta_{\boldsymbol{a}_{\mathrm{c},i}} \right|_k^{\max_{\mathcal{D}_{\mathrm{rob}}}}} |u_k| \\
&= \left(\left| \boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}} \right|^{\max_{\mathcal{D}_{\mathrm{rob}}}} \right)^{\mathsf{T}} |\boldsymbol{u}| \; . \qquad \text{(A.6)}
\end{aligned}$$

Combining this result with the bound on $\Delta_{b_{\mathrm{c},i}}$ from Lemma 4.1 yields an upper bound on the invariance condition (3.30)

$$\begin{aligned}&(\boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})+\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}^{\mathsf{T}})\boldsymbol{u}+b_{\mathrm{c},i}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})+\Delta_{b_{\mathrm{c},i}}\\&\quad\leq\boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})\boldsymbol{u}+\left(\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}\right|^{\max_{\mathcal{D}_{\mathrm{rob}}}}\right)^{\mathsf{T}}|\boldsymbol{u}|+b_{\mathrm{c},i}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})+\Delta_{b_{\mathrm{c},i}}^{\max_{\mathcal{D}_{\mathrm{rob}}}}.\end{aligned}\tag{A.7}$$

Enforcing the optimization condition

$$\boldsymbol{a}_{\mathrm{c},i}^{\mathsf{T}}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})\boldsymbol{u}+\left(\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{c},i}}\right|^{\max_{\mathcal{D}_{\mathrm{rob}}}}\right)^{\mathsf{T}}|\boldsymbol{u}|+b_{\mathrm{c},i}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})+\Delta_{b_{\mathrm{c},i}}^{\max_{\mathcal{D}_{\mathrm{rob}}}}\leq z_{\mathrm{c},i}$$

which is an upper bound of all invariance conditions (3.30), results in the invariance conditions holding for all instances of the uncertainties. Therefore, by Theorem 3.1 the system state is controlled positive invariant w.r.t. all invariant sets $\mathcal{G}(\overline{\boldsymbol{x}_{\boldsymbol{\eta}}}+\boldsymbol{\Delta}_{\boldsymbol{\eta}},\boldsymbol{\gamma})$ corresponding to the uncertainties and as a result also positively invariant w.r.t. the robust invariant set $\mathcal{G}_{\mathrm{rob}}(\overline{\boldsymbol{x}_{\boldsymbol{\eta}}},\mathcal{D}_{\mathrm{rob}},\boldsymbol{\gamma})$ from Def. 4.2, which is the intersection of these sets. Similar to the case of constraints without uncertainties, the robust invariant set is a subset of the robust admissible set $\mathcal{G}_{\mathrm{rob}}(\overline{\boldsymbol{x}_{\boldsymbol{\eta}}},\mathcal{D}_{\mathrm{rob}},\boldsymbol{\gamma})\subseteq\mathcal{H}_{\mathrm{rob}}(\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})$ and therefore, positive invariance w.r.t. $\mathcal{G}_{\mathrm{rob}}(\overline{\boldsymbol{x}_{\boldsymbol{\eta}}},\mathcal{D}_{\mathrm{rob}},\boldsymbol{\gamma})$ results in robust constraint satisfaction. □

Proof of Corollary 4.2.1. The cost function $\|\boldsymbol{u}-\boldsymbol{u}_{\mathrm{no}}\|_2^2$ is strictly convex as its Hessian is the identity matrix. The constraints are a sum of a function which is linear in $\boldsymbol{u}$ and a function which is linear in $|\boldsymbol{u}|$. As $|\boldsymbol{u}|$ is a convex function and the sum of convex functions is convex, the constraints are convex. As there are no equality constraints, the optimization is strictly convex and any local minimum is a global minimum [Boy04, p.136ff.]. □

Proof of Theorem 4.3. In order to enforce the robust constraints independently from the current value of the uncertainties, an independent upper bound on the optimization condition of (3.74) is determined. Using Lemma 4.1 yields

$$\begin{aligned}\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{B},i}}^{\mathsf{T}}\boldsymbol{u}\leq\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{B},i}}^{\mathsf{T}}\boldsymbol{u}\right|\leq\sum_{k=1}^{m}\left|\Delta_{\boldsymbol{a}_{\mathrm{B},i,k}}\right||u_k|&\leq\sum_{k=1}^{m}\underbrace{\max\left(\left|\Delta_{\boldsymbol{a}_{\mathrm{B},i,k}}^{\min_{\mathcal{D}_{\mathrm{rob}}}}\right|,\left|\Delta_{\boldsymbol{a}_{\mathrm{B},i,k}}^{\max_{\mathcal{D}_{\mathrm{rob}}}}\right|\right)}_{\left|\Delta_{\boldsymbol{a}_{\mathrm{B},i}}\right|_k^{\max_{\mathcal{D}_{\mathrm{rob}}}}}|u_k|\\&=\left(\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{B},i}}\right|^{\max_{\mathcal{D}_{\mathrm{rob}}}}\right)^{\mathsf{T}}|\boldsymbol{u}|\ .\end{aligned}\tag{A.8}$$

Combining this result with the bound on $\Delta_{b_{\mathrm{B},i}}$ from Lemma 4.1 yields an upper bound on the condition of (3.74)

$$\begin{aligned}&(\boldsymbol{a}_{\mathrm{B},i}^{\mathsf{T}}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})+\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{B},i}}^{\mathsf{T}})\boldsymbol{u}+b_{\mathrm{B},i}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})+\Delta_{b_{\mathrm{B},i}}\\&\quad\leq\boldsymbol{a}_{\mathrm{B},i}^{\mathsf{T}}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})\boldsymbol{u}+\left(\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{B},i}}\right|^{\max_{\mathcal{D}_{\mathrm{rob}}}}\right)^{\mathsf{T}}|\boldsymbol{u}|+b_{\mathrm{B},i}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})+\Delta_{b_{\mathrm{B},i}}^{\max_{\mathcal{D}_{\mathrm{rob}}}}.\end{aligned}\tag{A.9}$$

In addition the right side of the condition of (3.74) is bounded by

$$\frac{\mu_i}{B_i(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})+\Delta_{B_i}^{\max_{\mathcal{D}_{\mathrm{rob}}}}}\leq\frac{\mu_i}{B_i(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})+\Delta_{B_i}}$$

Therefore, enforcing the condition

$$\boldsymbol{a}_{\mathrm{B},i}^{\mathsf{T}}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})\boldsymbol{u}+\left(\left|\boldsymbol{\Delta}_{\boldsymbol{a}_{\mathrm{B},i}}\right|^{\max_{\mathcal{D}_{\mathrm{rob}}}}\right)^{\mathsf{T}}|\boldsymbol{u}|+b_{\mathrm{B},i}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})+\Delta_{b_{\mathrm{B},i}}^{\max_{\mathcal{D}_{\mathrm{rob}}}}\leq\frac{\mu_i}{B_i(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})+\Delta_{B_i}^{\max_{\mathcal{D}_{\mathrm{rob}}}}}$$

results in the condition of (3.74) holding for all instances of the uncertainties. Therefore, by Theorem 3.7 the interior of all admissible sets $\mathcal{H}(\overline{\boldsymbol{x}_{\boldsymbol{\eta}}}+\boldsymbol{\Delta}_{\boldsymbol{\eta}})$ corresponding to the uncertainties is rendered controlled positively invariant. As a result the interior of the robust admissible set $\mathcal{H}_{\mathrm{rob}}(\overline{\boldsymbol{x}_{\boldsymbol{\eta}}},\mathcal{D}_{\mathrm{rob}})$ from Def. 4.1, which is the intersection of the individual admissible sets, is rendered controlled positively invariant as well and hence, robust satisfaction of the constraints $i \in \mathcal{B}_{\mathrm{rob}}$ is achieved. □

Proofs from Section 4.2

Proof of Lemma 4.2. As the functions Δ_k from (4.4)-(4.12) are continuous in $\boldsymbol{\Delta}_{\boldsymbol{\eta}}$ by Assumption 2.2 and by Def. 3.11, $\Delta_k^{\min_{\mathcal{D}_{p_i}}}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})$, $\Delta_k^{\max_{\mathcal{D}_{p_i}}}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}})$ exist and the sets $\mathcal{D}_k$ are bounded by Lemma 4.1. By definition and due to the continuity of the functions k, $\mathcal{D}_{p_i}$ and $\mathcal{D}_k$ fulfill

$$\mathcal{D}_{p_i} \subseteq \mathcal{D}_k$$

and therefore

$$p_i = \mathscr{P}(\mathcal{D}_{p_i}) \leq \mathscr{P}(\mathcal{D}_k)$$

with Assumption 4.2. □

Proof of Theorem 4.4. With the sets $\mathcal{T}_i$ and $\mathcal{V}_i$, the probability of an undetected violation $\mathscr{P}(\mathcal{T}_i \wedge \mathcal{V}_i)$ is expressed using $\mathcal{D}_{p_i}$ and its inverse $\neg\mathcal{D}_{p_i}$.

$$\begin{aligned}\mathscr{P}(\mathcal{T}_i \wedge \mathcal{V}_i) &= \mathscr{P}(\mathcal{T}_i \wedge \mathcal{V}_i \wedge \mathcal{D}_{p_i}) + \mathscr{P}(\mathcal{T}_i \wedge \mathcal{V}_i \wedge \neg\mathcal{D}_{p_i}) \\ &= \mathscr{P}(\mathcal{T}_i | \mathcal{V}_i \wedge \mathcal{D}_{p_i})\, \mathscr{P}(\mathcal{V}_i \wedge \mathcal{D}_{p_i}) + \mathscr{P}(\mathcal{T}_i \wedge \mathcal{V}_i | \neg\mathcal{D}_{p_i})\, \mathscr{P}(\neg\mathcal{D}_{p_i}) \qquad \text{(A.10)}\end{aligned}$$

By Theorem 4.1, $\Phi_{\mathrm{pr},i}(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}},\mathcal{D}_{p_i},\gamma_i) > 0$ holds if there exists at least one $\boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}_{p_i}$ for which $\Phi_i(\boldsymbol{x},\boldsymbol{x}_{\boldsymbol{\eta}},\gamma_i) > 0$, i.e.

$$\mathcal{V}_i \wedge \mathcal{D}_{p_i} \Rightarrow \neg\mathcal{T}_i$$

where $\neg\mathcal{T}_i$ is the inverse set of $\mathcal{T}_i$. This translates into the conditional probabilities

$$\mathscr{P}(\neg\mathcal{T}_i | \mathcal{V}_i \wedge \mathcal{D}_{p_i}) = 1 \qquad \text{and} \qquad \mathscr{P}(\mathcal{T}_i | \mathcal{V}_i \wedge \mathcal{D}_{p_i}) = 0\ .$$

Additionally, Assumption 4.2 provides

$$\mathscr{P}(\mathcal{D}_{p_i}) = p_i \qquad \text{and} \qquad \mathscr{P}(\neg\mathcal{D}_{p_i}) = 1 - p_i\ .$$

Combining these considerations with (A.10) yields

$$\begin{aligned}&\mathscr{P}(\mathcal{T}_i \wedge \mathcal{V}_i) = 0 \cdot \mathscr{P}(\mathcal{V}_i \wedge \mathcal{D}_{p_i}) + \overbrace{\mathscr{P}(\mathcal{T}_i \wedge \mathcal{V}_i | \neg\mathcal{D}_{p_i})}^{\leq 1}(1 - p_i) \\ &\mathscr{P}(\mathcal{T}_i \wedge \mathcal{V}_i) \leq 1 - p_i\end{aligned}$$

which corresponds to (4.21). □

Proof of Theorem 4.5. The set $\mathcal{D}_{p_i}$ is bounded by Assumption 4.2. Therefore the control input (4.24) robustly enforces constraints $i \in \mathcal{B}_{\mathrm{pr}}$ for all $\boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}_{p_i}$ by Theorem 4.2, i.e. $\Phi_i(\boldsymbol{x},\overline{\boldsymbol{x}_{\boldsymbol{\eta}}}+\boldsymbol{\Delta}_{\boldsymbol{\eta}},\gamma_i) \leq 0$ for $\boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}_{p_i}$. This means that

$$\mathcal{D}_{p_i} \;\Rightarrow\; \neg\mathcal{V}_i \quad \text{and} \quad \mathcal{V}_i \;\Rightarrow\; \neg\mathcal{D}_{p_i}$$

hold with $\mathcal{V}_i$ from (4.23), which translate into the probabilities

$$\begin{aligned} \mathscr{P}(\neg\mathcal{V}_i|\mathcal{D}_{p_i}) &= 1 \ , \quad \mathscr{P}(\mathcal{V}_i|\mathcal{D}_{p_i}) = 0 \\ \mathscr{P}(\neg\mathcal{D}_{p_i}|\mathcal{V}_i) &= 1 \ , \quad \mathscr{P}(\mathcal{D}_{p_i}|\mathcal{V}_i) = 0 \end{aligned}$$

The probability of violating a constraint i is given by $\mathscr{P}(\mathcal{V}_i)$. With the previous considerations and Assumption 4.2, the violation probability is expressed by

$$\begin{aligned} \mathscr{P}(\mathcal{V}_i) &= \mathscr{P}(\mathcal{V}_i \wedge \mathcal{D}_{p_i}) + \mathscr{P}(\mathcal{V}_i \wedge \neg\mathcal{D}_{p_i}) \\ &= \underbrace{\mathscr{P}(\mathcal{V}_i|\mathcal{D}_{p_i})}_{=0}\mathscr{P}(\mathcal{D}_{p_i}) + \underbrace{\mathscr{P}(\mathcal{V}_i|\neg\mathcal{D}_{p_i})}_{\leq 1}\underbrace{\mathscr{P}(\neg\mathcal{D}_{p_i})}_{=1-p_i} \\ &\leq 1 - p_i \end{aligned}$$

The probability of a non-positive invariance function is determined by the inverse set $\neg\mathcal{V}_i$

$$\mathscr{P}(\neg\mathcal{V}_i) = 1 - \mathscr{P}(\mathcal{V}_i) \geq 1 - (1 - p_i) = p_i \ .$$

This means that the probability of the control input $\boldsymbol{u} = \boldsymbol{u}_\mathrm{c}$ achieving $\Phi_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_{\boldsymbol{\eta}}, \gamma_i) \leq 0$ is at least p_i. In addition,

$$\Phi_i(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_{\boldsymbol{\eta}}, \gamma_i) \leq 0 \ \Rightarrow \ h_{\mathrm{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_{\boldsymbol{\eta}}) \leq 0$$

holds which follows from the design of the invariance function (3.10). Therefore, the probability of satisfying constraint i is given by

$$\mathscr{P}(h_{\mathrm{c},i}(\boldsymbol{x}, \overline{\boldsymbol{x}_{\boldsymbol{\eta}}} + \boldsymbol{\Delta}_{\boldsymbol{\eta}}) \leq 0) \geq \mathscr{P}(\neg\mathcal{V}_i) \geq p_i$$

meaning that at least p_i-satisfaction of each constraint $i \in \mathcal{B}_\mathrm{pr}$ is achieved. □

Proof of Theorem 4.6. The set $\mathcal{D}_{p_i}$ is bounded by Assumption 4.2. Therefore the control input (4.25) robustly enforces constraints $i \in \mathcal{B}_\mathrm{pr}$ for all $\boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D}_{p_i}$ by Theorem 4.3. This means that

$$\mathcal{D}_{p_i} \ \Rightarrow \ \mathcal{H}_i \quad \text{and} \quad \neg\mathcal{H}_i \ \Rightarrow \ \neg\mathcal{D}_{p_i}$$

hold with $\mathcal{H}_i$ given by

$$\mathcal{H}_i = \{\boldsymbol{\Delta}_{\boldsymbol{\eta}} \in \mathcal{D} |\, y_{\mathrm{B},i}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}}) \geq 0\} \ .$$

The implications translate into the probabilities

$$\begin{aligned} \mathscr{P}(\mathcal{H}_i|\mathcal{D}_{p_i}) &= 1 \ , \quad \mathscr{P}(\neg\mathcal{H}_i|\mathcal{D}_{p_i}) = 0 \\ \mathscr{P}(\neg\mathcal{D}_{p_i}|\neg\mathcal{H}_i) &= 1 \ , \quad \mathscr{P}(\mathcal{D}_{p_i}|\neg\mathcal{H}_i) = 0 \end{aligned}$$

The probability of violating a constraint i is given by $\mathscr{P}(\neg\mathcal{H}_i)$. With the previous considerations and Assumption 4.2, the violation probability is expressed by

$$\begin{aligned} \mathscr{P}(\neg\mathcal{H}_i) &= \mathscr{P}(\neg\mathcal{H}_i \wedge \mathcal{D}_{p_i}) + \mathscr{P}(\neg\mathcal{H}_i \wedge \neg\mathcal{D}_{p_i}) \\ &= \underbrace{\mathscr{P}(\neg\mathcal{H}_i|\mathcal{D}_{p_i})}_{=0}\mathscr{P}(\mathcal{D}_{p_i}) + \underbrace{\mathscr{P}(\neg\mathcal{H}_i|\neg\mathcal{D}_{p_i})}_{\leq 1}\underbrace{\mathscr{P}(\neg\mathcal{D}_{p_i})}_{=1-p_i} \\ &\leq 1 - p_i \end{aligned}$$

The probability of satisfying a constraint is determined by

$$\mathscr{P}(\mathcal{H}_i) = 1 - \mathscr{P}(\neg\mathcal{H}_i) \geq 1 - (1 - p_i) = p_i \ ,$$

meaning that at least p_i-satisfaction of each constraint $i \in \mathcal{B}_\mathrm{pr}$ is achieved. □

Proofs from Section 4.3

Proof of Theorem 4.8. The determination of corrective control for scenario invariance control (4.26) is equivalent to invariance control (3.31) with $l_{\mathrm{sc}} = \sum_{i=1}^{|\mathcal{B}_{\mathrm{sc}}|} N_i$ constraints. By Theorem 3.1, the solution $\boldsymbol{u} = \boldsymbol{u}_{\mathrm{c}}$ of the optimization ensures

$$\Phi_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s}, \gamma_i) \leq 0 \qquad \forall\ (i, s) \in \mathcal{B}_{\mathrm{sc}} \times \{1 \dots N_i\}$$

for all l_{sc} constraints and therefore also $h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s}) \leq 0$.

Furthermore, the optimization is convex, as the cost function is convex due to the identity matrix being the positive definite Hessian and the constraints being linear in $\boldsymbol{u}$. Therefore Theorem 4.7 may be applied with $\beta = \beta_i$ and $\epsilon = 1 - p_i$, thus showing that the probability of the solution $\boldsymbol{u} = \boldsymbol{u}_{\mathrm{c}}$ of the convex optimization not satisfying the optimization condition related to constraint i with at least probability $1 - \epsilon = p_i$, is at most β_i. Hence (p_i, β_i)-satisfaction of the optimization constraints (3.30) is achieved. As each optimization constraint corresponds to one invariance function, the same satisfaction probabilities hold for the invariance functions $\Phi_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}}, \gamma_i)$ and since

$$\Phi_i(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}}, \gamma_i) \leq 0 \ \Rightarrow \ h_{\mathrm{c},i}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}} + \boldsymbol{\Delta}_{\boldsymbol{\eta}}) \leq 0 \ ,$$

also for the output functions, which means that (p_i, β_i)-satisfaction of the constraints is achieved. □

Proof of Theorem 4.9. The determination of corrective control for scenario CBF-based control (4.27) is equivalent to CBF-based control (3.74) with $l_{\mathrm{sc}} = \sum_{i=1}^{|\mathcal{B}_{\mathrm{sc}}|} N_i$ constraints. By Theorem 3.7, the solution $\boldsymbol{u} = \boldsymbol{u}_{\mathrm{c}}$ of the optimization ensures

$$h_{\mathrm{B},i}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}_s}) > 0 \qquad \forall\ (i, s) \in \mathcal{B}_{\mathrm{sc}} \times \{1 \dots N_i\}$$

for all l_{sc} constraints.

Furthermore, the optimization is convex, as the cost function is convex due to the identity matrix being the positive definite Hessian and the constraints being linear in $\boldsymbol{u}$. Therefore Theorem 4.7 may be applied with $\beta = \beta_i$ and $\epsilon = 1 - p_i$, thus showing that the probability of the solution $\boldsymbol{u} = \boldsymbol{u}_{\mathrm{c}}$ of the convex optimization not satisfying the constraint i with at least probability $1 - \epsilon = p_i$, is at most β_i. Hence (p_i, β_i)-satisfaction of the optimization constraints (3.30) is achieved. As each optimization constraint corresponds to a single constraint, the same satisfaction probabilities hold for the barrier constraint functions $h_{\mathrm{B},i}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}})$, which means that (p_i, β_i)-satisfaction of the constraints is achieved. □

Proofs from Section 4.4

Proof of Theorem 4.10. The optimization is convex, as the cost function is convex due the identity matrix being the positive definite Hessian and the constraints being convex as discussed in the proofs of Corollaries 4.2.1–4.6.1 and Theorems 4.8 and 4.9. The solution $\boldsymbol{u} = \boldsymbol{u}_{\mathrm{c}}$ of (4.28) fulfills all optimization constraints. Therefore, it is possible to consider the three constraint types separately. As Theorems 4.2–4.9 all apply, robust satisfaction of the constraints $i \in \mathcal{B}_{\mathrm{rob}}$, at least p_j-satisfaction of each individual constraint $j \in \mathcal{B}_{\mathrm{pr}}$ and at least (p_k, β_k)-satisfaction of each constraint $k \in \mathcal{B}_{\mathrm{sc}}$ are achieved. □

Proofs from Chapter 5

Proof of Theorem 5.1. For the proof, we consider the agents as a single system described by (2.1) and (5.1). The minimization (5.3) is then re-written as

$$\boldsymbol{u}_{\mathrm{c}} = \underset{\boldsymbol{u}=[\boldsymbol{u}_1^\intercal,\ldots,\boldsymbol{u}_{n_{\mathrm{ag}}}^\intercal]^\intercal}{\operatorname{argmin}} \sum_{j\in\mathcal{N}_{\mathrm{ag}}} ||\boldsymbol{u}_j - \boldsymbol{u}_{\mathrm{no},j}||_2^2$$

subject to the conditions (5.4), which yields the same result as the decoupled optimization problems (5.3). Any solution $\boldsymbol{u}$ of the optimization fulfills all optimization conditions, i.e. the input $\boldsymbol{u}_j$ of each agent $j \in \mathcal{A}_i$ fulfills (5.4). Therefore, the inequality also holds for the sum of the individual inequalities

$$\sum_{j\in\mathcal{A}_i} \left(\boldsymbol{\alpha}_{i,j}^\intercal \boldsymbol{u}_j + w_{i,j}(\boldsymbol{c}) \left(\sum_{j\in\mathcal{I}_i} (\boldsymbol{\alpha}_{i,j}^\intercal \boldsymbol{u}_j) + b_i \right) \right) \leq \sum_{j\in\mathcal{A}_i} w_{i,j}(\boldsymbol{c})\ d_i\ .$$

Using (5.5) yields

$$\begin{aligned} \sum_{j\in\mathcal{A}_i} \boldsymbol{\alpha}_{i,j}^\intercal \boldsymbol{u}_j + \sum_{j\in\mathcal{I}_i} \boldsymbol{\alpha}_{i,j}^\intercal \boldsymbol{u}_j + b_i &\leq d_i \\ \sum_{j\in\mathcal{N}_{\mathrm{ag}}} \boldsymbol{\alpha}_{i,j}^\intercal \boldsymbol{u}_j + b_i &\leq d_i \end{aligned}$$

since $\mathcal{A}_i \cup \mathcal{I}_i = \mathcal{N}_{\mathrm{ag}}$, which corresponds to the condition (5.2) of the centralized system. Hence constraint satisfaction is achieved by Theorem 3.1 for invariance control and by Theorem 3.7 for CBF-based control. □

Proof of Corollary 5.1.1. The cost function of the optimization is convex as it is a quadratic function with the positive definite Hessian being the identity matrix. Furthermore, the constraints are linear in the optimization variables $\boldsymbol{u}_j$ and therefore convex. Thus the optimization is strictly convex. □

Proof of Corollary 5.1.2. Both the cost function 5.3 and the optimization condition (5.4) solely depend on one optimization variable $\boldsymbol{u}_j$, meaning that the solution of the optimization problems for the other agents do not affect the solution. Hence the optimizations may be solved independently from each other, which allows the implementation in a distributed manner. □

List of Figures

List of Tables

Bibliography

[AEG+08] A. Albu-Schäffer, O. Eiberger, M. Grebenstein, S. Haddadin, C. Ott, T. Wimböck, S. Wolf, and G. Hirzinger. "Soft Robotics". In: 15.3 (Sept. 2008), pp. 20–30.

[AFG+14] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin. "Reachability-Based Safe Learning with Gaussian Processes". In: *IEEE 53rd Annu. Conf. on Decision and Control (CDC).* 2014, pp. 1424–1431.

[AGT14] A.D. Ames, J.W. Grizzle, and P. Tabuada. "Control Barrier Function based Quadratic Programs with Application to Adaptive Cruise Control". In: *IEEE 53rd Annu. Conf. on Decision and Control (CDC).* 2014, pp. 6271–6278.

[AM99] D. Angeli and E. Mosca. "Command Governors for Constrained Nonlinear Systems". In: 44.4 (1999), pp. 816–820.

[AMO07] J. J. Abbott, P. Marayong, and A. M. Okamura. "Haptic Virtual Fixtures for Robot-Assisted Manipulation". In: *Robotics Research: Results of the 12th Int. Symp. ISRR (Springer Tracts in Advanced Robotics).* Ed. by R. Brooks S. Thrun and H. Durrant-Whyte. Vol. 28. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 49–64.

[AMP+11] Javier Alonso, Vicente Milanés, Joshué Pérez, Enrique Onieva, Carlos González, and Teresa De Pedro. "Autonomous vehicle control systems for safe crossroads". In: *Transportation research part C: emerging technologies* 19.6 (2011), pp. 1095–1110.

[AOF+03] A. Albu-Schäffer, C. Ott, U. Frese, and G. Hirzinger. "Cartesian Impedance Control of Redundant Robots: Recent Results with the DLR-Light-Weight-Arms". In: *IEEE Int. Conf. on Robotics and Automation (ICRA).* Vol. 3. Sept. 2003, pp. 3704–3709.

[ASB09] M. Althoff, O. Stursberg, and M. Buss. "Safety Assessment of Driving Behavior in Multi-Lane Traffic for Autonomous Vehicles". In: *IEEE Intelligent Vehicles Symp.* 2009, pp. 893–900.

[BBS06] L. Buşoniu, R. Babuška, and B. De Schutter. "Multi-Agent Reinforcement Learning: A Survey". In: *9th Int. Conf. on Control, Automation, Robotics and Vision (ICARCV).* 2006, pp. 1–6.

[BCL16] J. P. Barthès, L. Callebert, and D. Lourdeaux. “Priority-based contextual local decision making in multi-agent systems”. In: *IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD).* May 2016, pp. 186–191.

[BG10] M. Bürger and M. Guay. “Robust Constraint Satisfaction for Continuous-Time Nonlinear Systems in Strict Feedback Form”. In: 55.11 (Nov. 2010). Ed. by Institute of Electrical and Electronics Engineers. eng, pp. 2597–2601.

[Bla99] F. Blanchini. “Set invariance in control”. In: *Automatica* 35.11 (1999), pp. 1747–1767.

[BM99] A. Bemporad and M. Morari. “Robust Model Predictive Control: A Survey”. In: *Robustness in Identification and Control.* Ed. by A. Garulli and A. Tesi. London: Springer London, 1999, pp. 207–226.

[Boy04] S. P. Boyd. *Convex Optimization.* Ed. by L. Vandenberghe. Cambridge University Press, 2004.

[BT04] A. Bicchi and G. Tonietti. “Fast and "Soft-Arm" Tactics [Robot Arm Design]”. In: 11.2 (June 2004), pp. 22–33.

[BUH17] T. Beckers, J. Umlauft, and S. Hirche. “Stable Model-based Control with Gaussian Process Regression for Robot Manipulators”. In: *20th IFAC World Congr.* 2017.

[CC05] G. Calafiore and M.C. Campi. “Uncertain convex programs: randomized solutions and confidence levels”. In: *Mathematical Programming* 102.1 (2005), pp. 25–46.

[CG08] M.C. Campi and S. Garatti. “The Exact Feasibility of Randomized Solutions of Uncertain Convex Programs”. In: *SIAM Journal on Optimization* 19.3 (2008), pp. 1211–1230.

[CGP09] M.C. Campi, S. Garatti, and M. Prandini. “The scenario approach for systems and control design”. In: *Annu. Reviews in Control* 33.2 (Dec. 2009), pp. 149–157.

[ĆJP+16] J. Ćesić, V. Joukov, I. Petrović, and D. Kulić. “Full Body Human Motion Estimation on Lie Groups Using 3D Marker Position Measurements”. In: *16th IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids).* 2016.

[CYZ+07] C. Cai, C. Yang, Q. Zhu, and Y. Liang. “Collision Avoidance in Multi-Robot Systems”. In: *2007 International Conference on Mechatronics and Automation.* Aug. 2007, pp. 2795–2800.

[DAO+07] A. De Santis, A. Albu-Schäffer, C. Ott, B. Siciliano, and G. Hirzinger. “The skeleton algorithm for self-collision avoidance of a humanoid manipulator”. In: *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM).* Sept. 2007, pp. 1–6.

[DKW+14] A. Dietrich, M. Kimmel, T. Wimböck, S. Hirche, and A. Albu-Schäffer. “Workspace Analysis for a Kinematically Coupled Torso of a Torque Controlled Humanoid Robot”. In: *IEEE Int. Conf. on Robotics and Automation (ICRA).* June 2014, pp. 3439–3445.

[DSD+08] A. DeSantis, B. Siciliano, A. DeLuca, and A. Bicchi. “An atlas of physical human–robot interaction”. In: *J. Mech. Mach. Theory* 43.3 (2008), pp. 253–270.

[FAC10] M. Fiacchini, T. Alamo, and E. F. Camacho. “On the computation of convex robust control invariant sets for nonlinear systems”. In: *Automatica* 46.8 (2010), pp. 1334–1338.

[FBT97] D. Fox, W. Burgard, and S. Thrun. “The Dynamic Window Approach to Collision Avoidance”. In: 4.1 (1997), pp. 23–33.

[FGP+09] D. Feth, R. Groten, A. Peer, S. Hirche, and M. Buss. “Performance Related Energy Exchange in Haptic Human-Human Interaction in a Shared Virtual Object Manipulation Task”. In: *3rd Joint EuroHaptics conference and Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems.* 2009, pp. 338–343.

[FH85] T. Flash and N. Hogan. “The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model”. In: *The Journal of Neuroscience* 5.7 (July 1985), pp. 1688–1703.

[FR03] E. Freund and J. Rossman. “The basic ideas of a proven dynamic collision avoidance approach for multi-robot manipulator systems”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Cat. No.03CH37453).* Vol. 2. Oct. 2003, 1173–1177 vol.2.

[FSR01] E. Freund, M. Schluse, and J. Rossmann. “Dynamic collision avoidance for redundant multi-robot systems”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180).* Vol. 3. 2001, 1201–1206 vol.3.

[GC96] B. Gillespie and M. Cutkosky. “Stable User-Specific Haptic Rendering of the Virtual Wall”. In: *ASME Int. Mechanical Engineering Congr. & Expo.* Vol. 58. 1996, pp. 397–406.

[GG04] P. Griffiths and R.B. Gillespie. “Shared Control Between Human and Machine: Haptic Display of Automation During Manual Control of Vehicle Heading”. In: *12th Int. Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems.* 2004, pp. 358–366.

[Gil13] J. H. Gillula. “Guaranteeing Safe Online Machine Learning via Reachability Analysis”. Dissertation. Stanford University, June 2013.

[GK02] E. Gilbert and I. Kolmanovsky. “Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor”. In: *Automatica* 38.12 (2002), pp. 2063–2073.

[GP14] M. Geravand and A. Peer. “Safety Constrained Motion Control of Mobility Assistive Robots”. In: *5th IEEE RAS EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob).* 2014, pp. 1073–1078.

[GZM+16] S. Grammatico, X. Zhang, K. Margellos, P. Goulart, and J. Lygeros. “A Scenario Approach for Non-Convex Control Design”. In: 61.2 (Feb. 2016), pp. 334–345.

[HAD+08] S. Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger. “Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction”. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS).* Sept. 2008, pp. 3356–3363.

[Hog85] N. Hogan. “Impedance Control: An approach to manipulation”. In: *J. Dyn. Sys., Meas., Control* 107 (1985), pp. 1–24.

[HPS+09] H. K. Hong, S. S. Park, S. K. Song, and H. Y. Youn. “A priority-based message scheduling scheme for multi-agent system dynamically, adapting to the environment change”. In: *2009 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.* Oct. 2009, pp. 191–196.

[Hur95] A. Hurwitz. “Über die Bedingungen, unter welchen eine Gleichung nur Wurzeln mit negativen reellen Theilen besitzt”. In: *Mathematische Annalen* 46.2 (1895), pp. 273–284.

[HXA15] S. Hsu, X. Xu, and A.D. Ames. “Control Barrier Function based Quadratic Programs with Application to Bipedal Robotic Walking”. In: *American Control Conf. (ACC).* 2015, pp. 4542–4548.

[Isi95] A. Isidori. *Nonlinear Control Systems.* 3rd ed. Springer-Verlag, Berlin, 1995.

[ISO14] ISO. *Robots and robotic devices – Safety requirements for personal care robots.* ISO 13482:2014. Geneva, Switzerland: International Organization for Standardization, 2014.

[ISO16] ISO/TS. *Robots and robotic devices – Collaborative robots.* ISO 15066:2016. Geneva, Switzerland: International Organization for Standardization, 2016.

[ITO+11] S. Ide, T. Takubo, K. Ohara, Y. Mae, and T. Arai. “Real-time Trajectory Planning for Mobile Manipulator Using Model Predictive Control with Constraints”. In: *8th Int. Conf. on Ubiquitous Robots and Ambient Intelligence (URAI).* 2011, pp. 244–249.

[JA14] J. Jiang and A. Astolfi. “Shared-Control for the Kinematic Model of a Mobile Robot”. In: *IEEE 53rd Annu. Conf. on Decision and Control (CDC).* 2014, pp. 62–67.

[JG12] T. Jacobs and B. Graf. “Practical evaluation of service robots for support and routine tasks in an elderly care facility”. In: *2012 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO).* 2012, pp. 46–49.

[JH17] C. Jähne and S. Hirche. “Augmented Invariance Control for Impedance-controlled Robots with Safety Margins”. In: *IFAC-PapersOnLine.* Vol. 50. 2017, pp. 12053–12058.

[KBS+06] H.K. Kim, J. Biggs, W. Schloerb, M. Carmena, M.A. Lebedev, M.A.L. Nicolelis, and M.A. Srinivasan. “Continuous Shared Control for Stabilizing Reaching and Grasping With Brain-Machine Interfaces”. In: 53.6 (2006), pp. 1164–1173.

[KH14] M. Kimmel and S. Hirche. “Invariance Control with Chattering Reduction”. In: *IEEE 53rd Annu. Conf. on Decision and Control (CDC).* 2014, pp. 68–74.

[KH15] M. Kimmel and S. Hirche. “Active Safety Control for Dynamic Human-Robot Interaction”. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS).* Hamburg, Germany, 2015, pp. 4685–4691.

[KH16] M. Kimmel and S. Hirche. “Invariance Control with Time-varying Constraints”. In: *European Control Conf. (ECC).* 2016, pp. 867–872.

[KH17] M. Kimmel and S. Hirche. “Invariance Control for Safe Human-Robot Interaction in Dynamic Environments”. In: 33.6 (2017), pp. 1327–1342.

[Kha85] O. Khatib. “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots”. In: *IEEE Int. Conf. on Robotics and Automation (ICRA).* Vol. 2. Mar. 1985, pp. 500–505.

[Kha96] H.K. Khalil. *Nonlinear Systems.* 3rd ed. Prentice Hall, 1996.

[Kha99] O. Khatib. “Mobile manipulation: The robotic assistant”. In: *Robotics and Autonomous Systems* 26.2-3 (Feb. 1999), pp. 175–183.

[Kim11] M. Kimmel. “Invariance Control for Mobile Physical Human-Robot Interaction”. Bachelor’s Thesis, Technische Universität München. 2011.

[KJH16] M. Kimmel, C. Jähne, and S. Hirche. “Augmented Invariance Control for Systems with Smoothness Constraints”. In: *IEEE 55th Annu. Conf. on Decision and Control (CDC).* 2016, pp. 4867–4873.

[KLH12] M. Kimmel, M. Lawitzky, and S. Hirche. “6D Workspace Constraints for Physical Human-Robot Interaction using Invariance Control with Chattering Reduction”. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS).* 2012, pp. 3377–3383.

[KPHed] M. Kimmel, M. Prandini, and S. Hirche. “Enforcing Constraints With Uncertain Parameters Using Invariance Control”. In: (2018, submitted).

[KPW+ed] M. Kimmel, J. Pfort, J. Wöhlke, and S. Hirche. “Shared Invariance Control for Constraint Satisfaction in Multi-Robot Systems”. In: *The Int. Journal of Robotics Research (IJRR)* (2018, submitted).

[KSC91] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis. “Weighted round-robin cell multiplexing in a general-purpose ATM switch chip”. In: *IEEE Journal on Selected Areas in Communications* 9.8 (Oct. 1991), pp. 1265–1279.

[LAW+11] A Lawitzky, D. Althoff, D. Wollherr, and M. Buss. “Dynamic Window Approach for Omnidirectional Robots with Polygonal Shape”. In: *IEEE Int. Conf. on Robotics and Automation (ICRA).* May 2011, pp. 2962–2963.

[LFS17] Przemyslaw A. Lasota, Terrence Fong, and Julie A. Shah. “A Survey of Methods for Safe Human-Robot Interaction”. In: *Foundations and Trends in Robotics* 5.4 (2017), pp. 261–349.

[LJX15] Chen Lianghong, Huang Jian, and Ji Xiang. “A research on priority-based sequential task allocation model in Multi-Agent System”. In: *Chinese Automation Congress (CAC).* Nov. 2015, pp. 1058–1063.

[MB11] Jacob Mattingley and Stephen Boyd. “CVXGEN: a code generator for embedded convex optimization”. In: *Optimization and Engineering* 13.1 (2011), pp. 1–27.

[MB99] J. Mareczek and M. Buss. “Robust stabilization of SISO non-minimum phase nonlinear systems”. In: *IEEE 38th Annu. Conf. on Decision and Control (CDC).* Vol. 3. 1999, 2493–2494 vol.3.

[MBH+15] A. Maoudj, B. Bouzouia, A. Hentout, and R. Toumi. “Multi-agent approach for task allocation and scheduling in cooperative heterogeneous multi-robot team: Simulation results”. In: *IEEE 13th International Conference on Industrial Informatics (INDIN).* July 2015, pp. 179–184.

[MBS00] J. Mareczek, M. Buss, and G. Schmidt. “Sufficient Conditions for Invariance Control of a Class of Nonlinear Systems”. In: *IEEE 39th Annu. Conf. on Decision and Control (CDC).* Sydney, Australia, 2000, pp. 1436–1442.

[MF67] O.L. Mangasarian and S. Fromovitz. “The Fritz John Necessary OptOptimal Conditions in the Presence of Equality and Inequality Constraints”. In: *Journal of Mathematical Analysis and Applications* 17 (1967), pp. 37–47.

[MLK+12] A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche. “The Role of Roles: Physical Cooperation between Humans and Robots”. In: *The Int. Journal of Robotics Research (IJRR)* 31.13 (2012), pp. 1656–1674.

[MLM+11] J.R. Medina Hernández, M. Lawitzky, A. Mörtl, D. Lee, and S. Hirche. “An Experience-Driven Robotic Assistant Acquiring Human Knowledge to Improve Haptic Cooperation”. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS).* 2011, pp. 2416–2422.

[MLO+03] P. Marayong, Ming Li, A.M. Okamura, and G.D. Hager. “Spatial Motion Constraints: Theory and Demonstrations for Robot Guidance Using Virtual Fixtures”. In: *IEEE Int. Conf. on Robotics and Automation (ICRA).* Vol. 2. Taipei, Taiwan, Sept. 2003, pp. 1954–1959.

[MLS94] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994.

[MPA13] B. Morris, M.J. Powell, and A.D. Ames. “Sufficient Conditions for the Lipschitz Continuity of QP-based Multi-Objective Control of Humanoid Robots”. In: *IEEE 52nd Annu. Conf. on Decision and Control (CDC).* 2013, pp. 2920–2926.

[MRR+00] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. “Constrained model predictive control: Stability and optimality”. In: *Automatica* 36.6 (2000), pp. 789–814.

[NRL94] J. Najera, F. de la Rosa, and C. Laugier. “Planning Robot Motion Strategies under Geometric Uncertainty Constraints”. In: *IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems, Advanced Robotic Systems and the Real World (IROS).* Vol. 1. 1994, pp. 462–469.

[OAK+08] C. Ott, A. Albu-Schäffer, A. Kugi, and G. Hirzinger. “On the Passivity-Based Impedance Control of Flexible Joint Robots”. In: 24.2 (2008), pp. 416–429.

[PK97] S. Palanki and C. Kravaris. “Controller synthesis for time-varying systems by input/output linearization”. In: *Computers & Chemical Engineering* 21.8 (1997), pp. 891–903.

[Pon10] J.L. Pons. “Rehabilitation Exoskeletal Robotics”. In: 29.3 (2010), pp. 57–63.

[PR07] S. Prajna and A. Rantzer. “Convex Programs for Temporal Verification of Nonlinear Dynamical Systems”. In: *SIAM Journal on Control and Optimization* 46.3 (2007), pp. 999–1021.

[RK92] E. Rimon and D.E. Koditschek. “Exact Robot Navigation Using Artificial Potential Functions”. In: 8.5 (1992), pp. 501–518.

[RKH16] M. Rauscher, M. Kimmel, and S. Hirche. “Constrained Robot Control Using Control Barrier Functions”. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS).* Daejeon, Korea, 2016, pp. 279–285.

[Ros93] L.B. Rosenberg. “Virtual Fixtures: Perceptual Tools for Telerobotic Manipulation”. In: *IEEE Virtual Reality Annu. Int. Symp.* 1993, pp. 76–82.

[Ros96] J. Rossmann. “On-line collision avoidance for multi-robot systems: a new solution considering the robots’ dynamics”. In: *1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems (Cat. No.96TH8242).* Dec. 1996, pp. 249–256.

[RPH+05] K.B. Reed, M. Peshkin, M.J. Hartmann, J.E. Colgate, and J. Patton. “Kinesthetic Interaction”. In: *9th Int. Conf. on Rehabilitation Robotics (ICORR).* 2005, pp. 569–574.

[SCC+13] L. Saleh, P. Chevrel, F. Claveau, J.F. Lafay, and F. Mars. “Shared Steering Control Between a Driver and an Automation: Stability in the Presence of Driver Behavior Uncertainty”. In: 14.2 (2013), pp. 974–983.

[SK08] B. Siciliano and O. Khatib, eds. *Springer Handbook of Robotics.* Springer-Verlag Berlin Heidelberg, 2008.

[SKK03] R. Suda, K. Kosuge, and H. Kakuya. “Object-Impedance-Based Cooperative Handling of Object by Mobile Robot Helper and Human Using Visual and Force Information”. In: *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM).* Vol. 1. 2003, pp. 592–597.

[SM15] G. Schildbach and M. Morari. “Scenario MPC for Linear Time-Varying Systems with Individual Chance Constraints”. In: *American Control Conf. (ACC).* 2015, pp. 415–421.

[SR11] C. Schlette and J. Rossmann. “Motion Control Strategies for Humanoids Based on Ergonomics”. In: ed. by S. Jeschke, H. Liu, and D. Schilberg. Vol. 2. Springer Berlin Heidelberg, 2011. Chap. 4th International Conference on Intelligent Robotics and Applications (ICIRA), pp. 229–240.

[SS12] K. Sekiguchi and M. Sampei. “Change of controller based on partial feedback linearization with time-varying function”. In: *IEEE 51st Annu. Conf. on Decision and Control (CDC).* 2012, pp. 3557–3563.

[Sta06] B. Stanczyk. “Development and Control of an Anthropomorphic Telerobotic System”. PhD thesis. Technische Universität München, 2006.

[SWB07] M. Sobotka, J. Wolff, and M. Buss. “Invariance Controlled Balance of Legged Robots”. In: *European Control Conf. (ECC).* 2007, pp. 3179–3186.

[SWB08] M. Scheint, J. Wolff, and M. Buss. “Invariance Control in Robotic Applications: Trajectory Supervision and Haptic Rendering”. In: *American Control Conf. (ACC).* Seattle, USA, 2008, pp. 1436–1442.

[TCG16] B. Tiwari, R. Chandraker, and N. Goel. “Comparative analysis of different lottery bus arbitration techniques for SoC communication”. In: *International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT).* Mar. 2016, pp. 495–499.

[TTH13] Z.-L. Tang, K. P. Tee, and W. He. “Tangent Barrier Lyapunov Functions for the Control of Output-Constrained Nonlinear Systems”. In: *3rd IFAC Int. Conf. on Intelligent Control and Automation Science.* Ed. by P. M. Ferreira. Vol. 3. 1. Chengdu, China, 2013, pp. 449–455.

[UBK+17] J. Umlauft, T. Beckers, M. Kimmel, and S. Hirche. “Feedback Linearization using Gaussian Processes”. In: *IEEE 56th Annu. Conf. on Decision and Control (CDC).* 2017.

[Van01] R.J. Vanderbei. *Linear Programming: Foundations and Extensions.* 2nd ed. Vol. 37. International Series in Operations Research & Management Science. Springer US, 2001.

[VB13] M. Vasic and A. Billard. “Safety Issues in Human-Robot Interactions”. In: *IEEE Int. Conf. on Robotics and Automation (ICRA).* 2013, pp. 197–204.

[VKS07] A. Vahidi, I. Kolmanovsky, and A. Stefanopoulou. “Constraint Handling in a Fuel Cell System: A Fast Reference Governor Approach”. In: 15.1 (2007), pp. 86–98.

[WB04] J. Wolff and M. Buss. “Invariance Control Design for Nonlinear Control Affine Systems under Hard State Constraints”. In: *6th IFAC Symp. on Nonlinear Control Systems (NOLCOS).* Stuttgart, Germany, 2004, pp. 711–716.

[WH04] A.G. Wills and W.P. Heath. “Barrier function based model predictive control”. In: *Automatica* 40.8 (2004), pp. 1415–1422.

[Wil03] A. Wills. “Barrier Function Based Model Predictive Control”. PhD thesis. The University of Newcastle, 2003.

[WS15] G. Wu and K. Sreenath. “Safety-Critical and Constrained Geometric Control Synthesis using Control Lyapunov and Control Barrier Functions for Systems Evolving on Manifolds”. In: *American Control Conf. (ACC).* 2015, pp. 2038–2044.

[WSB90] S.D. Whitehead, R.S. Sutton, and D.H. Ballard. “Advances in Reinforcement Learning and Their Implications for Intelligent Control”. In: *5th IEEE Int. Symp. on Intelligent Control.* Vol. 2. 1990, pp. 1289–1297.

[Xu18] Xiangru Xu. “Constrained control of input–output linearizable systems using control sharing barrier functions”. In: *Automatica* 87.Supplement C (2018), pp. 195–201.

[Yos85] T. Yoshikawa. “Dynamic manipulability of robot manipulators”. In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation.* Vol. 2. Mar. 1985, pp. 1033–1038.

[ZRK+04] M. Zinn, B. Roth, O. Khatib, and J.K. Salisbury. "A New Actuation Approach for Human Friendly Robot Design". In: *The Int. Journal of Robotics Research (IJRR)* 23.4–5 (2004), pp. 379–398.