Technische Universität München
TUM School of Computation, Information and Technology

Scalable Robust Controller Synthesis for Computing Safe Sets

Felix Kevin Gruber

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:
Prof. Dr. Cristina Piazza

Prüfende der Dissertation:
1. Prof. Dr.-Ing. Matthias Althoff
2. Prof. Dr.-Ing. Sandra Hirche

Die Dissertation wurde am 27.11.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 18.06.2024 angenommen.

Abstract

Autonomous systems have the potential to permeate our day-to-day lives deeply. For instance, such systems will transform mobility, increase productivity, reduce costs, and use limited computational and environmental resources optimally. Formal safety guarantees must be provided to leverage these systems in safety-critical applications where human lives are at stake. Thus, the safety of the autonomous system must be formally ensured for an infinite time horizon despite disturbances.

In this thesis, we address this issue by computing safe sets along with corresponding safety-preserving controllers. If the initial state of the autonomous system lies within such a safe set, the corresponding safety-preserving controller guarantees the safety of this system at all times. In the literature, a wide variety of robust control approaches exist for computing safe sets. However, these approaches typically suffer from an exponential computational complexity with respect to the problem dimension or excessive conservativeness. To present scalable algorithms for computing nonconservative safe sets, we combine scalable reachability analysis and convex optimization. This combination allows us to efficiently determine safe sets with minimum or maximum volume.

The efficient computation of these safe sets is beneficial not only for leveraging autonomous systems in safety-critical applications but also for improving other popular control methods. Thus, we also present a scalable robust model predictive control approach that uses our safe sets as terminal sets. In particular, we constrain the state at the end of the finite prediction horizon to lie within our safe set. In addition to model predictive control, we also integrate our safe sets into supervisory safety filters. Such filters aim to minimally modify the desired control input of an unverified high-performance controller while formally guaranteeing safety. We achieve this goal by enforcing the state to always stay within our safe set. We also use the concept of safe sets to verify the safety of autonomous vehicles quickly. In particular, the corresponding safety-preserving controller overwrites the desired control input if the planned trajectory of the controlled autonomous vehicle intersects the reachable set of another traffic participant. To evaluate the performance of our different robust control approaches, we consider various numerical examples taken from the literature.

Zusammenfassung

Autonome Systeme haben das Potenzial, unser tagtägliches Leben tiefgreifend zu durchdringen. Derartige Systeme werden beispielsweise unsere Mobilität transformieren, Produktivität steigern, Kosten senken und die begrenzten Rechenleistungen sowie Umweltressourcen optimal nutzen. Um diese Systeme in sicherheitskritischen Anwendungen einsetzen zu können, bei denen Menschenleben auf dem Spiel stehen, müssen formale Sicherheitsgarantien gegeben werden. Folglich muss die Sicherheit des autonomen Systems trotz Störungen für einen unendlichen Zeithorizont formal gewährleistet sein.

Die vorliegende Dissertation befasst sich mit dieser Problemstellung, indem sichere Mengen zusammen mit zugehörigen, sicherheitserhaltenden Reglern berechnet werden. Wenn sich der Anfangszustand des autonomen Systems innerhalb einer solchen sicheren Menge befindet, garantiert der zugehörige, sicherheitserhaltende Regler die Sicherheit dieses Systems zu jeder Zeit. In der Literatur existiert eine Vielzahl von robusten Regelungsansätzen zur Berechnung sicherer Mengen. Diese Ansätze leiden jedoch üblicherweise unter einer exponentiellen Komplexität bezüglich der Dimension des Problems oder einer übermäßigen Konservativität. Um skalierbare Algorithmen für die Berechnung nicht-konservativer, sicherer Mengen zu präsentieren, kombiniert die vorliegende Dissertation skalierbare Erreichbarkeitsanalyse und konvexe Optimierung. Diese Kombination ermöglicht die effiziente Bestimmung sicherer Mengen mit minimalem oder maximalem Volumen.

Die effiziente Berechnung dieser sicheren Mengen ist nicht nur für den Einsatz von autonomen Systemen in sicherheitskritischen Anwendungen von Vorteil, sondern auch für die Verbesserung anderer weitverbreiteter Regelungsansätze. Deshalb wird auch ein skalierbarer Ansatz zur robusten, modellprädiktiven Regelung präsentiert, der die sicheren Mengen als Endmengen verwendet. Hierbei ist der Zustand am Ende des endlichen Prädiktionshorizonts derart beschränkt, dass er sich stets in einer sicheren Menge befinden muss. Neben der modellprädiktiven Regelung werden die sicheren Mengen auch in übergeordnete Sicherheitsfilter integriert. Das Ziel derartiger Filter ist es, den gewünschten Steuereingang eines ungeprüften, leistungsfähigen Reglers minimal zu verändern und gleichzeitig die Sicherheit formal zu gewährleisten. Dieses Ziel wird dadurch erreicht, dass sich der Zustand jederzeit innerhalb einer sicheren Menge befinden muss. Darüber hinaus nutzt die vorliegende Dissertation das Konzept der sicheren Mengen, um die Sicherheit von autonomen Fahrzeugen so schnell wie möglich formal zu verifizieren. Hierbei überschreibt der zugehörige, sicherheitserhaltende Regler die gewünschten Steuereingänge, falls die geplante Trajektorie des geregelten, autonomen Fahrzeugs die erreichbare Menge eines anderen Verkehrsteilnehmers schneidet. Um die Leistungsfähigkeit der verschiedenen robusten Regelungsansätze zu bewerten, wird eine Vielzahl numerischer Beispiele aus der Literatur betrachtet.

Acknowledgments

First and foremost, I would like to thank Prof. Matthias Althoff for being a great advisor and inspiring mentor. I highly appreciate the freedom he gave me to explore my intellectual interests, the time he spent providing accurate feedback on my paper drafts, and the excellent working environment in his research group. I am also profoundly grateful to all European taxpayers who enable such independent research.

Thanks also to Prof. Murat Arcak, who supervised my Master's thesis, taught me how to define challenging research problems, and encouraged me to pursue a PhD. A big mahalo also goes out to Prof. Markus Maurer, who gave me a warm and hearty welcome to the research community at my first conference in Hawaii.

Special thanks to my awesome office mates Dr. Stefanie Manzinger, Anna-Katharina Rettinger, and Egon Ye for having intellectually stimulating discussions, being brilliant wildlife observers, and their (mostly) defensive Nerf gun use. For the great atmosphere at work, I am also grateful to my excellent TUM colleagues Adrian Kulmburg, Prof. Amr Alanwar, Dr. Bastian Schürmann, Carmella Schürmann, Dr. Christian Pek, Christina Miller, Constantin Dresel, Daniel Auge, Edmond Irani Liu, Emeç Erçelik, Etienne Müller, Gerald Würsching, Hanna Krasowski, Prof. Jagat J. Rath, Lukas Schäfer, Mark Wetzlinger, Dr. Markus Koschi, Matthias Mayer, Moritz Klischat, Dr. Niklas Kochdumper, Paul Maroldt, Roman Hölzl, Sebastian Maierhofer, Dr. Silvia Breşug, Stefan Liu, Victor Gaßmann, and Xiao Wang. I would also like to thank Dr. Ph.D. Alexander Lenz, Amy Bücherl, Dr. Daniel Renjewski, Dr. Morteza Hashemi Farzaneh, Dr. Sina Shafaei, and Ute Lomp for the major help regarding all administrative matters.

I am also genuinely fortunate to have supervised and instructed numerous excellent students whose curiosity and actions broadened my horizon in many ways. In addition, I thank all M.Sc. Informatics applicants that provided the module descriptions of their study program in paper form for helping me to gain hands-on experience in search algorithms.

Finally, I want to thank my family very much for their unconditional love, encouragement, and continuous support throughout my entire life. I am deeply grateful to my parents, Petra and Karl, and my siblings, Kendra and Fabian, for always being there for me. In addition, I would like to thank my grandmother Helga, who was always full of positive energy and sadly passed away during my PhD. Thanks also to my brother-out-law Daniel for being the best agile coach and my niece Happy for being such a good girl. My deepest thanks to my partner Alicia for being my primary source of energy and a welcome distraction from writing this thesis.

Contents

List of Algorithms

List of Figures

List of Symbols and Acronyms

Sets

$\mathbb{B}$	set of Booleans
$\mathbb{N}$	set of natural numbers
$\mathbb{N}_{>0}$	set of positive natural numbers
$\mathbb{N}_{[m,n]}$	set of natural numbers within $[m, n]$, which is equivalent to $\{m, m+1, \dots, n\} \subseteq \mathbb{N}$
$\mathbb{R}$	set of real numbers
$\mathbb{R}_{\geq 0}$	set of nonnegative real numbers
$\mathbb{R}_{>0}$	set of positive real numbers
$\mathbb{R}^n$	set of real n-vectors with $n \in \mathbb{N}$
$\mathbb{R}^{m \times n}$	set of real $(m \times n)$-matrices with $m, n \in \mathbb{N}$
$2^{\mathbb{R}^n}$	set of all subsets of $\mathbb{R}^n$, which is also known as the power set of $\mathbb{R}^n$

Conventions

We use the following conventions throughout this thesis:

- Functions and operators are denoted by typewriter letters, e.g., $\texttt{diag}(\cdot)$ and $\texttt{center}\,(\cdot)$.
- The origin is denoted by $\{\mathbf{0}\}$, the empty set is denoted by $\emptyset$, and sets in the Euclidean space are denoted by calligraphic letters, e.g., $\mathcal{X}$ and $\mathcal{Z}$.
- The set of Booleans $\mathbb{B}$ comprises two elements: “true” and “false”. The logical equality, nonequality, conjunction, and disjunction are denoted by $\equiv$, $\not\equiv$, $\wedge$, and $\vee$, respectively.
- The square identity matrix is denoted by I, a column vector of ones is denoted by $\mathbf{1}$, and a matrix of zeros is denoted by $\mathbf{0}$. The dimensions of these matrices are determined from the context.
- The i^{th} element of the vector $v \in \mathbb{R}^n$ or of the list v of length n with $i \in \mathbb{N}_{[1,n]}$ is denoted by $v^{(i)}$.
- The absolute value, $\leq$, $<$, $=$, $>$, and $\geq$ are applied elementwise, e.g., $|v|$ is equivalent to $\begin{bmatrix} \left|v^{(1)}\right| & \left|v^{(2)}\right| & \dots & \left|v^{(n)}\right| \end{bmatrix}^T$ for any vector $v \in \mathbb{R}^n$.

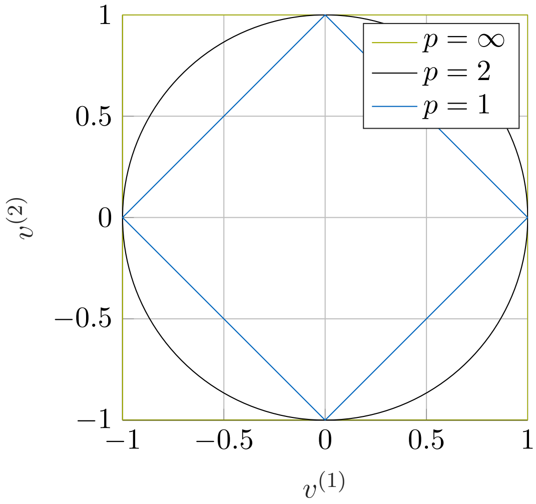


Figure 1: Unit balls in $\mathbb{R}^2$ corresponding to different p-norms.

Norms and Functions

The p-norm of a vector $v \in \mathbb{R}^n$ with $p \geq 1$ is defined by

$$\|v\|_p = \left(\sum_{i=1}^{n} \left| v^{(i)} \right|^p \right)^{1/p} .$$

The unit balls in $\mathbb{R}^2$ corresponding to different p-norms are visualized in Fig. 1. In addition to these vector norms, we introduce important matrix norms subsequently. The 1-norm of a matrix $M \in \mathbb{R}^{m \times n}$ is defined by

$$\|M\|_1 = \max \left(\mathbf{1}^T |M| \right) ,$$

i.e., it is the maximum absolute column sum of M. Conversely, the infinity norm of M is defined by

$$\|M\|_\infty = \max \left(|M| \, \mathbf{1} \right) ,$$

i.e., it is the maximum absolute row sum of M. In addition, the Frobenius norm of M is defined by

$$\|M\|_F = \sqrt{\mathtt{trace} \left(M^T M \right)},$$

where `trace` returns the sum of the diagonal elements of the square input matrix.

The function `diag` returns a diagonal matrix with the input on the diagonal if the input is a vector; otherwise, `diag` returns a vector of the diagonal elements of the square input matrix. For instance, $\mathtt{trace}\,(M)$ equals $\mathbf{1}^T \mathtt{diag}\,(M)$ for any square matrix $M \in \mathbb{R}^{n \times n}$.

Acronyms

Notation	Description	Page List
COP	convex optimization problem	5, 7–9, 11, 15, 17, 20, 25, 28–30, 38, 42–46, 48–53, 55, 58, 63, 65, 67, 68, 72, 77, 84, 86, 89, 93, 98–101, 103–106, 132
CT	continuous-time	xv, 18–20, 24, 25, 29–31, 33, 35–39, 42, 44, 46, 48–57, 59–62, 76, 77, 79, 80, 85, 87, 88, 97, 101, 110, 111, 119, 120
DT	discrete-time	xv, 20–25, 27, 29, 30, 39–42, 46, 49, 53, 58, 60, 63, 65, 69, 76, 78, 87, 97–101, 104–106, 108, 110
LMI	linear matrix inequality	64
LQR	linear-quadratic regulator	28, 30, 41, 53–60, 68, 75, 79, 91, 106, 108, 111, 113
LTI	linear time-invariant	3, 19–21, 28, 29, 77, 93
MPC	model predictive control	xv, 4, 5, 27, 41, 42, 74–77, 79, 81–85, 88, 91–93, 96, 103, 131, 132

Notation	Description	Page List
MRCI	maximal robust control invariant	xiii, xv, 24, 27, 28, 46, 58, 60, 63, 68, 105, 107, 109, 110
mRPI	minimal robust positively invariant	xiii, 21–23, 39–42, 46, 49, 69, 87
MSD	mass-spring-damper	xxi, 53, 68–71, 73, 74, 106
NASA	National Aeronautics and Space Administration	1
PID	proportional-integral-derivative	7
RCI	robust control invariant	xiii, xv, xxi, 24, 27–30, 38, 39, 42–45, 51–53, 55, 58, 60, 62–65, 68–70, 72, 74, 96, 103, 105, 107, 109, 131
RPI	robust positively invariant	xiii, xv, 21–23, 27, 42, 78, 81, 87, 88
SDP	semidefinite programming	28, 52, 63

List of Tables

List of Theorems

1 Introduction

The failure of control systems can have disastrous consequences [1], ranging from significant financial losses to human deaths. For instance, the Lewis spacecraft was an Earth-orbiting satellite launched by the National Aeronautics and Space Administration (NASA) on 23 August 1997, which lost contact three days later and burned up soon after [2]. The NASA Investigation Board "found that the loss of the Lewis Spacecraft was the direct result of an implementation of a technically flawed Safe Mode in the Attitude Control System. This error was made fatal to the spacecraft by the reliance on that unproven Safe Mode by the on orbit operations team and by the failure to adequately monitor spacecraft health and safety during the critical initial mission phase" [2]. Because this unverified safe mode failed to expose the solar panels of the satellite to sunlight, as shown in Fig. 1.1a, the onboard batteries quickly discharged, and millions of dollars were lost.

Another example highlighting the dangers of using unverified control systems in safety-critical applications is the radiation therapy machine Therac-25, shown in Fig. 1.1b. Between 1985 and 1987, at least six patients were exposed to massive radiation overdoses of this medical device [3, Sec. 8.2], resulting in severe injuries and three deaths. An investigation revealed that many software bugs were already present in predecessor devices [4], namely, the Therac-6 and the Therac-20. However, these issues were not detected in the older machines because they used hardware safety interlocks instead of software checks to prevent dangerous radiation overexposure. In summary, control systems can fail in catastrophic ways. Such failures are unacceptable when using these systems in safety-critical applications where human lives are at stake.

Traditionally, developing sophisticated control systems involves extensive simulating, testing, and debugging. However, this standard procedure can only reveal the presence of errors but not guarantee their absence [5], also known as "absence of evidence is not evidence of absence" [6]. In addition, only a tiny fraction of the state and input spaces can be covered because many combinations typically exist. This problem is getting even more serious due to the growing complexity of modern hybrid or cyber-physical systems, which exhibit not only discrete but also continuous dynamics. To address these issues, formal methods have been proposed that specify, verify, and synthesize systems in an automated and mathematically rigorous way. Thus, these approaches can obtain correct-by-construction control systems suitable for safety-critical applications. In the following section, a brief overview of formal methods is provided.

1.1 Formal Methods

Formal verification methods are widely used in computer science to formally verify the correct behavior of software and hardware, such as computer programs and electronic circuits [7, 8]. For instance, these approaches formally verify that a system never reaches an unsafe state or a deadlock state [9], i.e., a state in which it is trapped forever. These formal verification

(a) Lewis spacecraft[1]. **(b)** Radiation therapy machine Therac-25[2].

Figure 1.1: Famous failures of control systems.

methods rigorously check the correctness of specified system properties, usually expressed as logic specifications. Nowadays, there exists a wide variety of such methods.

One approach for formally verifying systems is automated theorem proving [10, 11]. This technique uses deduction and a set of previously verified theorems to prove the correctness of a given specification. Although theorem proving is a powerful tool that can deal with large-scale systems, finite termination of the algorithm cannot be guaranteed. In addition, no counterexample is provided in the case of falsification, resulting in a challenging error analysis.

Another popular formal verification approach is model checking [12, 13]. Based on a model of the underlying discrete dynamics, e.g., given by a simple finite state machine, it is exhaustively checked whether the temporal logic specification is met; otherwise, in contrast to theorem proving, a counterexample is provided. However, model checking approaches typically suffer from an infamous exponential computational complexity with respect to the state space dimension [14], which is also known as the state explosion problem or the curse of dimensionality:

> “In view of all that we have said in the foregoing sections, the many obstacles we appear to have surmounted, what casts the pall over our victory celebration? It is the curse of dimensionality, a malediction that has plagued the scientist from the earliest days.” (Richard E. Bellman [15, p. 94])

Formally verifying cyber-physical systems is an even more challenging task because they exhibit not only discrete but also continuous dynamics. Thus, verifying these systems would require a standard model checker to verify infinitely many combinations. To address this issue, a finite state system abstraction of the continuous dynamics is constructed, i.e., a discrete one suitably approximates the original continuous system. Then, model checking can be applied to this finite abstraction.

As an alternative for formally verifying cyber-physical systems, set-based reachability analysis can be used [16]. This method computes reachable sets of a system, i.e., the set of states that contains all possible state trajectories starting from a set of initial states, as illustrated in Fig. 1.2. Thus, reachability analysis is also crucial in abstraction-based approaches for suitably approximating the continuous system by a finite abstraction. Moreover, if the reachable set of a system never intersects a given forbidden or unsafe region of the state space, the safety of

[1] Picture is taken from https://space.skyrocket.de/doc_sdat/lewis.htm.

[2] Picture is taken from https://interestingengineering.com/when-bad-programming-turns-deadly.

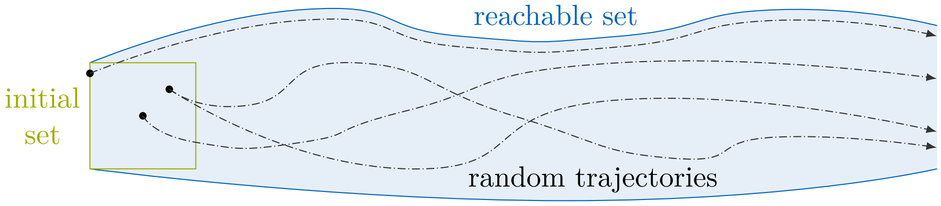


Figure 1.2: Reachability analysis. The reachable set of a system contains all possible state trajectories that start from a set of initial states, which is also known as the initial set.

the system is formally guaranteed despite bounded disturbances. Thus, reachability analysis is widely used in safety-critical applications, such as autonomous driving [17, 18], biological and medical systems [19, 20], power systems [21, 22], and robotics [23, 24]. In addition, dealing with linear time-invariant (LTI) systems having up to a billion dimensions can be achieved by tailored system order reduction and decomposition techniques [25–27]. Therefore, reachability analysis is well suited for formally verifying the safety of large-scale dynamical systems.

Up to now, we have mainly focused on the verification of different system classes. Nevertheless, formal methods can be used not only to verify systems but also to synthesize controllers that enforce the closed-loop system to satisfy temporal logic specifications [28, 29]. Using model checking algorithms, correct-by-construction controllers have been synthesized for discrete systems [30]. As for the verification, dealing with continuous dynamics is achieved by constructing a finite state system abstraction and synthesizing a controller based on this system approximation [31–33]. Because the construction of discrete system approximations typically relies on discretizing the continuous state space, abstraction-based methods severely suffer from the curse of dimensionality [34, 35].

One of the essential temporal logic specifications is safety, as seen by the failures of the Lewis spacecraft and the radiation therapy machine Therac-25 in Fig. 1.1. In the robust controller synthesis setting, safety refers to the robust satisfaction of the state and input constraints for an infinite time horizon in an uncertain environment, i.e., it can be seen as a formal robustness guarantee against bounded disturbances. A straightforward way to achieve safety is the construction of a safe set along with a corresponding safety-preserving controller. Such controllers formally guarantee robust constraint satisfaction at all times if the initial state of the system lies within the safe set, as illustrated in Fig. 1.3. A wide variety of approaches exist to compute safe sets, which are often desired to have minimum or maximum volume, depending on the specific application. However, these approaches typically suffer from an exponential computational complexity with respect to the problem dimension or an excessive conservativeness [36–40]. Thus, the characterization and efficient computation of nonconservative safe sets is an open research problem.

In this thesis, we address this issue by proposing scalable algorithms for computing nonconservative safe sets along with corresponding set-based, safety-preserving controllers. Because we combine scalable reachability analysis and convex optimization, the computational complexity of our algorithms is only polynomial with respect to the problem dimension. This low complexity allows us to compute safe sets for higher-dimensional systems compared to existing approaches in the literature. The efficient computation of safe sets is beneficial not only for leveraging autonomous systems in safety-critical applications but also for enhancing other popular control

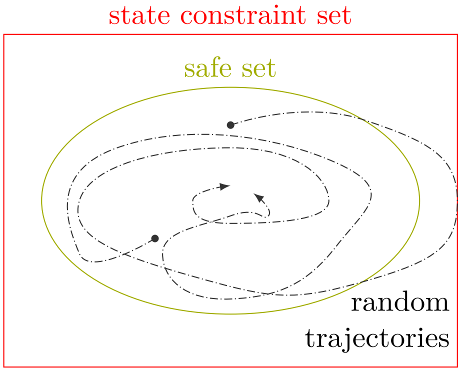


Figure 1.3: Safe set and two random state trajectories. The trajectories start in the safe set and never leave the state constraint set, which is the set of admissible states.

approaches. Thus, we also integrate our safe sets into model predictive control (MPC) [41–43]. In particular, we constrain the state at the end of the finite prediction horizon to lie within our safe set. In addition to MPC, we integrate our safe sets into supervisory safety filters [44, 45]. Such filters aim to minimally modify the desired control input of an unverified controller while formally guaranteeing safety. We achieve this goal by enforcing the state to always stay within our safe set. In this thesis, we also use the concept of safe sets to quickly verify the safety of autonomous vehicles. In particular, the corresponding safety-preserving controller overwrites the desired control input if the planned trajectory of the controlled autonomous vehicle intersects the reachable set of another traffic participant. Before we propose our novel robust control approaches, we present our publications contributing to this thesis in the following section.

1.2 Publications

This thesis is based on the following publications:

- [46] F. Gruber and M. Althoff. Anytime safety verification of autonomous vehicles. In *IEEE Conference on Intelligent Transportation Systems*, pages 1708–1714, 2018. `doi:10.1109/ITSC.2018.8569950`

- [47] F. Gruber and M. Althoff. Scalable robust model predictive control for linear sampled-data systems. In *IEEE Conference on Decision and Control*, pages 438–444, 2019. `doi:10.1109/CDC40024.2019.9029873`

- [48] N. Kochdumper, F. Gruber, B. Schürmann, V. Gaßmann, M. Klischat, and M. Althoff. AROC: A toolbox for automated reachset optimal controller synthesis. In *Conference on Hybrid Systems: Computation and Control*, pages 1–6, 2021. `doi:10.1145/3447928.3456703`

- [49] F. Gruber and M. Althoff. Computing safe sets of linear sampled-data systems. *IEEE Control Systems Letters*, 5(2):385–390, 2021. `doi:10.1109/LCSYS.2020.3002476`

- [50] F. Gruber and M. Althoff. Scalable robust output feedback MPC of linear sampled-data systems. In *IEEE Conference on Decision and Control*, pages 2563–2570, 2021. `doi:10.1109/CDC45484.2021.9683384`
- [51] F. Gruber and M. Althoff. Scalable robust safety filter with unknown disturbance bounds. *IEEE Transactions on Automatic Control*, 68(12):7756–7770, 2023. `doi:10.1109/TAC.2023.3292329`
- [52] L. Schäfer, F. Gruber, and M. Althoff. Scalable computation of robust control invariant sets of nonlinear systems. *IEEE Transactions on Automatic Control*, 69(2):755–770, 2024. `doi:10.1109/TAC.2023.3275305`

1.3 Organization of this Thesis

This thesis is organized as follows: In Chapter 2, we present essential mathematical concepts used throughout this thesis. To ensure the scalability of our robust control algorithms, the computational complexity of our approaches is always polynomial with respect to the problem dimension. In particular, we obtain the optimal control inputs as the solution of an efficiently-solvable convex optimization problem (COP). Because the constraint set of such an optimization problem must be convex, we present important representations of convex sets. Using zonotopes as a set representation, the computational complexity of our over-approximative reachability analysis is only cubic with respect to the state space dimension. Thus, solving COPs and using such reachability analysis enables the scalability of our robust control methods. Finally, we introduce important invariant sets and give an overview of the used software toolboxes.

In Chapter 3, we compute zonotopic safe sets along with corresponding safety-preserving controllers of sampled-data systems. These cyber-physical systems evolve in continuous time and are controlled by clocked digital controllers. We use state and disturbance feedback controllers to obtain a simple controller structure, which provides piecewise constant control inputs at periodic sampling times. Incorporating these controllers into our reachability analysis allows us to compute safe sets of large-scale sampled-data systems. Because safe sets are usually desired to have minimum or maximum volume, we propose several approaches for computing such sets. Finally, to evaluate the performance of the approaches and to validate the safety guarantees of their safety-preserving controllers, we consider multiple numerical examples taken from the literature.

In Chapter 4, we incorporate our safe sets along with corresponding safety-preserving controllers into (robust) MPC, which is one of the most popular control methods these days[3]. Thus, we first present the important concept of MPC, where an optimal control problem is iteratively solved online on a moving horizon. We explicitly consider all online computation times because such online computational delays would inevitably invalidate our formal safety guarantees. Moreover, because the exact measurement of the system state is unavailable, we propose an output feedback MPC approach, which exploits noisy measurements of the system. Based on these measurements, we use a simple linear state observer to estimate the inaccessible state of the system. Finally, we consider a vehicle platooning system to demonstrate the effectiveness of our real-time robust output feedback MPC approach.

[3] Google Scholar provides more than 7 million results for “model predictive control” (accessed: 20 April 2023).

In Chapter 5, we incorporate our safe sets along with corresponding safety-preserving controllers into minimally invasive safety filters, also known as supervisory control. Such filters aim to modify the desired input of a high-performance controller in a minimally invasive way so that safety is always guaranteed. Thus, safety filters serve as supervisory mediators between a simple, safety-preserving backup controller and a sophisticated, unverified high-performance controller, which is obtained, e.g., using machine learning techniques. Based on a finite set of training data, we first perform offline set membership identification to identify models that are conformant to the data. Because we make no assumptions about the availability of a model along with its corresponding disturbance set, a new measurement obtained online might invalidate the model conformance. Thus, we quickly update the model, the safety-preserving backup controller, and the safe set online to restore formal safety guarantees. Finally, we demonstrate the usefulness and scalability of our safety filter approach by considering multiple numerical examples from the literature.

In Chapter 6, we perform online safety verification of autonomous vehicles while considering the uniqueness of each traffic situation. A challenging aspect of online safety verification is the varying number of surrounding traffic participants, which causes significant variations in computational demand. To guarantee timely, safe trajectories of the controlled autonomous vehicle, we propose an anytime approach that quickly provides conservative formal verification results and continually refines them until the available computation time is elapsed. Thus, our anytime algorithm can be interrupted at any time while optimally using the available computational resources. Moreover, if the safety of the desired trajectory cannot be verified in time, the safety-preserving backup controller overwrites the desired control inputs. Finally, we consider multiple traffic scenario benchmarks to demonstrate the effectiveness of our anytime safety verification approach.

In Chapter 7, we conclude this thesis. In addition, we provide suggestions for promising future research directions.

2 Preliminaries

In this chapter, we present important mathematical concepts and overview the software toolboxes used throughout this thesis. First, we introduce the class of convex optimization problems (COPs) in Section 2.1. Because the constraint set of such an optimization problem must be convex, we present important representations of convex sets in Section 2.2. In Section 2.3, we give an overview of our reachability analysis that uses zonotopes as a set representation. After defining important invariant sets and presenting standard algorithms for their computation in Section 2.4, we provide an overview of the used software toolboxes in Section 2.5.

2.1 Convex Optimization

How quickly an optimization problem can be solved depends on many factors, such as the number of optimization variables and constraints and the exploitable structure of the problem. It has long been recognized that "the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity" [53]. In contrast to most nonconvex optimization algorithms, which typically suffer from an exponential computational complexity with respect to the problem dimension, there exist algorithms for many COPs that have a polynomial computational complexity [54–56]. In this thesis, we call an algorithm, method, procedure, or approach "efficient" if its computational complexity is polynomial with respect to the problem dimension, i.e., if it is scalable.

The efficient COP algorithms allow us to quickly and reliably solve large COPs arising in a wide variety of applications, such as portfolio optimization, statistical estimation, and data fitting [55]. Regarding applications in control, it was claimed that a "tuned convex optimization control policy is the proportional-integral-derivative (PID) controller of the 21^{st} century" [57].

To define the important class of COPs, we first introduce convex sets and convex functions [58].

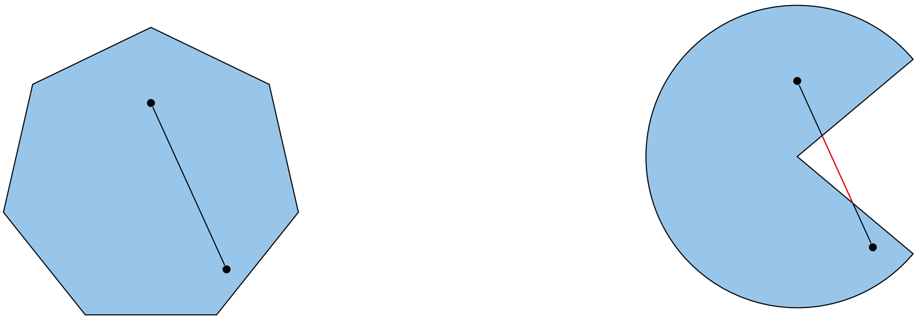
Definition 2.1 (Convex Set): A set $\mathcal{S} \subset \mathbb{R}^n$ is convex if

$$\alpha s_1 + (1-\alpha) s_2 \in \mathcal{S}$$

for any $s_1, s_2 \in \mathcal{S}$ and any $\alpha \in [0,1]$, i.e., if the line segment between any s_1 and s_2 lies within $\mathcal{S}$. ■

Examples of convex and nonconvex sets are shown in Fig. 2.1. Important convex sets are the empty set $\emptyset$, the origin $\{\mathbf{0}\}$, and the Euclidean space $\mathbb{R}^n$. Moreover, operations that preserve the convexity of sets are, e.g., intersections and affine functions. Thus, we also say that convex sets are closed under intersections and affine functions.

To define convex functions, we first introduce the domain $\texttt{domain}(f) \subseteq \mathbb{R}^n$ of a real-valued function $f : \mathbb{R}^n \to \mathbb{R}$ as the subset of $\mathbb{R}^n$ for which f is defined. For instance, the domain of the



(a) Convex set.

(b) Nonconvex set.

Figure 2.1: Convex and nonconvex sets.

logarithmic function $\log : \mathbb{R} \to \mathbb{R}$ is $\texttt{domain}\,(\log) = \mathbb{R}_{>0}$. After defining the domain of functions, we introduce convex functions in the following definition.

Definition 2.2 (Convex Function): A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if $\texttt{domain}\,(f) \subseteq \mathbb{R}^n$ is a convex set and if

$$f(\alpha s_1 + (1-\alpha)s_2) \leq \alpha f(s_1) + (1-\alpha) f(s_2)$$

for any $s_1, s_2 \in \texttt{domain}\,(f)$ and any $\alpha \in [0,1]$, i.e., if the line segment between any $(s_1, f(s_1))$ and $(s_2, f(s_2))$ lies on or above the graph of f. A function f is concave if $-f$ is convex. ■

Examples of convex and nonconvex functions are shown in Fig. 2.2. Often used convex functions of one or more variables with their corresponding domains are, e.g., the absolute value ($\mathbb{R}$), the maximum ($\mathbb{R}^n$), the cubic function ($\mathbb{R}_{\geq 0}$), the p-norm ($\mathbb{R}^n$), and the Frobenius norm ($\mathbb{R}^{n \times n}$). Conversely, often used concave functions are, e.g., the minimum ($\mathbb{R}^n$), the square root ($\mathbb{R}_{\geq 0}$), the logarithm ($\mathbb{R}_{>0}$), and the geometric mean $\left(\mathbb{R}^n_{>0}\right)$. Moreover, affine functions are the only functions that are not only convex but also concave. Similar to convex sets, a variety of operations also exist that preserve the convexity of functions, e.g., nonnegative weighted sums, compositions with an affine mapping, and the elementwise maximum [55].

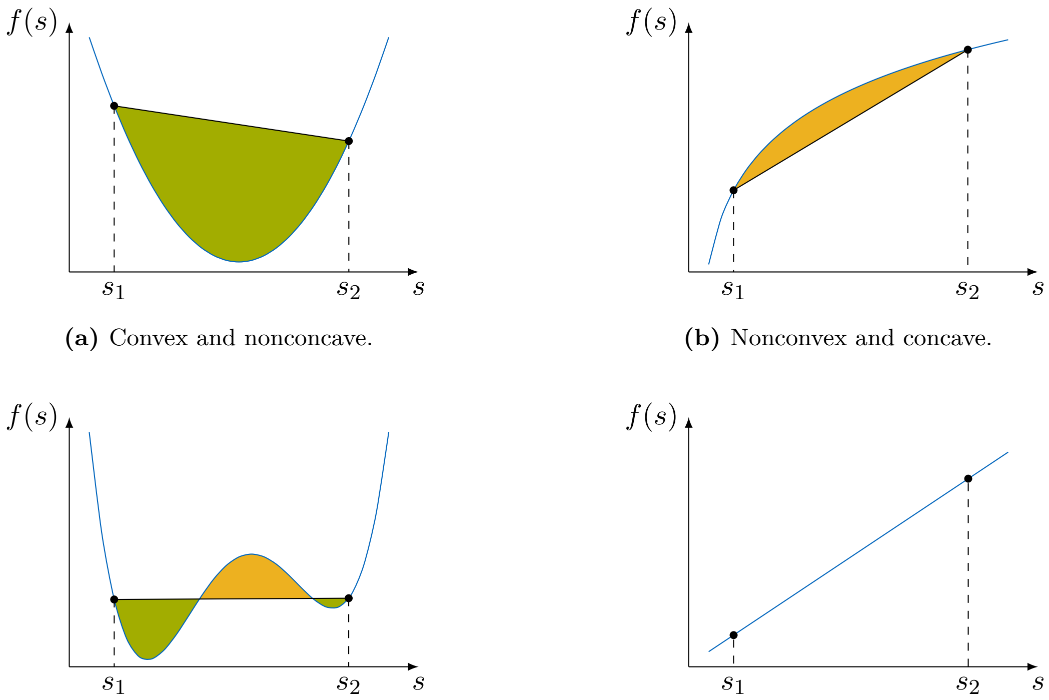
After introducing convex sets and functions, we finally define the important class of COPs, which includes least-squares and linear programming problems [59, 60].

Definition 2.3 (Convex Optimization Problem): An optimization problem

$$\begin{aligned} \underset{s}{\text{minimize}} \quad & f(s) \\ \text{subject to} \quad & s \in \mathcal{S} \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \underset{s}{\text{maximize}} \quad & -f(s) \\ \text{subject to} \quad & s \in \mathcal{S} \end{aligned}$$

is convex if $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathcal{S} \subset \mathbb{R}^n$ are convex. Moreover, this optimization problem is feasible if $\texttt{domain}\,(f) \cap \mathcal{S}$ is nonempty; otherwise, it is infeasible. In addition, $s^\star = \inf\{f(s) \mid s \in \mathcal{S}\}$ is called optimal or the solution of this optimization problem. ■

Because of the convexity, every local optimum of a COP is also a global optimum [61, Thm. 3.4.2]. This fundamental property is exploited by efficient convex optimization algorithms, such as the interior-point, subgradient, and ellipsoid methods [55, 56]. These algorithms are



(a) Convex and nonconcave.

(b) Nonconvex and concave.

(c) Nonconvex and nonconcave.

(d) Convex and concave.

Figure 2.2: Convex and nonconvex functions.

implemented in both freeware [62,63] and commercial solvers [64,65] for certain classes of COPs, e.g., linear, quadratic, second-order cone, and semidefinite programs. To invoke one of these solvers, the COP must be reformulated appropriately to match the solver-specific interfaces, i.e., transformed into an equivalent COP in standard form [55, Ch. 4]. Instead of reformulating COPs by hand, convex optimization modeling frameworks offer a convenient way to do this transformation procedure automatically [66–68].

To ensure the scalability of our control approaches in this thesis, we exclusively solve COPs to obtain optimal control inputs. Because the constraint set of any COP must be convex, as stated in Definition 2.3, we introduce important representations of convex sets in the following section.

2.2 Convex Sets

In this section, we introduce important representations of convex sets that are compact, i.e., closed and bounded. In addition, we present exact and over-approximative conversions between the convex set representations. Finally, we introduce corresponding set operations and present two approaches to solve the zonotope containment problem.

2.2.1 Representations

To efficiently perform operations on convex sets, choosing a suitable set representation is crucial. Subsequently, we introduce ellipsoids [69], polytopes [70, 71], zonotopes [72], and multidimensional intervals as valuable representations of closed, bounded, convex sets.

An ellipsoid can be seen as an affine transformation of a hypersphere and is introduced in the following definition.

Definition 2.4 (Ellipsoid): An ellipsoid $\mathcal{E} \subset \mathbb{R}^n$ is defined by

$$\begin{aligned}\mathcal{E} &= \left\{s \in \mathbb{R}^n \;\middle|\; (s-c)^T S^{-1}(s-c) \leq 1\right\} \\ &= \left\{s \in \mathbb{R}^n \;\middle|\; s = c + S^{1/2}b, \|b\|_2 \leq 1\right\},\end{aligned}$$

where $c \in \mathbb{R}^n$ is the center, $S \in \mathbb{R}^{n \times n}$ is the symmetric positive definite shape matrix, and the symmetric positive definite matrix $S^{1/2}$ denotes the root of S [39, Sec. 3.2], i.e., $(S^{1/2})^2 = S$. We use $\mathcal{E} = \langle c, S \rangle_E$ to obtain a more concise notation for representing an ellipsoid. ■

A polytope can be seen as the intersection of half-spaces and is introduced in the following definition.

Definition 2.5 (Polytope): A polytope $\mathcal{P} \subset \mathbb{R}^n$ in half-space representation is defined by

$$\mathcal{P} = \{s \in \mathbb{R}^n \mid Hs \leq h\},$$

where $H \in \mathbb{R}^{m \times n}$ and $h \in \mathbb{R}^m$ are the data representing the half-spaces, and $m \in \mathbb{N}_{>0}$ is the number of these half-spaces. We use $\mathcal{P} = \langle H, h \rangle_P$ to obtain a more concise notation for representing a polytope. ■

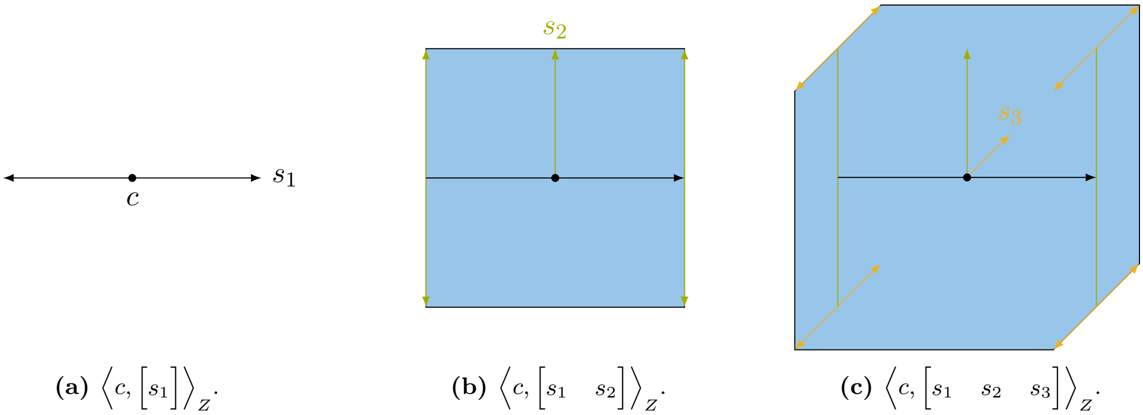
Polytopes can alternatively be expressed as the convex hull of a finite set of points, also known as the vertex representation. As the number of vertices for representing a general set grows exponentially with respect to the dimension, we restrict our attention to the half-space representation when proposing scalable control approaches. A special case of a polytope is a zonotope, which is centrally symmetric and introduced in the following definition.

Definition 2.6 (Zonotope): A zonotope $\mathcal{Z} \subset \mathbb{R}^n$ in generator representation is defined by

$$\mathcal{Z} = \{s \in \mathbb{R}^n \mid s = c + G\lambda, |\lambda| \leq \mathbf{1}\},$$

where $c \in \mathbb{R}^n$ is the center, $G \in \mathbb{R}^{n \times \texttt{gen}(\mathcal{Z})}$ is the generator matrix with $\texttt{gen}(\mathcal{Z}) \in \mathbb{N}$ denoting the number of generators, and $\lambda \in \mathbb{R}^{\texttt{gen}(\mathcal{Z})}$ is the parameter vector. The order of $\mathcal{Z}$ is $\texttt{order}(\mathcal{Z}) = \frac{\texttt{gen}(\mathcal{Z})}{n}$. We use $\mathcal{Z} = \langle c, G \rangle_Z$ to obtain a more concise notation for representing a zonotope. ■

Zonotopes can be constructed by the Minkowski addition of line segments, as illustrated in Fig. 2.3. Alternatively, zonotopes can also be seen as an affine transformation of a cube.



(a) $\left\langle c, \begin{bmatrix} s_1 \end{bmatrix} \right\rangle_Z$. **(b)** $\left\langle c, \begin{bmatrix} s_1 & s_2 \end{bmatrix} \right\rangle_Z$. **(c)** $\left\langle c, \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix} \right\rangle_Z$.

Figure 2.3: Step-by-step construction of the zonotope $\left\langle c, \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix} \right\rangle_Z$ in $\mathbb{R}^2$. By adding more generators to the generator matrix of the zonotope, the area increases.

For $\mathcal{Z} = \langle c, G \rangle_Z \subset \mathbb{R}^n$ and any point $s \in \mathcal{Z}$, the corresponding parameter vector $\lambda \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z})}$ with $|\lambda| \leq \mathbf{1}$ can be obtained by solving the COP

$$\underset{\lambda}{\text{minimize}} \quad J_\lambda(\lambda) \tag{2.1a}$$

$$\text{subject to} \quad s = c + G\lambda \tag{2.1b}$$

$$|\lambda| \leq \mathbf{1}, \tag{2.1c}$$

where J_λ is a convex cost function, e.g., $\|\lambda\|_2$ or 0. Thus, λ is not necessarily unique for parameterizing any s unless G is invertible. In this special case, the zonotope is called a parallelotope, its order is 1, and the unique parameter vector is

$$\lambda = G^{-1}(s - c). \tag{2.2}$$

Moreover, a parallelotope with a diagonal generator matrix is called a multidimensional interval, also known as an axis-aligned box, hyperrectangle, and orthotope. For these multidimensional intervals, (2.2) can be efficiently computed because the inverse of an invertible diagonal matrix is obtained by replacing each element on the diagonal with its reciprocal. In addition, multidimensional intervals can be equivalently represented by their lower and upper bounds, as introduced in the following definition.

Definition 2.7 (Multidimensional Interval): A multidimensional interval $\mathcal{I} \subset \mathbb{R}^n$ in interval representation is defined by

$$\mathcal{I} = \left\{ s \in \mathbb{R}^n \;\middle|\; \underline{\mathcal{I}} \leq s \leq \overline{\mathcal{I}} \right\},$$

where $\underline{\mathcal{I}} \in \mathbb{R}^n$ and $\overline{\mathcal{I}} \in \mathbb{R}^n$ denote the lower and upper bound of $\mathcal{I}$, respectively. We use $\mathcal{I} = \left[\underline{\mathcal{I}}, \overline{\mathcal{I}}\right]$ to obtain a more concise notation for representing a multidimensional interval, and

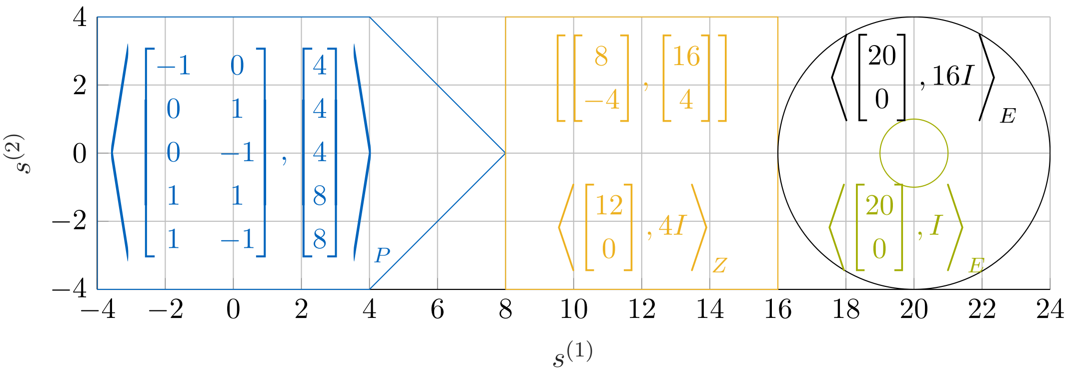


Figure 2.4: Convex set representations in $\mathbb{R}^2$. In particular, a polytope, a multidimensional interval that a zonotope can also represent, and two ellipsoids are shown.

define

$$\texttt{min}\,(\mathcal{I}) = \underline{\mathcal{I}} \tag{2.3a}$$

$$\texttt{max}\,(\mathcal{I}) = \overline{\mathcal{I}} \tag{2.3b}$$

$$\texttt{center}\,(\mathcal{I}) = 0.5\left(\overline{\mathcal{I}} + \underline{\mathcal{I}}\right) \tag{2.3c}$$

$$\texttt{radius}\,(\mathcal{I}) = 0.5\left(\overline{\mathcal{I}} - \underline{\mathcal{I}}\right). \tag{2.3d}$$

as the minimum, maximum, center, and radius of $\mathcal{I}$, respectively. ■

Multidimensional intervals can be seen as the Cartesian product of multiple intervals or as vectors whose elements are intervals. A straightforward generalization of these multidimensional intervals is interval matrices [73], as introduced in the following definition.

Definition 2.8 (Interval Matrix): A square interval matrix $\boldsymbol{\mathcal{M}} \subset \mathbb{R}^{n\times n}$ is defined by

$$\boldsymbol{\mathcal{M}} = \left\{ M \in \mathbb{R}^{n\times n} \;\middle|\; \underline{\boldsymbol{\mathcal{M}}} \leq M \leq \overline{\boldsymbol{\mathcal{M}}} \right\},$$

where $\underline{\boldsymbol{\mathcal{M}}} \in \mathbb{R}^{n\times n}$ and $\overline{\boldsymbol{\mathcal{M}}} \in \mathbb{R}^{n\times n}$ denote the lower and upper bound of $\boldsymbol{\mathcal{M}}$, respectively. We use $\boldsymbol{\mathcal{M}} = \left[\underline{\boldsymbol{\mathcal{M}}}, \overline{\boldsymbol{\mathcal{M}}}\right]$ to obtain a more concise notation for representing an interval matrix, and define the minimum, maximum, center, and radius of $\boldsymbol{\mathcal{M}}$ analogously to (2.3). ■

Finally, we visualize ellipsoids, polytopes, zonotopes, and multidimensional intervals in Fig. 2.4. After defining these important convex set representations, we consider the exact and over-approximative conversion between some.

2.2.2 Conversions

Because multidimensional intervals are special cases of zonotopes, they can be equivalently expressed in both generator and interval representation. For instance, both $[-\mathbf{1}, \mathbf{1}] \subset \mathbb{R}^2$ and $\langle \mathbf{0}, I \rangle_Z \subset \mathbb{R}^2$ represent the unit ball corresponding to the infinity norm in $\mathbb{R}^2$, as shown in Fig. 1 on page xviii. A conversion from the generator representation $\langle c_{\mathcal{I}}, G_{\mathcal{I}} \rangle_Z$ of the multidimensional interval $\mathcal{I} \subset \mathbb{R}^n$ to its interval representation $\left[\underline{\mathcal{I}}, \overline{\mathcal{I}}\right]$ is achieved by

$$\underline{\mathcal{I}} = c_{\mathcal{I}} - \texttt{diag}\left(|G_{\mathcal{I}}|\right) \tag{2.4a}$$

$$\overline{\mathcal{I}} = c_{\mathcal{I}} + \texttt{diag}\left(|G_{\mathcal{I}}|\right) \tag{2.4b}$$

and vice versa by

$$c_{\mathcal{I}} = \texttt{center}\left(\mathcal{I}\right) \tag{2.5a}$$

$$G_{\mathcal{I}} = \texttt{diag}\left(\texttt{radius}\left(\mathcal{I}\right)\right). \tag{2.5b}$$

Similarly, a conversion from the interval representation $\left[\underline{\mathcal{I}}, \overline{\mathcal{I}}\right]$ to its half-space representation $\langle H_{\mathcal{I}}, h_{\mathcal{I}} \rangle_P$ is achieved by

$$H_{\mathcal{I}} = \begin{bmatrix} I \\ -I \end{bmatrix} \tag{2.6a}$$

$$h_{\mathcal{I}} = \begin{bmatrix} \overline{\mathcal{I}} \\ -\underline{\mathcal{I}} \end{bmatrix}. \tag{2.6b}$$

The conversions in (2.4) to (2.6) follow directly from Definitions 2.5 to 2.7. Converting multidimensional intervals to the generator and half-space representation is particularly beneficial because multidimensional intervals generally represent most constraint sets. Thus, under-approximations of the constraint sets are required when choosing ellipsoids as set representations.

Because zonotopes are special cases of polytopes, they can be equivalently expressed in both half-space and generator representations. For instance, both $\langle \mathbf{0}, I \rangle_Z \subset \mathbb{R}^2$ and $\left\langle \begin{bmatrix} I & -I \end{bmatrix}^T, \mathbf{1} \right\rangle_P \subset \mathbb{R}^2$ represent the unit ball corresponding to the infinity norm in $\mathbb{R}^2$, as shown in Fig. 1 on page xviii. However, the half-space conversion of general zonotopes is combinatorially complex with respect to the order of the zonotope [74, 75]. Conversely, the interval conversion of a zonotope can be efficiently performed in an over-approximative way. According to [76, Prop. 2.2] and (2.4), the smallest multidimensional interval enclosure of the zonotope $\mathcal{Z} = \langle c, G \rangle_Z \subset \mathbb{R}^n$ is

$$\begin{aligned} \texttt{interval}\left(\mathcal{Z}\right) &= \langle c, \texttt{diag}(|G|\,\mathbf{1}) \rangle_Z \\ &= \left[c - |G|\,\mathbf{1}, c + |G|\,\mathbf{1}\right], \end{aligned} \tag{2.7}$$

which is also known as an axis-aligned bounding box. Similarly, the smallest multidimensional interval enclosure of the polytope $\mathcal{P} = \langle H, h \rangle_P \subset \mathbb{R}^n$ can be computed by solving $2n$ linear programming problems [59, 60, 77], e.g., the i^{th} element of the lower bound is obtained by solving

$$\begin{aligned} \underset{s}{\text{minimize}} \quad & s^{(i)} \\ \text{subject to} \quad & Hs \le h, \end{aligned}$$

where $i \in \mathbb{N}_{[1,n]}$. We also want to mention that the error when approximating the unit ball corresponding to the Euclidean norm $\langle \mathbf{0}, I \rangle_E \subset \mathbb{R}^n$ by a zonotope can be made arbitrarily small by increasing the zonotope order [78].

2.2.3 Operations

After introducing suitable convex set representations and conversions between them, we present some important set operations. For a matrix $M \in \mathbb{R}^{m\times n_{1,2}}$ and three closed, bounded, convex sets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^{n_{1,2}}$ and $\mathcal{S}_3 \subset \mathbb{R}^{n_3}$, we define the following set operations:

- Matrix multiplication: $M\mathcal{S}_1 = \{Ms_1 \mid s_1 \in \mathcal{S}_1\}$ (2.8a)
- Set-based multiplication: $\mathcal{S}_1\mathcal{S}_2 = \{s_1 s_2 \mid s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2\}$ (2.8b)
- Minkowski addition: $\mathcal{S}_1 \oplus \mathcal{S}_2 = \{s_1 + s_2 \mid s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2\}$ (2.8c)
- Minkowski difference: $\mathcal{S}_1 \ominus \mathcal{S}_2 = \{s \mid s \oplus \mathcal{S}_2 \subseteq \mathcal{S}_1\}$ (2.8d)
- Cartesian product: $\mathcal{S}_1 \times \mathcal{S}_3 = \left\{ \begin{bmatrix} s_1 \\ s_3 \end{bmatrix} \middle| \; s_1 \in \mathcal{S}_1, s_3 \in \mathcal{S}_3 \right\}$ (2.8e)
- Convex hull: $\texttt{conv}\,(\mathcal{S}_1, \mathcal{S}_2) = \{\alpha s_1 + (1-\alpha)s_2 \mid s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2, \alpha \in [0,1]\}$ (2.8f)
- Directed Hausdorff distance: $\texttt{dist}\,(\mathcal{S}_1, \mathcal{S}_2) = \min\{\delta \in \mathbb{R}_{\geq 0} \mid \mathcal{S}_1 \subseteq \mathcal{S}_2 \oplus \delta \langle \mathbf{0}, I\rangle_Z\}$ (2.8g)

Based on the definition of the directed Hausdorff distance in (2.8g) [79], it follows that $\mathcal{S}_1 \subseteq \mathcal{S}_2$ if and only if $\texttt{dist}\,(\mathcal{S}_1, \mathcal{S}_2) = 0$. Instead of using the unit ball corresponding to the infinity norm in (2.8g), other norms can also be used [79]. Moreover, to account for different orders of magnitude of the dimensions, they can also be weighted accordingly, e.g., using a suitable diagonal matrix instead of I in (2.8g).

Open-source software toolboxes are available to efficiently and accurately operate on, e.g., ellipsoids [80], polytopes [77], and zonotopes [81]. Subsequently, we present the computations of some set operations introduced in (2.8) on different convex set representations and comment on their corresponding computational complexity.

As we will see in Section 2.3, the two most critical set operations for our reachability analysis are the matrix multiplication and the Minkowski addition. According to [80], the multiplication of the ellipsoid $\langle c, S\rangle_E \subset \mathbb{R}^n$ by a matrix $M \in \mathbb{R}^{m\times n}$ is computed by

$$M\,\langle c, S\rangle_E = \left\langle Mc, MSM^T \right\rangle_E,$$

i.e., ellipsoids are closed under linear transformations. However, ellipsoids are not closed under Minkowski additions, which results in an over-approximation error when performing this operation.

In contrast to ellipsoids, polytopes are closed under both set operations. According to [77], the multiplication of the polytope $\langle H, h\rangle_P \subset \mathbb{R}^n$ with $n \in \mathbb{N}_{>0}$ by an invertible matrix $M \in \mathbb{R}^{n\times n}$ is computed by

$$M\,\langle H, h\rangle_P = \left\langle HM^{-1}, h \right\rangle_P,$$

i.e., this set operation has a polynomial computational complexity with respect to n [82]. However, performing the Minkowski sum of two general polytopes suffers from an exponential computational complexity [25, 83], which makes polytopes an unsuitable set representation for performing reachability analysis of large-scale systems. Nevertheless, polytopes are typically used for representing state and input constraint sets. Thus, we additionally introduce some other important set operations on polytopes subsequently. The Chebyshev center $\texttt{center}(\mathcal{P}) \in \mathbb{R}^n$ of the polytope $\mathcal{P} \subset \mathbb{R}^n$ is the center of the largest Euclidean ball that lies in $\mathcal{P}$ [55, Sec. 8.5.1], which can be efficiently determined by solving the COP

$$\underset{c,r}{\text{maximize}} \quad r \tag{2.9a}$$

$$\text{subject to} \quad \langle c, rI \rangle_E \subseteq \mathcal{P}. \tag{2.9b}$$

Thus, $\mathcal{P}$ is a lower-dimensional polytope if the radius of the Euclidean ball obtained by solving (2.9) is zero. Moreover, the Minkowski addition of $\langle H, h \rangle_P$ and a vector $s \in \mathbb{R}^n$, and the multiplication of $\langle H, h \rangle_P$ by a scalar $\alpha \in \mathbb{R}_{>0}$ are computed by

$$\begin{aligned} \{s\} \oplus \langle H, h \rangle_P &= \langle H, h + Hs \rangle_P \\ \alpha \langle H, h \rangle_P &= \langle H, \alpha h \rangle_P \, , \end{aligned}$$

which follows directly from Definition 2.5. Thus, scaling $\mathcal{P} = \langle H, h \rangle_P$ by α with respect to any $s \in \mathcal{P}$ is achieved by the function

$$\begin{aligned} \texttt{scalePolytope}(\mathcal{P}, \alpha, s) &= \alpha\big(\mathcal{P} \oplus \{-s\}\big) \oplus \{s\} \\ &= \langle H, \alpha(h - Hs) + Hs \rangle_P \, . \end{aligned}$$

In addition to polytopes, zonotopes are closed under matrix multiplications and Minkowski additions. According to [72], the Minkowski addition of two zonotopes $\langle c_1, G_1 \rangle_Z \subset \mathbb{R}^n$ and $\langle c_2, G_2 \rangle_Z \subset \mathbb{R}^n$, and the multiplication by a matrix $M \in \mathbb{R}^{m \times n}$ with $m, n \in \mathbb{N}_{>0}$ are computed by

$$\langle c_1, G_1 \rangle_Z \oplus \langle c_2, G_2 \rangle_Z = \left\langle c_1 + c_2, \begin{bmatrix} G_1 & G_2 \end{bmatrix} \right\rangle_Z \tag{2.10a}$$

$$M \langle c_1, G_1 \rangle_Z = \langle M c_1, M G_1 \rangle_Z \, . \tag{2.10b}$$

Because the computational complexity of these two crucial set operations is polynomial with respect to m and n [16], zonotopes are well suited as set representations for our efficient reachability analysis. Based on (2.10a), performing Minkowski additions increases the order of the resulting zonotope. To limit the storage space requirements, tight over-approximative zonotope order reduction techniques exist [84,85]. For a given zonotope $\mathcal{Z} \subset \mathbb{R}^n$ and a scalar $\alpha \in \mathbb{N}_{[1, \lceil \texttt{order}(\mathcal{Z}) \rceil - 1]}$, these methods compute a reduced order zonotope $\mathcal{Z}_{\text{red}} = \texttt{reduce}(\mathcal{Z}, \alpha)$ such that $\mathcal{Z} \subseteq \mathcal{Z}_{\text{red}}$ and $\texttt{order}(\mathcal{Z}_{\text{red}}) = \alpha$. Subsequently, we introduce more set operations on zonotopes used throughout this thesis.

Based on Definition 2.6, the Cartesian product of $\langle c_1, G_1 \rangle_Z \subset \mathbb{R}^n$ and $\langle c_2, G_2 \rangle_Z \subset \mathbb{R}^m$ is computed by

$$\langle c_1, G_1 \rangle_Z \times \langle c_2, G_2 \rangle_Z = \left\langle \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}, \begin{bmatrix} G_1 & \mathbf{0} \\ \mathbf{0} & G_2 \end{bmatrix} \right\rangle_Z ,$$

i.e., zonotopes are also closed under Cartesian products. In addition, we define the stacking of $\langle c_1, G_1 \rangle_Z$ and $\langle c_2, G_2 \rangle_Z$ by

$$\left\langle \begin{matrix} \langle c_1, G_1 \rangle_Z \\ \langle c_2, G_2 \rangle_Z \end{matrix} \right\rangle_Z = \left\langle \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}, \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \right\rangle_Z , \tag{2.11}$$

where the number of generators of both zonotopes must be equal. Generally, the convex hull of two zonotopes is not a zonotope, and finding a tight enclosing zonotope is a complex task [86]. According to [87], an over-approximation of the convex hull $\texttt{conv}\left(\mathcal{Z}_1, \mathcal{Z}_2\right)$ of $\mathcal{Z}_1 = \langle c_1, G_1\rangle_Z \subset \mathbb{R}^n$ and $\mathcal{Z}_2 = \langle c_2, G_2\rangle_Z \subset \mathbb{R}^n$ with $\texttt{gen}\left(\mathcal{Z}_1\right) = \texttt{gen}\left(\mathcal{Z}_2\right)$ is

$$\texttt{conv}_{\text{over}}\left(\langle c_1, G_1\rangle_Z, \langle c_2, G_2\rangle_Z\right) = 0.5\left\langle c_1 + c_2, \begin{bmatrix} G_1 + G_2 & c_1 - c_2 & G_1 - G_2 \end{bmatrix}\right\rangle_Z. \tag{2.12}$$

Typically, the over-approximation in (2.12) is reasonably tight if $\mathcal{Z}_2$ is obtained by multiplying $\mathcal{Z}_1$ with a matrix whose eigenvalues are close to 1. According to [76, Thm. 3.3], the set-based multiplication $\boldsymbol{\mathcal{M}}\mathcal{Z}$ of an interval matrix $\boldsymbol{\mathcal{M}} = \left[\underline{\boldsymbol{\mathcal{M}}}, \overline{\boldsymbol{\mathcal{M}}}\right] \subset \mathbb{R}^{n\times n}$ and a zonotope $\mathcal{Z} = \langle c, G\rangle_Z \subset \mathbb{R}^n$ can be tightly over-approximated by

$$\boldsymbol{\mathcal{M}} \otimes_{\text{over}} \langle c, G\rangle_Z = \left\langle Cc, \begin{bmatrix} CG & \texttt{diag}\left(R\left|\begin{bmatrix} c & G\end{bmatrix}\right|\mathbf{1}\right)\end{bmatrix}\right\rangle_Z, \tag{2.13}$$

where $C = \texttt{center}\left(\boldsymbol{\mathcal{M}}\right) \in \mathbb{R}^{n\times n}$ is the center of $\boldsymbol{\mathcal{M}}$ and $R = \texttt{radius}\left(\boldsymbol{\mathcal{M}}\right) \in \mathbb{R}^{n\times n}_{\geq 0}$ is the radius of $\boldsymbol{\mathcal{M}}$. To visualize some of the presented set operations on two zonotopes in $\mathbb{R}^2$, we present Example 2.9.

Example 2.9 (Set Operations on Zonotopes): In Fig. 2.5, we illustrate some set operations performed on the two zonotopes

$$\mathcal{Z}_1 = \left\langle \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 & 3 & 1 \\ 3 & 0 & 1 \end{bmatrix}\right\rangle_Z$$
$$\mathcal{Z}_2 = \left\langle \begin{bmatrix} 20 \\ 6 \end{bmatrix}, \begin{bmatrix} 3 & 3 & 1 \\ 0 & -3 & 1 \end{bmatrix}\right\rangle_Z$$

using the matrices

$$M = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.4 \end{bmatrix}$$
$$\boldsymbol{\mathcal{M}} = \begin{bmatrix} [0.8, 1.2] & [-1.2, -0.8] \\ [-1.2, -0.8] & [-0.2, 0.2] \end{bmatrix}.$$

To compute the parallelotope $\texttt{reduce}\left(\mathcal{Z}_1, 1\right)$, we use the principle component analysis zonotope order reduction method [84]. ■

2.2.4 Zonotope Containment

After introducing all set operations that are needed for later computations, we present two approaches for determining if a zonotope $\mathcal{Z}_1 \subset \mathbb{R}^n$ is contained or included within another zonotope $\mathcal{Z}_2 \subset \mathbb{R}^n$, which is co-NP-complete [75]. The first zonotope containment approach transforms $\mathcal{Z}_2$ from generator to half-space representation [74], which is usually a computationally complex task for high-order zonotopes [75]. According to [88], $\mathcal{Z}_1 = \langle c_1, G_1\rangle_Z$ is contained in $\mathcal{Z}_2 = \langle H_2, h_2\rangle_P$ if and only if

$$H_2 c_1 + \left|H_2 G_1\right| \mathbf{1} \leq h_2. \tag{2.14}$$

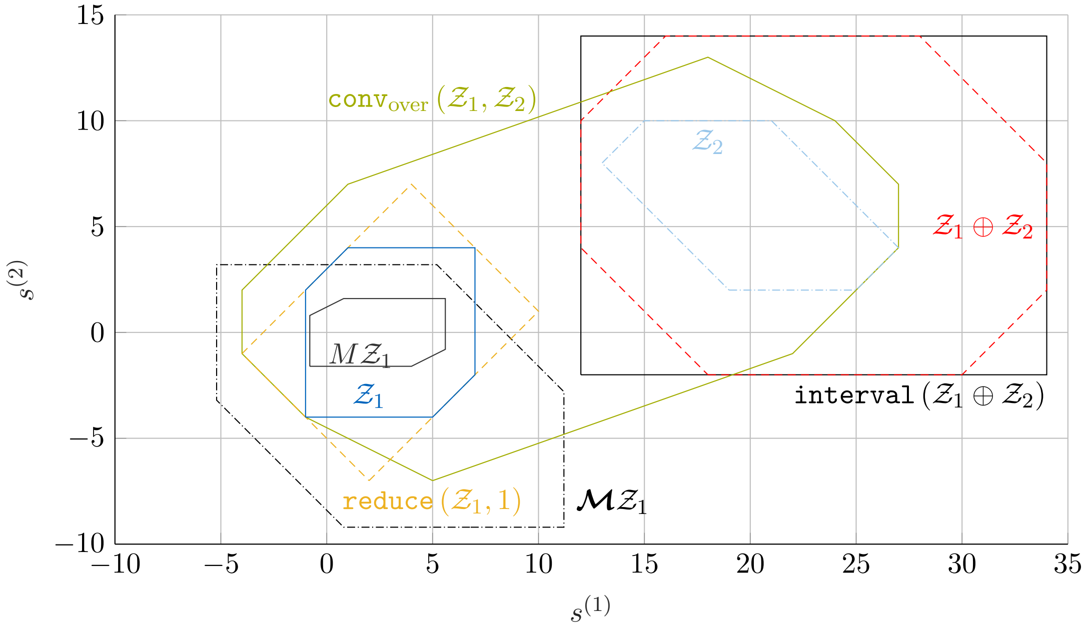


Figure 2.5: Set operations performed on the two zonotopes $\mathcal{Z}_1, \mathcal{Z}_2 \subset \mathbb{R}^2$ that are defined in Example 2.9.

In the special case of $\mathcal{Z}_2$ being expressed in interval representation $\mathcal{Z}_2 = \left[\underline{\mathcal{Z}_2}, \overline{\mathcal{Z}_2}\right]$, the condition in (2.14) simplifies to

$$\begin{aligned} \underline{\mathcal{Z}_2} &\leq c_1 - |G_1|\,\mathbf{1} \\ \overline{\mathcal{Z}_2} &\geq c_1 + |G_1|\,\mathbf{1}. \end{aligned}$$

The second zonotope containment approach directly solves a linear feasibility problem, allowing us to efficiently incorporate such constraints into COPs. According to [52, 79, 89], $\mathcal{Z}_1 = \langle c_1, G_1 \rangle_Z$ is contained in $\mathcal{Z}_2 = \langle c_2, G_2 \texttt{diag}(s) \rangle_Z$ with arbitrary scaling factor $s \in \mathbb{R}_{>0}^{\texttt{gen}(\mathcal{Z}_2)}$ if a matrix $\Gamma \in \mathbb{R}^{\texttt{gen}(\mathcal{Z}_2) \times \texttt{gen}(\mathcal{Z}_1)}$ and a vector $\gamma \in \mathbb{R}^{\texttt{gen}(\mathcal{Z}_2)}$ exist such that

$$G_1 = G_2 \Gamma \tag{2.15a}$$

$$c_2 - c_1 = G_2 \gamma \tag{2.15b}$$

$$\left|\begin{bmatrix} \Gamma & \gamma \end{bmatrix}\right| \mathbf{1} \leq s. \tag{2.15c}$$

In contrast to (2.15), (2.14) is a necessary and sufficient condition for zonotope containment. Nevertheless, (2.15) can be solved for comparably higher-order zonotopes by using efficient convex optimization algorithms without involving the computationally demanding half-space conversion.

2.3 Reachability Analysis

Because zonotopes are concise set representations and essential operations are performed efficiently and accurately, they are becoming increasingly popular in various applications.

For instance, zonotopes are used in, e.g., state estimation [90–93], hybrid systems verification [74, 94], robust control [50, 95, 96], and collision detection [86]. Similar to ellipsoids [97, 98] and polytopes [99, 100], zonotopes are also widely used as set representation in reachability analysis [27, 72, 87], as presented subsequently.

In this thesis, we consider continuous-time (CT), time-invariant systems that evolve according to

$$\dot{x}(t) = f\left(x(t), u(t), w(t)\right), \tag{2.16}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the system state, $u(t) \in \mathbb{R}^{n_u}$ is the control input, and $w(t) \in \mathbb{R}^{n_w}$ is the unknown disturbance at time $t \in \mathbb{R}_{\geq 0}$. The disturbance trajectory $w(\cdot)$ is unknown but bounded by the disturbance set $\mathcal{W} \subset \mathbb{R}^{n_w}$, i.e., $w(t) \in \mathcal{W}$ at all times t. To obtain a more concise notation for representing such constraints, we use $w(\cdot) \in \mathcal{W}$, i.e., we refer by $w(\cdot)$ to the whole disturbance trajectory and by $w(t)$ to the value of this trajectory at time t. We also use this concise notation for other trajectories and their values at specific points in time. In addition, we define the model $\mathbf{M} = \left(f\left(x, u, w\right), \mathcal{W}, \mathrm{CT}\right)$ for compactly representing the CT dynamics in (2.16) with corresponding disturbance set $\mathcal{W}$.

In contrast to the CT dynamics, the digital controller of our cyber-physical system provides a piecewise constant control input only at periodic sampling times $t_k = k\Delta t$ with $k \in \mathbb{N}$ and fixed sampling period $\Delta t \in \mathbb{R}_{>0}$ [101], i.e.,

$$u(t) = u(t_k) \quad \text{for } t \in [t_k, t_{k+1}). \tag{2.17}$$

Such systems, composed of a physical plant evolving in continuous time and a digital controller being implemented in discrete time, are also known as sampled-data systems [102]. To account for (2.17) in the reachability analysis, we augment the state space according to

$$\begin{bmatrix} \dot{x}(t) \\ \dot{u}(t) \end{bmatrix} = \begin{bmatrix} f\left(x(t), u(t), w(t)\right) \\ \mathbf{0} \end{bmatrix} \tag{2.18}$$

and update the piecewise constant control input at sampling times. Conversely, to project a set of augmented states onto the lower-dimensional original state and input space [103], we define the two matrices

$$\Pi_x = \begin{bmatrix} I & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n_x \times (n_x + n_u)} \tag{2.19a}$$

$$\Pi_u = \begin{bmatrix} \mathbf{0} & I \end{bmatrix} \in \mathbb{R}^{n_u \times (n_x + n_u)}. \tag{2.19b}$$

For instance, the center and generator matrix of the zonotope $\Pi_u \left\langle c, G \right\rangle_Z \subset \mathbb{R}^{n_u}$ is obtained by erasing the first n_x rows of c and G, respectively.

We denote the solution of (2.18) at time $t \in [0, \Delta t)$ by $\tilde{\chi}\big(t, \tilde{x}(0), w(\cdot)\big)$, where $\tilde{x}(0) = \begin{bmatrix} x^T(0) & u^T(0) \end{bmatrix}^T$ is the augmented initial state. Based on this solution, the set of augmented states that the system in (2.18) can reach is called the exact reachable set and is introduced in the following definition [76, Defs. 3.1 and 3.2].

Definition 2.10 (Exact Reachable Set): For the model $\mathbf{M} = \left(f\left(x, u, w\right), \mathcal{W}, \mathrm{CT}\right)$ and the augmented initial state set $\widetilde{\mathcal{Z}}_0 \subset \mathbb{R}^{n_x+n_u}$, the exact reachable set at time $t \in [0, \Delta t)$ is

$$\widetilde{\mathcal{R}}^{\mathbf{M}}_{\mathrm{exact}}\left(t, \widetilde{\mathcal{Z}}_0\right) = \left\{ \tilde{x}(t) \in \mathbb{R}^{n_x+n_u} \;\middle|\; \tilde{x}(t) = \tilde{\chi}\big(t, \tilde{x}(0), w(\cdot)\big), \tilde{x}(0) \in \widetilde{\mathcal{Z}}_0, w(\cdot) \in \mathcal{W} \right\},$$

which is also known as the robust forward-reachable set. Similarly, the exact reachable set over the time interval $[0, \Delta t)$ is defined by

$$\widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{exact}}\left([0, \Delta t), \widetilde{\mathcal{Z}}_0\right) = \bigcup_{t \in [0, \Delta t)} \widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{exact}}\left(t, \widetilde{\mathcal{Z}}_0\right),$$

i.e., it is the union of exact reachable sets at all times within $[0, \Delta t)$. ■

Because exact reachable sets cannot be obtained for general systems [104, 105], we settle for tight over-approximations $\widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{over}}\left(\cdot, \widetilde{\mathcal{Z}}_0\right) \supseteq \widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{exact}}\left(\cdot, \widetilde{\mathcal{Z}}_0\right)$ to enclose all possible system trajectories. These over-approximations can be computed efficiently and accurately when restricting the CT system in (2.16) to be linear time-invariant (LTI) and using zonotopes as set representation, as presented subsequently.

In the LTI case, the CT, time-invariant system evolves according to

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t), \tag{2.20}$$

where $A \in \mathbb{R}^{n_x \times n_x}$ is the system matrix and $B \in \mathbb{R}^{n_x \times n_u}$ is the input matrix. To compactly represent the CT dynamics in (2.20) with corresponding disturbance set $\mathcal{W} \subset \mathbb{R}^{n_x}$, we use the model $\mathbf{M} = (A, B, \mathcal{W}, \text{CT})$ as a shorthand for $\mathbf{M} = (Ax + Bu + w, \mathcal{W}, \text{CT})$ for concisely specifying LTI systems. When considering this important class of dynamical systems, the augmented system in (2.18) simplifies to

$$\begin{bmatrix} \dot{x}(t) \\ \dot{u}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\widetilde{A}} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} + \begin{bmatrix} w(t) \\ \mathbf{0} \end{bmatrix}, \tag{2.21}$$

where $\widetilde{A} \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$ is the augmented system matrix. The well-known solution of (2.21) at time $t \in [0, \Delta t)$ is

$$\tilde{\chi}\big(t, \tilde{x}(0), w(\cdot)\big) = e^{\widetilde{A}t}\tilde{x}(0) + e^{\widetilde{A}t} \int_0^t e^{-\widetilde{A}\tau} \begin{bmatrix} w(\tau) \\ \mathbf{0} \end{bmatrix} \mathrm{d}\tau,$$

where the superposition principle is exploited. Although exact reachable sets cannot be computed for general LTI systems [104, 105], tight zonotopic over-approximations can be efficiently obtained for this important class. According to [76, Sec. 3.2], the over-approximative reachable sets of (2.21) for the point in time Δt and the time interval $[0, \Delta t)$ are

$$\widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{over}}\left(\Delta t, \widetilde{\mathcal{Z}}_0\right) = e^{\widetilde{A}\Delta t}\widetilde{\mathcal{Z}}_0 \oplus \widetilde{\mathcal{W}} \tag{2.22a}$$

$$\widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{over}}\left([0, \Delta t), \widetilde{\mathcal{Z}}_0\right) = \texttt{conv}_{\text{over}}\left(\widetilde{\mathcal{Z}}_0, e^{\widetilde{A}\Delta t}\widetilde{\mathcal{Z}}_0\right) \oplus \boldsymbol{\mathcal{F}}\widetilde{\mathcal{Z}}_0 \oplus \widetilde{\mathcal{W}}, \tag{2.22b}$$

where $\widetilde{\mathcal{W}} = \langle c_{\widetilde{\mathcal{W}}}, G_{\widetilde{\mathcal{W}}} \rangle_Z \subset \mathbb{R}^{n_x+n_u}$ is the augmented reachable disturbance set [76, Thm. 3.1] and $\boldsymbol{\mathcal{F}} \subset \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$ is the correction interval matrix that accounts for the curvature of trajectories during $[0, \Delta t)$ [76, Prop. 3.1]. The computations in (2.22) are valid if $\mathbf{0} \in \mathcal{W}$, which can typically be satisfied by performing a suitable coordinate transformation. Nevertheless, an extension for incorporating $\mathbf{0} \notin \mathcal{W}$ also exists [76, Sec. 3.2]. Because zonotopes are not closed under the convex hull operator, the over-approximation in (2.12) is used in (2.22b). As all operations in (2.22) have a polynomial computational complexity with respect to the augmented state space dimension $n_x + n_u$ [106], the presented reachability analysis of sampled-data LTI systems is well suited to be integrated into scalable robust control algorithms.

Subsequently, we construct the equivalent discrete-time (DT) system of the CT system in (2.20) at sampling times. Based on (2.22a), there exists an augmented disturbance sequence $\widetilde{w}(\cdot) \in \widetilde{\mathcal{W}}$ with not necessarily unique parameter vector sequence $\lambda_{\widetilde{w},(\cdot)} \in \mathbb{R}^{\mathtt{gen}(\widetilde{\mathcal{W}})}$ such that

$$\begin{aligned} \begin{bmatrix} x(t_{k+1}) \\ u(t_k) \end{bmatrix} &= e^{\widetilde{A}\Delta t} \begin{bmatrix} x(t_k) \\ u(t_k) \end{bmatrix} + \widetilde{w}(t_k) && \text{(2.23a)} \\ \widetilde{w}(t_k) &= c_{\widetilde{\mathcal{W}}} + G_{\widetilde{\mathcal{W}}} \lambda_{\widetilde{w},k} && \text{(2.23b)} \end{aligned}$$

with $\left|\lambda_{\widetilde{w},k}\right| \leq \mathbf{1}$ and $k \in \mathbb{N}$. Obtaining $\lambda_{\widetilde{w},k}$ in (2.23b) can be achieved by solving (2.1) for $\widetilde{w}(t_k)$ and $\widetilde{\mathcal{W}}$, i.e., by solving the COP

$$\begin{aligned} \underset{\lambda_{\widetilde{w},k}}{\text{minimize}} \quad & J_{\lambda_{\widetilde{w},k}}(\lambda_{\widetilde{w},k}) && \text{(2.24a)} \\ \text{subject to} \quad & \begin{bmatrix} x(t_{k+1}) \\ u(t_k) \end{bmatrix} - e^{\widetilde{A}\Delta t} \begin{bmatrix} x(t_k) \\ u(t_k) \end{bmatrix} \overset{(2.23)}{=} c_{\widetilde{\mathcal{W}}} + G_{\widetilde{\mathcal{W}}} \lambda_{\widetilde{w},k} && \text{(2.24b)} \\ & \left|\lambda_{\widetilde{w},k}\right| \leq \mathbf{1}, && \text{(2.24c)} \end{aligned}$$

where $J_{\lambda_{\widetilde{w},k}}$ is a convex cost function. Based on $\lambda_{\widetilde{w},(\cdot)}$, we define the concatenated disturbance parameter vector

$$\begin{aligned} \lambda_{\widetilde{\mathcal{W}},k} &= \begin{bmatrix} \lambda_{\widetilde{w},0}^T & \lambda_{\widetilde{w},1}^T & \dots & \lambda_{\widetilde{w},k}^T \end{bmatrix}^T \in \mathbb{R}^{(k+1)\mathtt{gen}(\widetilde{\mathcal{W}})} && \text{(2.25a)} \\ \lambda_{\widetilde{\mathcal{W}},-1} &\in \mathbb{R}^0 && \text{(2.25b)} \end{aligned}$$

with $\left|\lambda_{\widetilde{\mathcal{W}},k}\right| \leq \mathbf{1}$. Moreover, the matrix exponential $e^{\widetilde{A}\Delta t} \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$ can be expressed by

$$e^{\widetilde{A}\Delta t} = \begin{bmatrix} A_D & B_D \\ \mathbf{0} & I \end{bmatrix}, \tag{2.26}$$

where

$$\begin{aligned} A_D &= e^{A\Delta t} && \text{(2.27a)} \\ B_D &= \left(\int_0^{\Delta t} e^{A\tau} \, \mathrm{d}\tau \right) B && \text{(2.27b)} \end{aligned}$$

are the DT system and input matrices, respectively. Then, the equivalent DT system of the CT system in (2.20) at sampling times is

$$\begin{aligned} x(t_{k+1}) &= A_D x(t_k) + B_D u(t_k) + \Pi_x \widetilde{w}(t_k) \\ &= A_D x(t_k) + B_D u(t_k) + w_D(t_k), \end{aligned} \tag{2.28}$$

where $w_D(t_k)$ lies within the DT disturbance set $\mathcal{W}_D = \Pi_x\widetilde{\mathcal{W}} \subset \mathbb{R}^{n_x}$ for all $k \in \mathbb{N}$, i.e., $w_D(\cdot) \in \mathcal{W}_D$. To compactly represent these DT dynamics with corresponding DT disturbance set $\mathcal{W}_D$, we use the model $\mathbf{M} = (A_D, B_D, \mathcal{W}_D, \mathrm{DT})$.

Instead of implementing the presented reachability analysis by hand, many freely available software toolboxes can be used. For instance, reachable sets of LTI systems can be computed using, e.g., C2E2 [107], CORA [81], HyDRA [108], HyLAA [109], JuliaReach [110], SpaceEx [111], and XSpeed [112]. To compare the performance of these tools, they regularly participate in the friendly competition of the ARCH workshop [113]. In addition, there exist approaches that perform reachability analysis not only for LTI systems but also for, e.g., linear time-varying systems [114], nonlinear systems [115], and hybrid systems [94].

2.4 Invariant Sets

If the initial state $x(t_0) \in \mathbb{R}^{n_x}$ lies within an invariant set, future states are guaranteed to stay within this set indefinitely. Because invariant sets are fundamental for many control approaches, there exists a vast body of literature that focuses on the characterization and computation of such sets [39, 42, 116–119]. Subsequently, we define important invariant sets and present standard algorithms for computing such sets [38, 40].

First, we consider autonomous, DT, LTI systems. They can be obtained from DT, LTI systems with controllable inputs, e.g., by choosing the simple control law $u(t_k) = Kx(t_k)$ with state feedback matrix $K \in \mathbb{R}^{n_u \times n_x}$. By closing the control loop with such a state feedback controller, the system in (2.28) evolves according to

$$x(t_{k+1}) = (A_D + B_D K)\, x(t_k) + w_D(t_k)$$

with $A_D + B_D K$ being the overall system matrix. For a general model $\mathbf{M} = (A_D, \mathbf{0}, \mathcal{W}_D, \mathrm{DT})$, a set of states whose sequences stay within this set indefinitely despite all possible disturbances contained in $\mathcal{W}_D$ is a robust positively invariant (RPI) set, also known as disturbance invariant set, and introduced in the following definition [38, Def. 1].

Definition 2.11 (Robust Positively Invariant Set): The set $\mathcal{S} \subset \mathbb{R}^{n_x}$ is an RPI set for $\mathbf{M} = (A_D, \mathbf{0}, \mathcal{W}_D, \mathrm{DT})$, if $A_D\mathcal{S} \oplus \mathcal{W}_D \subseteq \mathcal{S}$, i.e., if the reachable set of $\mathcal{S}$ is contained in $\mathcal{S}$. ■

Thus, if the initial state $x(t_0)$ lies within an RPI set, the DT state sequence will never leave this set. An important RPI set is the one that cannot be under-approximated by another RPI set, which is introduced in the following definition [38, Def. 2].

Definition 2.12 (Minimal Robust Positively Invariant Set): The set $\mathcal{S} \subset \mathbb{R}^{n_x}$ is the minimal robust positively invariant (mRPI) set for $\mathbf{M} = (A_D, \mathbf{0}, \mathcal{W}_D, \mathrm{DT})$, if $\mathcal{S}$ is the RPI set in $\mathbb{R}^{n_x}$ that is contained in every closed RPI set for $\mathbf{M}$. ■

An illustration of the mRPI set along with another RPI set is shown in Fig. 2.6. In the following lemma, which is taken from [37, Sec. IV], we show that the mRPI set can be expressed as a series.

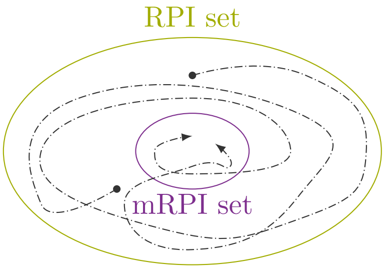


Figure 2.6: RPI set and two random state sequences, which never leave this set and converge to the mRPI set.

Lemma 2.13 (Minimal Robust Positively Invariant Set): Let $A_D \in \mathbb{R}^{n_x \times n_x}$ be asymptotically stable and $\mathcal{W}_D \subset \mathbb{R}^{n_x}$ be closed, bounded, and contain the origin. Then, the set $\bigoplus_{i=k}^{\infty} A_D^k \mathcal{W}_D$ is the mRPI set for $\mathbf{M} = (A_D, \mathbf{0}, \mathcal{W}_D, \mathrm{DT})$ [37, Sec. IV]. ■

Proof. Because $\mathbf{0} \in \mathcal{W}_D$, we choose $x(t_0) = \mathbf{0}$ and perform reachability analysis for the DT system:

$$\begin{aligned}
x(t_1) &\in A_D\{\mathbf{0}\} \oplus \mathcal{W}_D = \mathcal{W}_D \\
x(t_2) &\in A_D \mathcal{W}_D \oplus \mathcal{W}_D \\
x(t_3) &\in A_D(A_D \mathcal{W}_D \oplus \mathcal{W}_D) \oplus \mathcal{W}_D = A_D^2 \mathcal{W}_D \oplus A_D \mathcal{W}_D \oplus \mathcal{W}_D \\
&\;\vdots \\
x(t_k) &\in \underbrace{A_D^{k-1} \mathcal{W}_D \oplus A_D^{k-2} \mathcal{W}_D \oplus \ldots \oplus A_D \mathcal{W}_D \oplus \mathcal{W}_D}_{\bigoplus_{i=0}^{k-1} A_D^i \mathcal{W}_D} .
\end{aligned}$$

As k goes to infinity, we obtain the set $\bigoplus_{i=0}^{\infty} A_D^i \mathcal{W}_D$. □

It can be shown that $\bigoplus_{i=0}^{\infty} A_D^i \mathcal{W}_D$ exists, contains the origin, and is unique, closed, and bounded [37,38]. In addition, for any closed, bounded initial state set, the sequence of reachable sets converges to the mRPI set exponentially fast with respect to the Hausdorff distance [102]. Thus, the mRPI set is the limit set for all state sequences [37, Rmk. 4.1]. Due to the Minkowski sum of infinitely many terms, $\bigoplus_{i=0}^{\infty} A_D^i \mathcal{W}_D$ is not finitely determined unless A_D is a nilpotent matrix, i.e., unless $A_D^i = \mathbf{0}$ for some $i \in \mathbb{N}$ [103]. Nevertheless, there exist approaches to compute tight RPI over-approximations of the mRPI set [38,120–122]. For instance, Alg. 2.1 computes the RPI set $\frac{1}{1-\alpha} \bigoplus_{i=0}^{k} A_D^i \mathcal{W}_D$ with $k < \infty$ and suitable $\alpha \in [0,1)$ to over-approximate the mRPI set.

Before extending the notion of invariant sets to systems with controllable inputs, we define the DT state and input constraints

$$\begin{aligned}
x(\cdot) &\in \mathcal{X} && (2.29\text{a}) \\
u(\cdot) &\in \mathcal{U}, && (2.29\text{b})
\end{aligned}$$

where $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{U} \subset \mathbb{R}^{n_u}$ are the state and input constraint sets, respectively. Thus, the state and input sequences must always stay within the constraint sets $\mathcal{X}$ and $\mathcal{U}$, respectively.

Algorithm 2.1 RPI over-approximation of the mRPI set [38]

```
Input: $A_D, \mathcal{W}_D, \epsilon \in \mathbb{R}_{>0}$          ▷ $\epsilon$ is the convergence tolerance
Output: $\mathcal{S}_{\text{RPI}}$
$i \leftarrow 0$
$\alpha \leftarrow \infty$
$\beta \leftarrow 0$
$\mathcal{S}^{(1)} \leftarrow \{\mathbf{0}\}$
while $\frac{\epsilon}{\epsilon+\beta} < \alpha$ do
    $i \leftarrow i + 1$
    $\alpha \leftarrow \min\left\{\delta \in \mathbb{R}_{\geq 0} \mid A_D^i \mathcal{W}_D \subseteq \delta \mathcal{W}_D\right\}$
    $\mathcal{S}^{(i+1)} \leftarrow \mathcal{S}^{(i)} \oplus A_D^{i-1} \mathcal{W}_D$
    $\beta \leftarrow \min\left\{\delta \in \mathbb{R}_{\geq 0} \mid \mathcal{S}^{(i+1)} \subseteq \delta\,[-\mathbf{1}, \mathbf{1}]\right\}$
end while
$\mathcal{S}_{\text{RPI}} \leftarrow \frac{1}{1-\alpha} \mathcal{S}^{(i+1)}$
```

Based on (2.29), we introduce robust backward-reachable sets for a given terminal set, which are closely related to the robust forward-reachable sets for a given initial set in Definition 2.10. For $\mathbf{M} = (A_D, B_D, \mathcal{W}_D, \mathrm{DT})$, the set of states that can be robustly steered into a given terminal set in a single time step despite all possible disturbances contained in $\mathcal{W}_D$ is the one-step robust backward-reachable set and introduced in the following definition [42, Def. 10.15].

Definition 2.14 (One-Step Robust Backward-Reachable Set): The one-step robust backward-reachable set for the terminal set $\Omega \subset \mathbb{R}^{n_x}$ and the model $\mathbf{M} = (A_D, B_D, \mathcal{W}_D, \mathrm{DT})$ is

$$\mathcal{R}_{\text{back}}^{\mathbf{M}}(\Omega) = \{x \in \mathbb{R}^{n_x} \mid \exists u \in \mathcal{U} \text{ such that } \{A_D x + B_D u\} \oplus \mathcal{W}_D \subseteq \Omega\},$$

which is also known as the robust precursor set to Ω. ■

By iterating the one-step robust backward-reachable set computations k times with $k \in \mathbb{N}_{>0}$, the k-step robust backward-reachable set is obtained. This set is introduced in the following definition [42, Def. 10.18].

Definition 2.15 (k-Step Robust Backward-Reachable Set): The k-step robust backward-reachable set for $k \in \mathbb{N}_{>0}$, the terminal set $\Omega \subseteq \mathcal{X}$, and the model $\mathbf{M} = (A_D, B_D, \mathcal{W}_D, \mathrm{DT})$, is recursively defined by the set sequence

$$\mathcal{S}^{(1)} = \Omega \tag{2.30a}$$

$$\mathcal{S}^{(i+1)} = \mathcal{X} \cap \mathcal{R}_{\text{back}}^{\mathbf{M}}\left(\mathcal{S}^{(i)}\right), \tag{2.30b}$$

where $i \in \mathbb{N}_{[1,k]}$. The k-step robust backward-reachable set is also known as the k-step robust controllable set. ■

Thus, all states in the k-step robust controllable set can be robustly steered into a given terminal set in k steps despite the presence of disturbances while satisfying the constraints in (2.29). Subsequently, we introduce invariant sets for systems with controllable inputs.

For $\mathbf{M} = (A_D, B_D, \mathcal{W}_D, \mathrm{DT})$, a set of states for which there exists a controller such that its state and input sequences never violate the constraints in (2.29) despite all possible disturbances contained in $\mathcal{W}_D$ is a robust control invariant (RCI) set and introduced in the following definition [42, Def. 10.22].

Definition 2.16 (Robust Control Invariant Set): The set $\mathcal{S} \subseteq \mathcal{X}$ is an RCI set for $\mathbf{M} = (A_D, B_D, \mathcal{W}_D, \mathrm{DT})$, if for any $x \in \mathcal{S}$ there exists a $u \in \mathcal{U}$ such that $\{A_D x + B_D u\} \oplus \mathcal{W}_D \subseteq \mathcal{S}$. ■

Thus, if the initial state $x(t_0)$ lies within an RCI set, there always exists at least one control input in $\mathcal{U}$ such that the DT state sequence will never leave this set. An important RCI set is the one that contains all other RCI sets, which is introduced in the following definition [42, Def. 10.23].

Definition 2.17 (Maximal Robust Control Invariant Set): The set $\mathcal{S} \subseteq \mathcal{X}$ is the maximal robust control invariant (MRCI) set for $\mathbf{M} = (A_D, B_D, \mathcal{W}_D, \mathrm{DT})$, if $\mathcal{S}$ is an RCI set and contains all RCI sets contained in $\mathcal{X}$. ■

To compute the MRCI set, the recursively defined set sequence in (2.30) with terminal set $\mathcal{X}$ was proposed over 50 years ago [36], i.e.,

$$\mathcal{S}^{(1)} = \mathcal{X} \tag{2.31a}$$

$$\mathcal{S}^{(i+1)} = \mathcal{X} \cap \mathcal{R}^{\mathbf{M}}_{\text{back}}\left(\mathcal{S}^{(i)}\right), \tag{2.31b}$$

where $i \in \mathbb{N}_{>0}$. It can be shown that the set sequence in (2.31) converges to the MRCI set with respect to the Hausdorff distance as i goes to infinity [36]. However, the sets of the sequence in (2.31) are usually not RCI sets, and the MRCI set is typically not finitely determined [39]. Nevertheless, there exist approaches for computing tight RCI under-approximations of the MRCI set, such as Alg. 2.2 [40].

Algorithm 2.2 RCI under-approximation of the MRCI set [40]

```
Input: $A_D, B_D, \mathcal{X}, \mathcal{U}, \mathcal{W}_D, \epsilon \in \mathbb{R}_{>0}$        ▷ $\epsilon$ is the convergence tolerance
Output: $\mathcal{S}_{\text{RCI}}$
$i \leftarrow 1$
$\mathcal{S}^{(1)} \leftarrow \mathcal{X}$
while “true” do
    $\mathcal{S}^{(i+1)} \leftarrow \mathcal{X} \cap \left\{x \in \mathbb{R}^{n_x} \mid \exists u \in \mathcal{U} \text{ such that } \{A_D x + B_D u\} \oplus \mathcal{W}_D \oplus \epsilon\,[-\mathbf{1}, \mathbf{1}] \subseteq \mathcal{S}^{(i)}\right\}$
    if $\mathcal{S}^{(i)} \subseteq \mathcal{S}^{(i+1)} \oplus \epsilon\,[-\mathbf{1}, \mathbf{1}]$ then
        break
    end if
    $i \leftarrow i + 1$
end while
$\mathcal{S}_{\text{RCI}} \leftarrow \mathcal{S}^{(i+1)}$
```

So far, we have mainly focused on important invariant sets for DT systems. Nevertheless, the notion of invariant sets can be easily extended to deal with CT and sampled-data systems [39, 102, 123, 124]. For instance, a standard procedure to compute polytopic invariant sets for a CT system is based on constructing its corresponding DT Euler auxiliary system [39, Def. 4.25]. As a result, the presented approaches for DT systems can be exploited to compute invariant sets for the original CT systems [39, Lemma 4.26]. However, these standard approaches typically suffer from the curse of dimensionality due to using polytopes as underlying set representations. Thus, there is a lack of scalable algorithms for computing nonconservative sets that guarantee robust constraint satisfaction.

2.5 Software Setup

As reproducibility is crucially vital for advancing scientific research [125], we subsequently give an overview of the software toolboxes used throughout this thesis. All simulations are conducted on a Lenovo ThinkPad X1 Carbon Gen 9 laptop equipped with an Intel Core i7-1185G7 and 32 GB memory running Microsoft Windows 10 Home (version 21H2) and MATLAB (version R2021a Update 5). For performing reachability analysis using zonotopes as set representation and for visualizing convex sets, we use the open-source continuous reachability analysis toolbox CORA[1] [81]. In addition, for performing reachability analysis using polytopes as set representation, we use the open-source multi-parametric toolbox MPT3[2] [77].

To give an idea of the computational complexity when operating on polytopes [83], we report the computation times for adding two unit balls corresponding to the infinity norm represented in half-space representation $\left\langle \begin{bmatrix} I & -I \end{bmatrix}^T, \mathbf{1} \right\rangle_P$ in Table 2.1. Because this computation corresponds to the most straightforward nontrivial Minkowski addition, it is evident that polytopic methods are unsuitable for handling large-scale systems.

All COPs in this thesis are modeled using the open-source toolbox YALMIP[3] [66] with the parameter “allownonconvex” set to Boolean “false” and solved using the commercial mixed-integer conic programming solver MOSEK [64] with default parameters. When reporting the computation times for solving an optimization problem, we always exclude the time for modeling the optimization problem, i.e., for transforming the original problem into an equivalent one in standard form [55, Ch. 4]. To ensure that our simulations can be reproduced easily, most robust control algorithms proposed in this thesis are integrated into our seminal automated controller synthesis toolbox AROC[4] [48].

[1] tumcps.github.io/CORA

[2] mpt3.org

[3] yalmip.github.io

[4] tumcps.github.io/AROC

Table 2.1: Minkowski addition of two unit balls represented in half-space representation in $\mathbb{R}^n$.

n	computation time [s]
1	0.068
2	0.031
3	0.015
4	1.066
5	2.107
6	4.826
7	14.273
8	37.497
9	107.178
10	323.856
11	974.611
12	3464.203
13	20 208.689
14	118 398.148

3 Safe Sets

In this chapter, which is based on [47, 49–51], we synthesize zonotopic safe sets along with corresponding safety-preserving controllers, which ensure the safety of the sampled-data system for an infinite time horizon. After reviewing the relevant literature in Section 3.1, we formulate the control goal of this chapter in Section 3.2. In Section 3.3, we perform reachability analysis using a state and a disturbance feedback controller. These reachable set computations are the core of our robust control approaches for computing safe sets with minimum or maximum volume, as presented in Sections 3.4 and 3.5. To demonstrate the effectiveness of our safe set approaches, we consider four numerical examples in Section 3.6. Finally, we summarize this chapter in Section 3.7.

3.1 Introduction and State of the Art

Guaranteeing safety for an infinite time horizon is challenging to formally verify and yet crucially important for leveraging autonomous systems or learning-based control in safety-critical applications, such as autonomous driving or robot-assisted surgery [126, 127]. Thus, the sets of safe states along with corresponding safety-preserving controllers, which guarantee robust state and input constraint satisfaction at all times despite disturbances, are widely used in the robust controller synthesis.

For instance, a large safe robust positively invariant (RPI) terminal set with a corresponding terminal penalty is typically used for ensuring recursive feasibility in robust model predictive control (MPC) [41–43, 128], which is presented in more detail in Chapter 4. As soon as the system state enters this safe set, the safety-preserving terminal controller guarantees the robust satisfaction of the state and input constraints at all future times. Safe sets are also used in learning-based control as part of a supervisory safety filter [129, 130], which is presented in more detail in Chapter 5. This filter accepts only inputs that satisfy the input constraint and cause the state of the system to stay within the safe set. If the desired control input is rejected, the safety-preserving backup control is applied instead. To maximize the region of operation for the robust MPC approach and the safety filter, it is desirable to use safe sets with maximum volume, i.e., safe sets that are as large as possible.

The largest safe set is known as the discriminating kernel [119, 131], infinite-time reachable set [36], or maximal robust control invariant (MRCI) set [39, 40], which was introduced in Definition 2.17. Because of its high relevance in robust controller synthesis, computing the exact discriminating kernel and approximations thereof has a long history. The exact set for discrete-time (DT) systems can be obtained by the standard set recursion in (2.31) [36]. However, this procedure fails to terminate in finite time in most cases [39]. Thus, various approaches for computing approximations exist in the literature.

Polytopic robust control invariant (RCI) under-approximations and over-approximations of the MRCI set are presented in [40], where arbitrarily small violations of the state and input constraints are tolerated in the case of an over-approximation. The corresponding algorithm for computing an RCI under-approximation is presented in Alg. 2.2. To prevent the polytopic representation of an RCI set from becoming too complex, its desired number of representing half-spaces can be fixed or be chosen freely [132–135]. To obtain such RCI sets of desired complexity, a sequence of semidefinite programming (SDP) problems is solved that enforces the iterates to be RCI. In addition to explicit set representations, RCI sets are also represented implicitly [136, 137], e.g., by the Minkowski sum of a finite number of polytopes [138]. Then, the corresponding safety-preserving control input is usually obtained by solving a convex optimization problem (COP). However, recovering an explicit representation of such an RCI polytope is typically intractable due to the exponential computational complexity of the required set operations [25, 83].

To decrease the computational complexity when constructing invariant sets, ellipsoids instead of polytopes are widely used as set representation [39,139–141]. For instance, a scalable approach for computing an under-approximation of the finite-horizon discriminating kernel is presented in [142]. However, safety is ensured only for a finite time horizon. Nevertheless, compared to the exponential computational complexity of the standard polytopic approach with respect to the state space dimension, representing reachable sets by ellipsoids results in increased scalability. As a scalable alternative, zonotopes and bundles of multidimensional interval over-approximations of zonotopes are also used as set representation [143, 144]. Because zonotopes can exactly represent typical multidimensional interval constraints, zonotopic approximations often produce significantly less conservative results compared to ellipsoidal ones [143]. In summary, most existing approaches are unsuitable to ensure the safety of large-scale systems due to their conservativeness, exponential computational complexity, or limitation to finite time horizons.

In addition to safe sets that are as large as possible, it is also important to compute safe sets that are as small as possible [37–39, 102, 120–122, 145]. For instance, these small safe sets are widely used in compositional controller synthesis when treating the coupling between subsystems as disturbances [146, 147]. We also want to mention that a minimal RCI set that is contained in all other RCI sets does not exist in general [148]. This nonexistence contrasts the MRCI set, which contains all other RCI sets. To emphasize the importance of safe sets in general, we also want to mention that even the computation of (nonrobust) control invariant sets is an active area of research [149–151]. Similarly, the scalable computation of RCI sets of nonlinear systems is a challenging task [52, 152–154].

When considering stabilizable linear time-invariant (LTI) systems with full knowledge of the state, a simple linear state feedback controller can be computed, e.g., by linear-quadratic regulator (LQR)-based controller synthesis [155]. However, optimization problems that perform set-based reachability analysis while treating such state feedback matrices as optimization variables are nonconvex due to the multiplication of these matrices. Thus, the problem of determining suitable state feedback matrices by optimizing over reachable sets is typically intractable. Nevertheless, parameterizing the controller as an affine map of the past disturbances instead of the current state enables an equivalent convex optimization over disturbance feedback matrices [156, 157]. Closely related control parameterizations include, e.g., system level synthesis [158] and Youla parameterization [159].

It is clear from the presented literature review that guaranteeing safety for an infinite time horizon is typically achieved by computing RCI sets. Nevertheless, providing formal safety guarantees does not require a safe set to be RCI. In this chapter, we use this simple idea to compute not necessarily RCI safe sets, which allows us to reduce the computational complexity and use a simple set representation. In particular, we synthesize explicit, zonotopic safe sets along with corresponding safety-preserving controllers, which ensure the safety of the sampled-data system for an infinite time horizon. Thus, we guarantee robust constraint satisfaction not only at but also between sampling times. This guarantee also contrasts most existing approaches, which usually consider DT systems. Moreover, we use set-based state feedback controllers and set-based disturbance feedback controllers to obtain a simple controller structure. Incorporating these two controllers into our reachability analysis and using efficiently solvable COPs allows us to compute safe sets of large-scale systems because the corresponding computational complexity of our algorithms is polynomial with respect to the problem dimension. Because safe sets are usually desired to have minimum or maximum volume, we propose several scalable approaches for computing such sets in the following sections.

3.2 Problem Formulation

In this chapter, we consider a continuous-time (CT), LTI system that evolves according to (2.20), i.e., it is compactly represented by the model $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$. Similar to the DT constraints in (2.29), this system is constrained by

$$x(\cdot) \in \mathcal{X} \tag{3.1a}$$

$$u(\cdot) \in \mathcal{U}, \tag{3.1b}$$

where $\mathcal{X} = \langle H_\mathcal{X}, h_\mathcal{X} \rangle_P \subset \mathbb{R}^{n_x}$ and $\mathcal{U} = \langle H_\mathcal{U}, h_\mathcal{U} \rangle_P \subset \mathbb{R}^{n_u}$ are the given state and input constraint sets, respectively. Without loss of generality, we assume that the disturbance set $\mathcal{W} = \langle c_\mathcal{W}, G_\mathcal{W} \rangle_Z$ contains the origin $\{\mathbf{0}\}$, which can typically be satisfied by performing a suitable coordinate transformation. Because $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}$ are usually represented by multidimensional intervals, they can be easily expressed in both half-space and generator representations using (2.5) and (2.6). If other set representations are chosen, tight polytopic under-approximations of $\mathcal{X}$ and $\mathcal{U}$ and a zonotopic over-approximation of $\mathcal{W}$ must be computed to maintain the formal safety guarantees.

The initial state of the system $x(t_0) \in \mathbb{R}^{n_x}$ lies within the initial state set $\mathcal{Z}_x(t_0) = \langle c_x(t_0), G_x(t_0) \rangle_Z \subseteq \mathcal{X}$, i.e., it can be expressed by

$$x(t_0) = c_x(t_0) + G_x(t_0)\lambda_{x,0}, \tag{3.2}$$

where a not necessarily unique initial parameter vector $\lambda_{x,0} \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z}_x(t_0))}$ with $|\lambda_{x,0}| \leq \mathbf{1}$ is guaranteed to exist. Obtaining $\lambda_{x,0}$ can be achieved by solving (2.1) for $x(t_0)$ and $\mathcal{Z}_x(t_0)$, i.e., by solving the COP

$$\underset{\lambda_{x,0}}{\text{minimize}} \quad J_{\lambda_{x,0}}(\lambda_{x,0}) \tag{3.3a}$$

$$\text{subject to} \quad x(t_0) = c_x(t_0) + G_x(t_0)\lambda_{x,0} \tag{3.3b}$$

$$|\lambda_{x,0}| \leq \mathbf{1}, \tag{3.3c}$$

where $J_{\lambda_{x,0}}$ is a convex cost function.

Before formulating the control problem we want to solve in this chapter, we define safe sets associated with the safety constraints in (3.1). These sets are not necessarily RCI sets and are crucial for guaranteeing robust constraint satisfaction for an infinite time horizon, as mentioned in Section 3.1 and illustrated in Fig. 1.3.

Definition 3.1 (Safe Set): The set $\mathcal{S} \subseteq \mathcal{X}$ is a safe set for the model $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$, if a safety-preserving controller exists such that the safety constraints in (3.1) are satisfied for all $x(t_0) \in \mathcal{S}$. ■

To formulate a meaningful sampled-data control problem, we assume that (A_D, B_D) is stabilizable, where $A_D \in \mathbb{R}^{n_x \times n_x}$ and $B_D \in \mathbb{R}^{n_x \times n_u}$ are the DT system and input matrices defined in (2.27). Then, the control goal of this chapter is to determine an initial state set $\mathcal{Z}_x(t_0)$ along with a corresponding safety-preserving controller such that $\mathcal{Z}_x(t_0)$ is a safe set with minimum or maximum volume.

3.3 Reachability Analysis

In this section, we perform reachability analysis for the augmented system in (2.21) when using the following two piecewise constant controllers: a state feedback controller and a disturbance feedback controller. These reachable set computations are the core of our scalable robust control approaches for synthesizing safe sets, which we present in the subsequent sections.

3.3.1 State Feedback Control

To use a state feedback controller, we assume a stabilizing state feedback matrix $K \in \mathbb{R}^{n_u \times n_x}$ to be given. Because the tuple (A_D, B_D) defined in (2.27) is assumed to be stabilizable in Section 3.2, a stabilizing K can be easily obtained, e.g., by LQR-based controller synthesis [155]. Based on this K and the not necessarily unique initial parameter vector $\lambda_{x,0}$ in (3.2), we use the piecewise constant state feedback control law

$$u(t) = Kx(t_k) + c_u(t_k) + G_u(t_k)\lambda_{x,0} \quad \text{for } t \in [t_k, t_{k+1}), \tag{3.4}$$

where $\mathcal{Z}_u(t_k) = \langle c_u(t_k), G_u(t_k) \rangle_Z \subset \mathbb{R}^{n_u}$ with generator matrix $G_u(t_k) \in \mathbb{R}^{n_u \times \mathtt{gen}(\mathcal{Z}_x(t_0))}$ is the correction input zonotope at sampling time $t_k = k\Delta t$. Thus, in addition to the zonotopic parameterized interpolation-based control used in [143], our controller in (3.4) also consists of a stabilizing state feedback component to enhance the control performance [156]. To explicitly obtain the control law in (3.4) for any time $t \in \mathbb{R}_{\geq 0}$, the initial parameter vector $\lambda_{x,0}$ is computed at the initial time t_0 by solving the COP in (3.3).

Subsequently, we compute reachable sets when using the state feedback controller in (3.4) for an arbitrary sampling time t_k and time interval $[t_k, t_{k+1})$. To account for the piecewise constant control law in (3.4), we first compute reachable sets for consecutive time steps of size Δt until the specified sampling time is reached. Based on Section 2.3, we introduce the following recursively defined set sequence for the state feedback controller in (3.4) and the

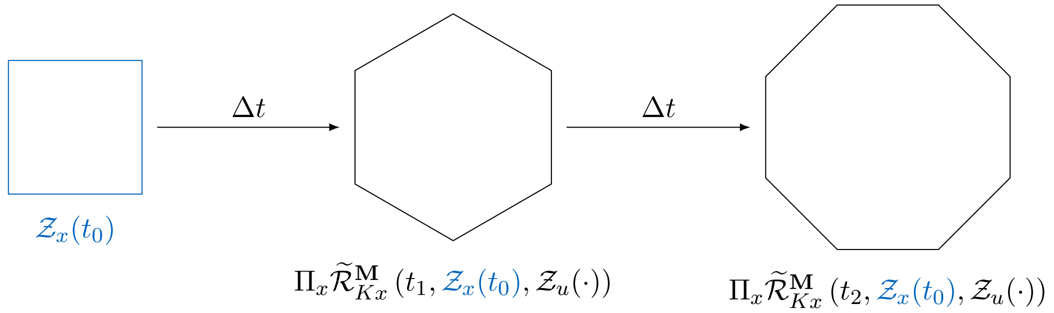

$\mathcal{Z}_x(t_0)$
Δt
$\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_1, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot))$
Δt
$\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_2, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot))$

Figure 3.1: Reachable sets defined in (3.5) for the initial state set $\mathcal{Z}_x(t_0) \subseteq \mathcal{X}$ and the specified time t_2.

model $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$, which is illustrated in Fig. 3.1 for t_2:

$$\begin{aligned}\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)) &= \left\langle c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))}, G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \right\rangle_Z \\ &= \left\langle \begin{bmatrix} c_x(t_k) \\ Kc_x(t_k) + c_u(t_k) \end{bmatrix}, \begin{bmatrix} G_x(t_k) \\ KG_x(t_k) + \begin{bmatrix} G_u(t_k) & \mathbf{0} \end{bmatrix} \end{bmatrix} \right\rangle_Z \end{aligned} \tag{3.5a}$$

$$\begin{aligned}\langle c_x(t_{k+1}), G_x(t_{k+1})\rangle_Z &= \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{over}}\left(\Delta t, \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot))\right) \\ &\overset{(2.22a)}{=} \Pi_x \left(e^{\tilde{A}\Delta t} \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)) \oplus \widetilde{\mathcal{W}}\right), \end{aligned} \tag{3.5b}$$

where $k \in \mathbb{N}$. In the following theorem, we prove that the sets in (3.5) are over-approximating the augmented reachable sets of $\mathbf{M}$ when using the controller in (3.4).

Theorem 3.2 (Set Propagation using State Feedback Control): For all $x(t_0) \in \mathcal{Z}_x(t_0)$, applying the state feedback controller in (3.4) to $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$ results in

$$\begin{bmatrix} x(t_k) \\ u(t_k) \end{bmatrix} \in \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)),$$

where $k \in \mathbb{N}$. ■

Proof. First, we prove that applying the state feedback controller in (3.4) to $\mathbf{M}$ results in

$$\begin{bmatrix} x(t_k) \\ u(t_k) \end{bmatrix} = c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k-1} \end{bmatrix}, \tag{3.6}$$

where $\lambda_{\widetilde{\mathcal{W}},k-1} \in \mathbb{R}^{k\mathtt{gen}(\widetilde{\mathcal{W}})}$ is defined in (2.25). We proceed by induction:

Base case: For $k = 0$, we obtain

$$\begin{aligned}\begin{bmatrix} x(t_0) \\ u(t_0) \end{bmatrix} &\overset{(3.2),(3.4)}{=} \begin{bmatrix} c_x(t_0) \\ Kc_x(t_0) + c_u(t_0) \end{bmatrix} + \begin{bmatrix} G_x(t_0) \\ KG_x(t_0) + G_u(t_0) \end{bmatrix} \lambda_{x,0} \\ &\overset{(3.5a)}{=} c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_0,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_0,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \lambda_{x,0} \\ &\overset{(2.25)}{=} c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_0,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_0,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},-1} \end{bmatrix}. \end{aligned}$$

Induction hypothesis: (3.6) holds for some arbitrary $k \in \mathbb{N}$.
Induction step: For $k + 1$, the state is

$$\begin{aligned} x(t_{k+1}) &\overset{(2.19),(2.23a)}{=} \Pi_x \left(e^{\widetilde{A}\Delta t} \begin{bmatrix} x(t_k) \\ u(t_k) \end{bmatrix} + \widetilde{w}(t_k) \right) \\ &\overset{(3.6)}{=} \Pi_x \left(e^{\widetilde{A}\Delta t} \left(c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k-1} \end{bmatrix} \right) + \widetilde{w}(t_k) \right) \\ &\overset{(2.23b)}{=} \Pi_x \Bigg(e^{\widetilde{A}\Delta t} c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + e^{\widetilde{A}\Delta t} G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k-1} \end{bmatrix} \\ &\qquad\qquad + c_{\widetilde{\mathcal{W}}} + G_{\widetilde{\mathcal{W}}} \lambda_{\widetilde{w},k} \Bigg) \\ &\overset{(2.25)}{=} \Pi_x \left(e^{\widetilde{A}\Delta t} c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + c_{\widetilde{\mathcal{W}}} + \begin{bmatrix} e^{\widetilde{A}\Delta t} G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} & G_{\widetilde{\mathcal{W}}} \end{bmatrix} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k} \end{bmatrix} \right) \\ &\overset{(2.10a),(2.19),(3.5a),(3.5b)}{=} \Pi_x \left(c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k} \end{bmatrix} \right), \end{aligned} \tag{3.7}$$

where a not necessarily unique $\lambda_{\widetilde{w},k} \in \mathbb{R}^{\texttt{gen}(\widetilde{\mathcal{W}})}$ with $\left|\lambda_{\widetilde{w},k}\right| \leq \mathbf{1}$ is guaranteed to exist because $\widetilde{w}(t_k) \in \widetilde{\mathcal{W}}$. Similarly, the input at time step $k+1$ is

$$\begin{aligned}
u(t_{k+1}) &\overset{(3.4)}{=} Kx(t_{k+1}) + c_u(t_{k+1}) + G_u(t_{k+1})\lambda_{x,0} \\
&\overset{(3.7)}{=} K\Pi_x \left(c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k} \end{bmatrix} \right) \\
&\quad + c_u(t_{k+1}) + G_u(t_{k+1})\lambda_{x,0} \\
&\overset{(2.19),(2.25)}{=} K\Pi_x c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + c_u(t_{k+1}) \\
&\qquad + K\Pi_x G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k} \end{bmatrix} + \begin{bmatrix} G_u(t_{k+1}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k} \end{bmatrix} \\
&\overset{(2.10a),(3.5a),(3.5b)}{=} \Pi_u c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + \Pi_u G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k} \end{bmatrix} \\
&\overset{(2.19)}{=} \Pi_u \left(c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k} \end{bmatrix} \right),
\end{aligned}$$

which completes the proof of (3.6). It also becomes clear from this proof that we horizontally concatenate $G_u(t_k) \in \mathbb{R}^{n_u \times \texttt{gen}(\mathcal{Z}_x(t_0))}$ and a matrix of zeros in (3.5a) to account for the Minkowski addition resulting from the augmented reachable disturbance set $\widetilde{\mathcal{W}} \subset \mathbb{R}^{n_x+n_u}$.

Based on (3.6), $|\lambda_{x,0}| \leq \mathbf{1}$, $\left|\lambda_{\widetilde{\mathcal{W}},k-1}\right| \leq \mathbf{1}$, and Definition 2.6, it follows that the sets in (3.5) are over-approximating the augmented reachable sets of $\mathbf{M}$ when using the controller in (3.4). □

We have focused on performing reachability analysis for discrete sampling times when using the state feedback controller in (3.4). Nevertheless, the state and input constraints in (3.1) must be satisfied not only at but also between sampling times. Thus, based on Theorem 3.2 and (2.22b), we compute reachable sets for an arbitrary time interval $[t_k, t_{k+1})$ according to

$$\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_k, t_{k+1}), \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) = \widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{over}}\left([0, \Delta t), \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right)\right). \tag{3.8}$$

Then, the projection of the over-approximative reachable set onto the original state and input space is obtained by $\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(\cdot, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right)$ and $\Pi_u \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(\cdot, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right)$, respectively. In summary, we can efficiently compute the set of states and inputs that are reachable for all $x(t_0) \in \mathcal{Z}_x(t_0)$ when applying the state feedback controller in (3.4) to $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$.

Because the model $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$ is time-invariant, we can also separate the reachable sets in (3.5) and (3.8) into controllable and uncontrollable parts based on the superposition principle. To enable formal safety guarantees, we separate all involved set operations in an over-approximative way, as presented subsequently.

Lemma 3.3 (Separation of Generator Matrix): For $c \in \mathbb{R}^n$ and $G_1, G_2 \in \mathbb{R}^{n \times m}$, the following set relation holds:

$$\langle c, G_1 + G_2 \rangle_Z \subseteq \left\langle c, \begin{bmatrix} G_1 & G_2 \end{bmatrix} \right\rangle_Z .$$

In addition, it holds with equality if $G_1 = \mathbf{0}$ or $G_2 = \mathbf{0}$. ■

Proof. By using the introduction of zonotopes in Definition 2.6, we obtain

$$\begin{aligned}\langle c, G_1 + G_2\rangle_Z &= \{s \in \mathbb{R}^n \mid s = c + (G_1 + G_2)\lambda, |\lambda| \leq \mathbf{1}\}\\ &= \{s \in \mathbb{R}^n \mid s = c + G_1\lambda + G_2\lambda, |\lambda| \leq \mathbf{1}\}\\ &\subseteq \{s \in \mathbb{R}^n \mid s = c + G_1\lambda_1 + G_2\lambda_2, |\lambda_1| \leq \mathbf{1}, |\lambda_2| \leq \mathbf{1}\}\\ &= \left\langle c, \begin{bmatrix} G_1 & G_2 \end{bmatrix} \right\rangle_Z .\end{aligned}$$

Moreover, vectors full of zeros in the generator matrix of a zonotope do not affect the zonotope and, thus, can be erased. Because changing the order of the columns of the generator matrix also has no effect on the zonotope, the relation above holds with equality if $G_1 = \mathbf{0}$ or $G_2 = \mathbf{0}$. □

Lemma 3.3 is used in the following proposition to separate the over-approximative convex hull operator in (2.12), which is required for computing (3.8).

Proposition 3.4 (Separation of Convex Hull): For $M \in \mathbb{R}^{n\times n}$, $\mathcal{Z}_1 = \langle c_1, G_1\rangle_Z \subset \mathbb{R}^n$, and $\mathcal{Z}_2 = \langle c_2, G_2\rangle_Z \subset \mathbb{R}^n$ with $\texttt{gen}(\mathcal{Z}_1) = \texttt{gen}(\mathcal{Z}_2)$, the following over-approximative separation of the over-approximative convex hull operator in (2.12) holds:

$$\texttt{conv}_{\text{over}}(\mathcal{Z}_1 \oplus \mathcal{Z}_2, M(\mathcal{Z}_1 \oplus \mathcal{Z}_2)) \subseteq \texttt{conv}_{\text{over}}(\mathcal{Z}_1, M\mathcal{Z}_1) \oplus \texttt{conv}_{\text{over}}(\mathcal{Z}_2, M\mathcal{Z}_2).$$

In addition, the relation holds with equality if $c_1 = \mathbf{0}$ or $c_2 = \mathbf{0}$. ■

Proof. We prove this relation by

$$\begin{aligned}&\texttt{conv}_{\text{over}}(\mathcal{Z}_1 \oplus \mathcal{Z}_2, M(\mathcal{Z}_1 \oplus \mathcal{Z}_2))\\ &\overset{\text{(2.10a),(2.10b)}}{=} \texttt{conv}_{\text{over}}\left(\left\langle c_1 + c_2, \begin{bmatrix} G_1 & G_2 \end{bmatrix}\right\rangle_Z, \left\langle Mc_1 + Mc_2, \begin{bmatrix} MG_1 & MG_2 \end{bmatrix}\right\rangle_Z\right)\\ &\overset{(2.12)}{=} \frac{1}{2}\Big\langle c_1 + Mc_1 + c_2 + Mc_2,\\ &\qquad \begin{bmatrix} G_1 + MG_1 & G_2 + MG_2 & c_1 - Mc_1 + c_2 - Mc_2 & G_1 - MG_1 & G_2 - MG_2 \end{bmatrix}\Big\rangle_Z\\ &\overset{\text{Lemma 3.3}}{\subseteq} \frac{1}{2}\Big\langle c_1 + Mc_1 + c_2 + Mc_2,\\ &\qquad \begin{bmatrix} G_1 + MG_1 & G_2 + MG_2 & c_1 - Mc_1 & c_2 - Mc_2 & G_1 - MG_1 & G_2 - MG_2 \end{bmatrix}\Big\rangle_Z\\ &\overset{\text{(2.10a)}}{=} \frac{1}{2}\left\langle c_1 + Mc_1, \begin{bmatrix} G_1 + MG_1 & c_1 - Mc_1 & G_1 - MG_1 \end{bmatrix}\right\rangle_Z\\ &\qquad \oplus \frac{1}{2}\left\langle c_2 + Mc_2, \begin{bmatrix} G_2 + MG_2 & c_2 - Mc_2 & G_2 - MG_2 \end{bmatrix}\right\rangle_Z\\ &\overset{(2.12)}{=} \texttt{conv}_{\text{over}}(\mathcal{Z}_1, M\mathcal{Z}_1) \oplus \texttt{conv}_{\text{over}}(\mathcal{Z}_2, M\mathcal{Z}_2).\end{aligned}$$

Based on Lemma 3.3, the relation holds with equality if $c_1 = \mathbf{0}$ or $c_2 = \mathbf{0}$. □

In addition to the over-approximative convex hull operator, we also need to separate the over-approximative interval matrix multiplication in (2.13), as presented in the following proposition.

Proposition 3.5 (Separation of Interval Matrix Multiplication): For $\boldsymbol{\mathcal{M}} \subset \mathbb{R}^{n\times n}$, $\mathcal{Z}_1 = \langle c_1, G_1 \rangle_Z \subset \mathbb{R}^n$, and $\mathcal{Z}_2 = \langle c_2, G_2 \rangle_Z \subset \mathbb{R}^n$, the following over-approximative separation of the over-approximative interval matrix multiplication operator in (2.13) holds:

$$\boldsymbol{\mathcal{M}} \otimes_{\text{over}} (\mathcal{Z}_1 \oplus \mathcal{Z}_2) \subseteq \boldsymbol{\mathcal{M}} \otimes_{\text{over}} \mathcal{Z}_1 \oplus \boldsymbol{\mathcal{M}} \otimes_{\text{over}} \mathcal{Z}_2.$$

In addition, the relation holds with equality if $c_1 = \mathbf{0}$ or $c_2 = \mathbf{0}$. ■

Proof. First, we need the following simple relations that follow from the triangle inequality and the introduction of zonotopes in Definition 2.6 for $s_1, s_2 \in \mathbb{R}$, $c \in \mathbb{R}^n$, and $v_1, v_2 \in \mathbb{R}^n_{\geq 0}$ with $v_1 \leq v_2$:

$$|s_1 + s_2| \leq |s_1| + |s_2| \tag{3.9a}$$

$$\langle c, \texttt{diag}(v_1) \rangle_Z \subseteq \langle c, \texttt{diag}(v_2) \rangle_Z \tag{3.9b}$$

$$\texttt{diag}(v_1 + v_2) = \texttt{diag}(v_1) + \texttt{diag}(v_2) \tag{3.9c}$$

$$\langle c, \texttt{diag}(v_1) + \texttt{diag}(v_2) \rangle_Z = \left\langle c, \begin{bmatrix} \texttt{diag}(v_1) & \texttt{diag}(v_2) \end{bmatrix} \right\rangle_Z . \tag{3.9d}$$

In addition, let $C = \texttt{center}\,(\boldsymbol{\mathcal{M}}) \in \mathbb{R}^{n\times n}$ be the center of $\boldsymbol{\mathcal{M}}$ and let $R = \texttt{radius}\,(\boldsymbol{\mathcal{M}}) \in \mathbb{R}^{n\times n}_{\geq 0}$ be the radius of $\boldsymbol{\mathcal{M}}$. Finally, we prove this relation by

$$\begin{aligned}
&\boldsymbol{\mathcal{M}} \otimes_{\text{over}} (\mathcal{Z}_1 \oplus \mathcal{Z}_2) \\
&\overset{(2.10a)}{=} \boldsymbol{\mathcal{M}} \otimes_{\text{over}} \left\langle c_1 + c_2, \begin{bmatrix} G_1 & G_2 \end{bmatrix} \right\rangle_Z \\
&\overset{(2.13)}{=} \left\langle Cc_1 + Cc_2, \left[CG_1 \quad CG_2 \quad \texttt{diag}\left(R \left| \begin{bmatrix} c_1 + c_2 & G_1 & G_2 \end{bmatrix} \right| \mathbf{1} \right) \right] \right\rangle_Z \\
&\overset{(3.9a),(3.9b)}{\subseteq} \left\langle Cc_1 + Cc_2, \left[CG_1 \quad CG_2 \quad \texttt{diag}\left(R \left| \begin{bmatrix} c_1 & c_2 & G_1 & G_2 \end{bmatrix} \right| \mathbf{1} \right) \right] \right\rangle_Z \\
&\overset{(3.9c)}{=} \left\langle Cc_1 + Cc_2, \left[CG_1 \quad CG_2 \quad \texttt{diag}\left(R \left| \begin{bmatrix} c_1 & G_1 \end{bmatrix} \right| \mathbf{1} \right) + \texttt{diag}\left(R \left| \begin{bmatrix} c_2 & G_2 \end{bmatrix} \right| \mathbf{1} \right) \right] \right\rangle_Z \\
&\overset{(3.9d)}{=} \left\langle Cc_1 + Cc_2, \left[CG_1 \quad CG_2 \quad \texttt{diag}\left(R \left| \begin{bmatrix} c_1 & G_1 \end{bmatrix} \right| \mathbf{1} \right) \quad \texttt{diag}\left(R \left| \begin{bmatrix} c_2 & G_2 \end{bmatrix} \right| \mathbf{1} \right) \right] \right\rangle_Z \\
&\overset{(2.10a)}{=} \left\langle Cc_1, \left[CG_1 \quad \texttt{diag}\left(R \left| \begin{bmatrix} c_1 & G_1 \end{bmatrix} \right| \mathbf{1} \right) \right] \right\rangle_Z \oplus \left\langle Cc_2, \left[CG_2 \quad \texttt{diag}\left(R \left| \begin{bmatrix} c_2 & G_2 \end{bmatrix} \right| \mathbf{1} \right) \right] \right\rangle_Z \\
&\overset{(2.13)}{=} \boldsymbol{\mathcal{M}} \otimes_{\text{over}} \mathcal{Z}_1 \oplus \boldsymbol{\mathcal{M}} \otimes_{\text{over}} \mathcal{Z}_2.
\end{aligned}$$

Based on the triangle inequality, the relation holds with equality if $c_1 = \mathbf{0}$ or $c_2 = \mathbf{0}$. □

Finally, we separate the reachable sets in (3.5) and (3.8) into controllable and uncontrollable parts, as presented in the following theorem.

Theorem 3.6 (Separation of State Feedback Reachable Sets): The reachable sets in (3.5) and (3.8) for $\mathbf{M} = (A, B, \mathcal{W}, \text{CT})$ can be separated by

$$\begin{aligned}
\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) &= \widetilde{\mathcal{R}}^{(A,B,\{\mathbf{0}\},\text{CT})}_{Kx}\left(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \oplus \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_k, \{\mathbf{0}\}, \{\mathbf{0}\}\right) \\
\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_k, t_{k+1}), \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) &\subseteq \widetilde{\mathcal{R}}^{(A,B,\{\mathbf{0}\},\text{CT})}_{Kx}\left([t_k, t_{k+1}), \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \\
&\qquad \oplus \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_k, t_{k+1}), \{\mathbf{0}\}, \{\mathbf{0}\}\right).
\end{aligned}$$

■

Proof. For $M \in \mathbb{R}^{m\times n}$ and $\mathcal{Z}_1, \mathcal{Z}_2 \subset \mathbb{R}^n$, we obtain

$$M(\mathcal{Z}_1 \oplus \mathcal{Z}_2) \overset{(2.10a),(2.10b)}{=} M\mathcal{Z}_1 \oplus M\mathcal{Z}_2.$$

This result shows that all operations required to compute the reachable set at sampling times in (3.5) are linear functions. In addition to these operators, computing the reachable sets for time intervals involves the over-approximative convex hull and over-approximative matrix interval multiplication. Based on Propositions 3.4 and 3.5, the reachable sets for time intervals can be separated over-approximatively as claimed. □

We use Theorem 3.6 to efficiently compute safe sets when using the state feedback controller in (3.4). Before presenting these computations, we perform reachability analysis when using disturbance feedback control instead of state feedback control.

3.3.2 Disturbance Feedback Control

Based on the not necessarily unique initial parameter vector $\lambda_{x,0}$ in (3.2) and the concatenated disturbance parameter vector $\lambda_{\widetilde{\mathcal{W}},k-1}$ in (2.25), we use the piecewise constant disturbance feedback control law

$$u(t) = c_u(t_k) + G_u(t_k) \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k-1} \end{bmatrix} \quad \text{for } t \in [t_k, t_{k+1}), \tag{3.10}$$

where $\mathcal{Z}_u(t_k) = \langle c_u(t_k), G_u(t_k)\rangle_Z$ with generator matrix $G_u(t_k) \in \mathbb{R}^{n_u\times\left(\mathtt{gen}(\mathcal{Z}_x(t_0))+k\mathtt{gen}(\widetilde{\mathcal{W}})\right)}$ is the correction input zonotope at sampling time $t_k = k\Delta t$. Thus, the size of $G_u(t_k)$ grows linearly with k in contrast to the state feedback controller in (3.4). As a result, the disturbance feedback controller in (3.10) is usually more flexible but also more complex compared to (3.4). As for (3.4), we compute $\lambda_{x,0}$ by solving (3.3) at t_0 to explicitly obtain the control law in (3.10) for any time $t \in \mathbb{R}_{\geq 0}$. In addition, we also compute $\lambda_{\widetilde{\mathcal{W}},k-1}$ at t_k, which is defined in (2.25) as the concatenation of the disturbance parameter vectors $\lambda_{\widetilde{w},i}$ with $i \in \mathbb{N}_{[0,k-1]}$. Thus, we compute $\lambda_{\widetilde{\mathcal{W}},k-1}$ by determining $\lambda_{\widetilde{w},k-1}$ at t_k based on (2.24) and appending $\lambda_{\widetilde{w},k-1}$ to $\lambda_{\widetilde{\mathcal{W}},k-2}$.

Subsequently, we compute reachable sets when using the disturbance feedback controller in (3.10) for an arbitrary sampling time t_k and time interval $[t_k, t_{k+1})$. To account for the piecewise constant control law in (3.10), we first compute reachable sets for consecutive time steps of size Δt until the specified sampling time is reached. Similar to (3.5), we introduce the following recursively defined set sequence for the disturbance feedback controller in (3.10) and the model $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$:

$$\begin{aligned} \widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)) &= \left\langle c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))}, G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \right\rangle_Z \\ &= \left\langle \begin{bmatrix} c_x(t_k) \\ c_u(t_k) \end{bmatrix}, \begin{bmatrix} G_x(t_k) \\ G_u(t_k) \end{bmatrix} \right\rangle_Z \end{aligned} \tag{3.11a}$$

$$\begin{aligned} \langle c_x(t_{k+1}), G_x(t_{k+1})\rangle_Z &= \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{over}}\left(\Delta t, \widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot))\right) \\ &\overset{(2.22a)}{=} \Pi_x\left(e^{\widetilde{A}\Delta t}\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)) \oplus \widetilde{\mathcal{W}}\right), \end{aligned} \tag{3.11b}$$

where $k \in \mathbb{N}$. Similar to Theorem 3.2, we prove in the following theorem that the sets in (3.11) are over-approximating the augmented reachable sets of $\mathbf{M}$ when using the controller in (3.10).

Theorem 3.7 (Set Propagation using Disturbance Feedback Control): For all $x(t_0) \in \mathcal{Z}_x(t_0)$, applying the disturbance feedback controller in (3.10) to $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$ results in

$$\begin{bmatrix} x(t_k) \\ u(t_k) \end{bmatrix} \in \widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}\left(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right),$$

where $k \in \mathbb{N}$. ■

Proof. First, we prove that applying the disturbance feedback controller in (3.10) to $\mathbf{M}$ results in

$$\begin{bmatrix} x(t_k) \\ u(t_k) \end{bmatrix} = c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k-1} \end{bmatrix}, \tag{3.12}$$

where $\lambda_{\widetilde{\mathcal{W}},k-1} \in \mathbb{R}^{k\mathtt{gen}(\widetilde{\mathcal{W}})}$ is defined in (2.25). We proceed by induction:

Base case: For $k = 0$, we obtain

$$\begin{aligned}
\begin{bmatrix} x(t_0) \\ u(t_0) \end{bmatrix} &\overset{(2.25),(3.2),(3.10)}{=} \begin{bmatrix} c_x(t_0) \\ c_u(t_0) \end{bmatrix} + \begin{bmatrix} G_x(t_0) \\ G_u(t_0) \end{bmatrix} \lambda_{x,0} \\
&\overset{(3.11a)}{=} c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_0,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_0,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \lambda_{x,0} \\
&\overset{(2.25)}{=} c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_0,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_0,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},-1} \end{bmatrix}.
\end{aligned}$$

Induction hypothesis: (3.12) holds for some arbitrary $k \in \mathbb{N}$.

Induction step: For $k+1$, the state is

$$\begin{aligned}
x(t_{k+1}) &\overset{(2.19),(2.23a)}{=} \Pi_x \left(e^{\widetilde{A}\Delta t} \begin{bmatrix} x(t_k) \\ u(t_k) \end{bmatrix} + \widetilde{w}(t_k) \right) \\
&\overset{(3.12)}{=} \Pi_x \left(e^{\widetilde{A}\Delta t} \left(c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k-1} \end{bmatrix} \right) + \widetilde{w}(t_k) \right) \\
&\overset{(2.23b)}{=} \Pi_x \Bigg(e^{\widetilde{A}\Delta t} c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + e^{\widetilde{A}\Delta t} G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k-1} \end{bmatrix} \\
&\qquad + c_{\widetilde{\mathcal{W}}} + G_{\widetilde{\mathcal{W}}} \lambda_{\widetilde{w},k} \Bigg) \\
&\overset{(2.25)}{=} \Pi_x \left(e^{\widetilde{A}\Delta t} c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + c_{\widetilde{\mathcal{W}}} + \begin{bmatrix} e^{\widetilde{A}\Delta t} G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_k,\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} & G_{\widetilde{\mathcal{W}}} \end{bmatrix} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k} \end{bmatrix} \right) \\
&\overset{(2.10a),(2.19),(3.11a),(3.11b)}{=} \Pi_x \left(c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k} \end{bmatrix} \right),
\end{aligned}$$

where a not necessarily unique $\lambda_{\widetilde{w},k} \in \mathbb{R}^{\mathtt{gen}\left(\widetilde{\mathcal{W}}\right)}$ with $\left|\lambda_{\widetilde{w},k}\right| \leq \mathbf{1}$ is guaranteed to exist because $\widetilde{w}(t_k) \in \widetilde{\mathcal{W}}$. Similarly, the input at time step $k+1$ is

$$\begin{aligned} u(t_{k+1}) &\overset{(3.10)}{=} c_u(t_{k+1}) + G_u(t_{k+1}) \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k} \end{bmatrix} \\ &\overset{(2.10a),(2.19),(3.11a),(3.11b)}{=} \Pi_u \left(c_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} + G_{\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(t_{k+1},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot))} \begin{bmatrix} \lambda_{x,0} \\ \lambda_{\widetilde{\mathcal{W}},k} \end{bmatrix} \right), \end{aligned}$$

which completes the proof of (3.12). It also becomes clear from this proof that the size of $G_u(t_{k+1}) \in \mathbb{R}^{n_u \times \left(\mathtt{gen}(\mathcal{Z}_x(t_0)) + (k+1)\mathtt{gen}\left(\widetilde{\mathcal{W}}\right)\right)}$ grows linearly with k to account for the Minkowski addition resulting from the augmented reachable disturbance set $\widetilde{\mathcal{W}} \subset \mathbb{R}^{n_x+n_u}$.

Based on (3.12), $|\lambda_{x,0}| \leq \mathbf{1}$, $\left|\lambda_{\widetilde{\mathcal{W}},k-1}\right| \leq \mathbf{1}$, and Definition 2.6, it follows that the sets in (3.11) are over-approximating the augmented reachable sets of $\mathbf{M}$ when using the controller in (3.10). □

We have focused on performing reachability analysis for discrete sampling times when using the disturbance feedback controller in (3.10). Nevertheless, the state and input constraints in (3.1) must be satisfied not only at but also between sampling times. Thus, similar to (3.8), we compute reachable sets for an arbitrary time interval $[t_k, t_{k+1})$ according to

$$\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}\left([t_k, t_{k+1}), \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) = \widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{over}}\left([0, \Delta t), \widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}\left(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right)\right).$$

Then, the projection of the over-approximative reachable set onto the original state and input space is obtained by $\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}\left(\cdot, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right)$ and $\Pi_u \widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}\left(\cdot, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right)$, respectively.

In summary, we can efficiently compute the set of states and inputs that are reachable for all $x(t_0) \in \mathcal{Z}_x(t_0)$ when applying the disturbance feedback controller in (3.10) to $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$. In the following section, we use the presented reachable set computations for both feedback controllers to construct safe sets with minimum volume, i.e., safe sets that are as small as possible.

3.4 Small Safe Sets

In this section, we present three approaches to compute safe sets for $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$ that are as small as possible. In Subsection 3.4.1, we determine a small safe set using a simplified state feedback controller, i.e., by setting the correction input zonotope to zero. As a result, no optimization problem must be solved, which renders this simple approach very efficient. In Subsection 3.4.2, we compute a small sampled-data RCI set by solving a COP. Finally, we construct small safe sets using disturbance feedback control in Subsection 3.4.3.

3.4.1 Simplified State Feedback Control

By setting $\langle c_u(t_k), G_u(t_k)\rangle_Z = \{\mathbf{0}\}$ in (3.4) for all $k \in \mathbb{N}$, we obtain the simplified piecewise constant state feedback controller

$$u(t) = Kx(t_k) \quad \text{for } t \in [t_k, t_{k+1}). \tag{3.13}$$

In the following lemma, we provide conditions for a set $\mathcal{S}_{Kx} \subset \mathbb{R}^{n_x}$ to be a safe set when using the simplified controller in (3.13), i.e., the state and input constraints in (3.1) are satisfied for all $x(t_0) \in \mathcal{S}_{Kx}$ despite the presence of disturbances.

Lemma 3.8 (Safe Set using (3.13)): Let $\mathbf{0} \in \mathcal{X}$, $\mathbf{0} \in \mathcal{U}$, and let there exist a set $\mathcal{S}_{Kx} \subseteq \mathcal{X}$ and a corresponding time step $k_{Kx} \in \mathbb{N}_{>0}$ such that

$$\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{k_{Kx}}, \mathcal{S}_{Kx}, \{\mathbf{0}\}\right) \subseteq \mathcal{S}_{Kx} \tag{3.14a}$$

$$\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_k, t_{k+1}), \mathcal{S}_{Kx}, \{\mathbf{0}\}\right) \subseteq \mathcal{X} \quad \text{for } k \in \mathbb{N}_{[0,k_{Kx}-1]} \tag{3.14b}$$

$$\Pi_u \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_k, t_{k+1}), \mathcal{S}_{Kx}, \{\mathbf{0}\}\right) \subseteq \mathcal{U} \quad \text{for } k \in \mathbb{N}_{[0,k_{Kx}-1]}. \tag{3.14c}$$

Then, $\mathcal{S}_{Kx}$ with corresponding safety-preserving controller in (3.13) is a safe set for $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$ and, thus, $\mathcal{S}_{Kx}$ is an over-approximation of the corresponding DT minimal robust positively invariant (mRPI) set. ■

Proof. Safe set: The condition in (3.14a) ensures that all states starting in $\mathcal{S}_{Kx}$ can be steered into $\mathcal{S}_{Kx}$ in k_{Kx} steps. However, in contrast to invariant sets, the CT state trajectory $x(\cdot)$ might leave $\mathcal{S}_{Kx}$ during $[0, t_{k_{Kx}})$. Nevertheless, during this time, the state and input constraints in (3.1) are always fulfilled because of (3.14b) and (3.14c). Consequently, it follows by induction that robust constraint satisfaction for all $x(t_0) \in \mathcal{S}_{Kx}$ is achieved for an infinite time horizon when applying the controller in (3.13) to $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$. Therefore, $\mathcal{S}_{Kx}$ is a safe set.

Over-approximation of DT mRPI set: Subsequently, we show by contradiction that $\mathcal{S}_{Kx}$ is an over-approximation of the DT mRPI set $\mathcal{S}_{\mathrm{mRPI}} = \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(\infty, \{\mathbf{0}\}, \{\mathbf{0}\})$ [102], i.e., we assume that $\mathcal{S}_{\mathrm{mRPI}} \nsubseteq \mathcal{S}_{Kx}$. Because $\mathbf{0} \in \mathcal{W}$ and $\mathbf{0} \in \mathcal{X}$ by assumption, we know that $\mathbf{0} \in \mathcal{S}_{\mathrm{mRPI}}$ and $\mathbf{0} \in \mathcal{S}_{Kx}$. As mentioned in Section 2.4, $\mathcal{S}_{\mathrm{mRPI}}$ is the limit set for all state trajectories at sampling times when using the controller in (3.13) [37, Rmk. 4.1]. Therefore, there exists a disturbance sequence that steers the state sequence starting at $\mathbf{0}$ to any point in $\mathcal{S}_{\mathrm{mRPI}}$ and enforces the state to stay at this point. Because $\mathcal{S}_{\mathrm{mRPI}} \nsubseteq \mathcal{S}_{Kx}$, there exists a state sequence starting in $\mathcal{S}_{Kx}$ that leaves $\mathcal{S}_{Kx}$ and never returns to $\mathcal{S}_{Kx}$, which contradicts $\mathcal{S}_{Kx}$ being a safe set. As a result, the assumption $\mathcal{S}_{\mathrm{mRPI}} \nsubseteq \mathcal{S}_{Kx}$ is wrong, which shows that $\mathcal{S}_{Kx}$ is an over-approximation of the DT mRPI set $\mathcal{S}_{\mathrm{mRPI}}$. □

Because the safe set $\mathcal{S}_{Kx}$ can be safely steered into itself in k_{Kx} steps, it is also known as k_{Kx}-step recurrent set [136]. Similarly, invariant sets are also known as one-step recurrent sets. Although the union $\bigcup_{k=0}^{k_{Kx}-1} \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(t_k, \mathcal{S}_{Kx}, \{\mathbf{0}\}) \subseteq \mathcal{X}$ is an RCI set, its complex set representation typically prohibits its use in safe set applications. Subsequently, we propose a scalable algorithm to construct safe sets when using the safety-preserving controller in (3.13).

We now present Alg. 3.1 to compute a small safe set $\mathcal{S}_{Kx} \subset \mathbb{R}^{n_x}$ along with a corresponding time step $k_{Kx} \in \mathbb{N}_{>0}$. The algorithm has the following six inputs: $\mathbf{M}$, K, $\mathcal{X}$, $\mathcal{U}$, $\mathcal{W}$, and the convergence tolerance $\epsilon \in \mathbb{R}_{>0}$, which is usually chosen close to 0. Alg. 3.1 proceeds in two steps: First, two zonotope sequences are computed that converge to an over-approximation of the DT mRPI set [102]. Second, the zonotope order of this over-approximation is reduced as much as possible while ensuring that the conditions in (3.14b) and (3.14c) are satisfied. This second step reduces the complexity of subsequent computations involving $\mathcal{S}_{Kx}$. Subsequently, we describe both steps of Alg. 3.1 in more detail.

Algorithm 3.1 Small safe set using simplified state feedback control

```
Input: $\mathbf{M}, K, \mathcal{X}, \mathcal{U}, \mathcal{W}, \epsilon$
Output: $\mathcal{S}_{Kx}, k_{Kx}$
$\mathcal{X} \leftarrow \texttt{interval}\,(\mathcal{X})$
$k \leftarrow 1$
$\mathcal{Z}_{\{\mathbf{0}\},k} \leftarrow \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\,(t_k, \{\mathbf{0}\}, \{\mathbf{0}\})$
$\mathcal{Z}_{\mathcal{X},k} \leftarrow \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\,(t_k, \mathcal{X}, \{\mathbf{0}\})$
while $\epsilon \leq \texttt{dist}\left(\mathcal{Z}_{\mathcal{X},k}, \texttt{interval}\left(\mathcal{Z}_{\{\mathbf{0}\},k}\right)\right)$ do      ▷ converge to DT mRPI set
    $k \leftarrow k + 1$
    $\mathcal{Z}_{\{\mathbf{0}\},k} \leftarrow \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\,(t_k, \{\mathbf{0}\}, \{\mathbf{0}\})$
    $\mathcal{Z}_{\mathcal{X},k} \leftarrow \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\,(t_k, \mathcal{X}, \{\mathbf{0}\})$
end while
$k_{Kx} \leftarrow k$
$\mathcal{S}_{Kx} \leftarrow \emptyset$
$o_{Kx} \leftarrow 0$
while $o_{Kx} < \texttt{order}\,(\mathcal{Z}_{\mathcal{X},k_{Kx}})$ do      ▷ find smallest safe zonotope order
    $o_{Kx} \leftarrow o_{Kx} + 1$
    $\mathcal{S}_{Kx} \leftarrow \texttt{reduce}\,(\mathcal{Z}_{\mathcal{X},k_{Kx}}, o_{Kx})$
    if (3.14b) and (3.14c) are satisfied for $\mathcal{S}_{Kx}, k_{Kx}, \mathbf{M}, K, \mathcal{X}, \mathcal{U}$ then
        break
    else
        $\mathcal{S}_{Kx} \leftarrow \emptyset$
    end if
end while
```

In line 1 of Alg. 3.1, we compute the smallest multidimensional interval enclosure $\texttt{interval}\,(\mathcal{X})$ of the polytopic state constraint set $\mathcal{X}$ and express it in generator representation using (2.5). These computations are performed because we compute its reachable sets and our reachability analysis in Section 2.3 requires the initial set to be expressed in generator representation. In lines 2 to 9, we compute the set of reachable states for consecutive time steps corresponding to the following two initial state sets, namely, the origin $\{\mathbf{0}\}$ and the over-approximated state constraint set $\texttt{interval}\,(\mathcal{X})$. We denote these zonotope sequences by $\mathcal{Z}_{\{\mathbf{0}\},(\cdot)}$ and $\mathcal{Z}_{\mathcal{X},(\cdot)}$. Because the state feedback matrix of the simplified controller in (3.13) is stabilizing, $\mathcal{Z}_{\{\mathbf{0}\},(\cdot)}$ and $\mathcal{Z}_{\mathcal{X},(\cdot)}$ would converge to the DT mRPI set in the Hausdorff distance as time goes to infinity [102], if no over-approximation of reachable sets to reduce computational complexity was used. To achieve low computation times, we use an easily computable convergence criterion in line 5 based on the directed Hausdorff distance in (2.8g). Instead of directly computing $\texttt{dist}\left(\mathcal{Z}_{\mathcal{X},k}, \mathcal{Z}_{\{\mathbf{0}\},k}\right)$ using (2.15), we use multidimensional interval enclosures because they can be obtained by (2.7) and transformed to half-space representation by (2.6) such that the simple zonotope containment condition in (2.14) can be applied.

To reduce the complexity of the subsequent computations, we want the safe set $\mathcal{S}_{Kx} \supseteq \mathcal{Z}_{\mathcal{X},k_{Kx}}$ to have a reduced zonotope order compared to $\mathcal{Z}_{\mathcal{X},k_{Kx}}$ [84, 85]. Thus, in lines 13 to 21 of Alg. 3.1, we increment $\texttt{order}\,(\mathcal{S}_{Kx})$ starting from 1 until the conditions in (3.14b) and (3.14c) are satisfied eventually. However, if these two conditions are even violated for the tight over-approximation $\mathcal{Z}_{\mathcal{X},k_{Kx}}$ of the DT mRPI set, Alg. 3.1 returns an empty set as first output.

Proposition 3.9 (Properties of Alg. 3.1): Let $\mathbf{0} \in \mathcal{X}$, $\mathbf{0} \in \mathcal{U}$, and let the first output $\mathcal{S}_{Kx} \subset \mathbb{R}^{n_x}$ of Alg. 3.1 be nonempty. Then, $\mathcal{S}_{Kx}$ is a safe set with the second output $k_{Kx} \in \mathbb{N}_{>0}$ being a corresponding time step, i.e., they satisfy the safe set conditions in (3.14). ■

Proof. Because $\mathcal{S}_{Kx}$ is assumed to be a nonempty set, we know that the check in line 16 of Alg. 3.1 has been passed successfully. Therefore, the conditions in (3.14b) and (3.14c) are fulfilled for $\mathcal{S}_{Kx}$ and k_{Kx}. Because (3.14b) is satisfied, it follows that

$$\mathcal{S}_{Kx} \subseteq \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_0, t_1), \mathcal{S}_{Kx}, \{\mathbf{0}\}\right) \subseteq \mathcal{X}.$$

This relation results in

$$\begin{aligned}\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{k_{Kx}}, \mathcal{S}_{Kx}, \{\mathbf{0}\}\right) &\subseteq \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{k_{Kx}}, \mathcal{X}, \{\mathbf{0}\}\right)\\ &\subseteq \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{k_{Kx}}, \texttt{interval}\,(\mathcal{X}), \{\mathbf{0}\}\right)\\ &\overset{\text{line 15 of Alg. 3.1}}{\subseteq} \mathcal{S}_{Kx},\end{aligned}$$

which shows the satisfaction of (3.14a). Thus, all conditions in (3.14) are satisfied. Therefore, $\mathcal{S}_{Kx}$ is a safe set with k_{Kx} being a corresponding time step. □

For our subsequent computations, we assume that $\mathcal{S}_{Kx}$ obtained by executing Alg. 3.1 is nonempty. This assumption is also widely used in robust MPC [41–43, 160], as discussed in more detail in Chapter 4. Thus, if this assumption is violated, a different stabilizing state feedback matrix K might be required, e.g., obtained by using LQR-based controller synthesis with different weighting matrices [155]. Similarly, the convergence tolerance $\epsilon \in \mathbb{R}_{>0}$ could be decreased to satisfy this assumption.

Instead of using Alg. 3.1 to construct a small safe set, Alg. 2.1 can also be used to obtain a tight RPI over-approximation of the DT mRPI set. However, due to the involved Minkowski additions in line 8 of Alg. 2.1, the number of generators of the resulting zonotope increases with the number of performed Minkowski additions. To bound the complexity of the safe set representation, zonotope order reduction operations are required. However, performing these set operations invalidates the robust invariance guarantees of Alg. 2.1. Thus, Alg. 3.1 offers an efficient way to compute a small safe set while bounding the complexity for representing this safe set. Subsequently, we discuss potential computational speed-ups of Alg. 3.1.

To determine the time step k_{Kx} in Alg. 3.1, we use the two converging zonotope sequences $\mathcal{Z}_{\{\mathbf{0}\},(\cdot)}$ and $\mathcal{Z}_{\mathcal{X},(\cdot)}$. Alternatively, we could also use only $\mathcal{Z}_{\mathcal{X},(\cdot)}$ and terminate if the distance between two consecutive zonotopes $\mathcal{Z}_{\mathcal{X},k}$ and $\mathcal{Z}_{\mathcal{X},k+1}$ of this sequence is below the convergence tolerance for some $k \in \mathbb{N}$. By exploiting the superposition principle, we could alternatively also compute the sequence $\Pi_x \widetilde{\mathcal{R}}_{Kx}^{(A,B,\{\mathbf{0}\},\mathrm{CT})}\left(t_{(\cdot)}, \texttt{interval}\left(\mathcal{X}\right), \{\mathbf{0}\}\right)$ and terminate if the distance between a set of this sequence and the origin is below the convergence tolerance. Although both alternatives offer a potential computational speed-up, we opt for Alg. 3.1 because its steps are easy to follow and the computation time is typically negligible.

In summary, if the initial state $x(t_0)$ lies within the safe set $\mathcal{S}_{Kx}$, the simplified controller in (3.13) ensures robust constraint satisfaction for an infinite time horizon. When using the simplified controller in (3.13), a stabilizing state feedback matrix K is assumed to be given such that $\mathcal{S}_{Kx}$ obtained by executing Alg. 3.1 is nonempty. Although this is also a widely used assumption in robust MPC [41–43, 160], it is unclear how to systematically and efficiently find such a K. In particular, if both constraint sets $\mathcal{X}$ and $\mathcal{U}$ are small, it usually becomes challenging to find a safe set at all. In the following subsection, we propose a robust control method that addresses these issues.

3.4.2 Robust Control Invariance

In this subsection, we present an approach for computing a small RCI set. Thus, no stabilizing state feedback matrix K is required in contrast to the presented state feedback control method in Subsection 3.4.1. In addition, the assumption that both constraint sets $\mathcal{X}$ and $\mathcal{U}$ contain the origin is no longer necessary, which offers more flexibility of this approach compared to the state feedback control method.

In [147], a COP is presented to obtain a small RCI set for a DT system. This set is obtained by directly minimizing a matrix norm over the generator matrix $G_x(t_0) \in \mathbb{R}^{n_x \times \texttt{gen}(\mathcal{Z}_x(t_0))}$ of the initial state set $\mathcal{Z}_x(t_0) = \langle c_x(t_0), G_x(t_0)\rangle_Z$ and incrementing the number of generators $\texttt{gen}\left(\mathcal{Z}_x(t_0)\right) \in \mathbb{N}_{>0}$ until feasibility of the optimization problem is detected. Inspired by this approach, we solve a COP to compute a small RCI set for the model $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$.

In particular, let $s_{\mathcal{X}}^\star$, $\langle c_x^\star(t_0), G_x^\star(t_0)\rangle_Z$, $\mathcal{Z}_u^\star(t_0)$ be the solution of the COP

$$\begin{aligned}
&\underset{s_{\mathcal{X}},\langle c_x(t_0),G_x(t_0)\rangle_Z,\mathcal{Z}_u(t_0)}{\text{minimize}} && s_{\mathcal{X}} && \text{(3.15a)}\\
&\text{subject to} && 0 < s_{\mathcal{X}} && \text{(3.15b)}\\
& && \langle c_x(t_0), G_x(t_0)\rangle_Z \subseteq \texttt{scalePolytope}(\mathcal{X}, s_{\mathcal{X}}, \mathring{x}) && \text{(3.15c)}\\
& && \langle c_x(t_1), G_x(t_1)\rangle_Z = \Pi_x \widetilde{\mathcal{R}}_{Lw}^{\mathbf{M}}\left(\Delta t, \langle c_x(t_0), G_x(t_0)\rangle_Z, \mathcal{Z}_u(t_0)\right) && \text{(3.15d)}\\
& && \begin{bmatrix} c_x(t_1) & G_x(t_1)\end{bmatrix} = \begin{bmatrix} c_x(t_0) & \mathbf{0} & G_x(t_0)\end{bmatrix} && \text{(3.15e)}\\
& && \Pi_x \widetilde{\mathcal{R}}_{Lw}^{\mathbf{M}}\left([0,\Delta t), \langle c_x(t_0), G_x(t_0)\rangle_Z, \mathcal{Z}_u(t_0)\right) \subseteq \mathcal{X} && \text{(3.15f)}\\
& && \Pi_u \widetilde{\mathcal{R}}_{Lw}^{\mathbf{M}}\left([0,\Delta t), \langle c_x(t_0), G_x(t_0)\rangle_Z, \mathcal{Z}_u(t_0)\right) \subseteq \mathcal{U}, && \text{(3.15g)}
\end{aligned}$$

where $\mathring{x} \in \mathcal{X}$ is a scaling point, e.g., the Chebyshev center $\texttt{center}(\mathcal{X})$ of $\mathcal{X}$. Because only a single time step is considered, solving (3.15) by using the operator $\widetilde{\mathcal{R}}_{Kx}^{\mathbf{M}}(\cdot,\cdot,\cdot)$ with any stabilizing K instead of $\widetilde{\mathcal{R}}_{Lw}^{\mathbf{M}}(\cdot,\cdot,\cdot)$ results in the same optimal RCI set $\mathcal{Z}_x^\star(t_0) = \langle c_x^\star(t_0), G_x^\star(t_0)\rangle_Z$ but different $\mathcal{Z}_u^\star(t_0)$. If the COP in (3.15) is feasible for a given number of generators $\texttt{gen}(\mathcal{Z}_x^\star(t_0)) \in \mathbb{N}_{>0}$, $\mathcal{Z}_x^\star(t_0)$ is a sampled-data RCI set, as shown in the following proposition.

Proposition 3.10 (Properties of (3.15)): Let $s_{\mathcal{X}}^\star$, $\mathcal{Z}_x^\star(t_0) = \langle c_x^\star(t_0), G_x^\star(t_0)\rangle_Z$, $\mathcal{Z}_u^\star(t_0)$ be the solution of (3.15) for a given number of generators $\texttt{gen}(\mathcal{Z}_x^\star(t_0)) \in \mathbb{N}_{>0}$. Then, $\mathcal{Z}_x^\star(t_0)$ is an RCI set, (3.15) is also feasible for $\texttt{gen}(\mathcal{Z}_x^\star(t_0)) + i$ with $i \in \mathbb{N}$, and the cost in (3.15a) is monotonically decreasing with an increasing number of generators. ■

Proof. RCI set: If the constraint in (3.15e) is satisfied, both zonotopes $\langle c_x(t_1), G_x(t_1)\rangle_Z$ and $\left\langle c_x(t_0), \begin{bmatrix}\mathbf{0} & G_x(t_0)\end{bmatrix}\right\rangle_Z$ represent the same set. Because horizontally concatenating the generator matrix of any zonotope with $\mathbf{0}$ leaves the set unchanged, $\left\langle c_x(t_0), \begin{bmatrix}\mathbf{0} & G_x(t_0)\end{bmatrix}\right\rangle_Z$ and $\langle c_x(t_0), G_x(t_0)\rangle_Z$ also describe the same set. Thus, the constraint in (3.15e) ensures that $\langle c_x(t_1), G_x(t_1)\rangle_Z \subseteq \langle c_x(t_0), G_x(t_0)\rangle_Z$, which implies set invariance. In addition, the constraints in (3.15f) and (3.15g) enforce robust constraint satisfaction during the time interval $[0, \Delta t)$.

Recursive feasibility: Let $\langle c_u^\star(t_0), G_u^\star(t_0)\rangle_Z = \mathcal{Z}_u^\star(t_0)$. Subsequently, we show that $s_{\mathcal{X}}^{\star,+}$, $\left\langle c_x^{\star,+}(t_0), G_x^{\star,+}(t_0)\right\rangle_Z$, $\left\langle c_u^{\star,+}(t_0), G_u^{\star,+}(t_0)\right\rangle_Z$ is a solution for $\texttt{gen}(\mathcal{Z}_x^\star(t_0)) + i$ with $i \in \mathbb{N}$, where $s_{\mathcal{X}}^{\star,+} = s_{\mathcal{X}}^\star$, $c_x^{\star,+}(t_0) = c_x^\star(t_0)$, $G_x^{\star,+}(t_0) = \begin{bmatrix}\mathbf{0} & G_x^\star(t_0)\end{bmatrix}$, $c_u^{\star,+}(t_0) = c_u^\star(t_0)$, and $G_u^{\star,+}(t_0) = \begin{bmatrix}\mathbf{0} & G_u^\star(t_0)\end{bmatrix}$. Because $\left\langle c_x^{\star,+}(t_0), G_x^{\star,+}(t_0)\right\rangle_Z = \langle c_x^\star(t_0), G_x^\star(t_0)\rangle_Z$ and $\left\langle c_u^{\star,+}(t_0), G_u^{\star,+}(t_0)\right\rangle_Z = \langle c_u^\star(t_0), G_u^\star(t_0)\rangle_Z$, the constraint satisfaction follows directly from the set propagation in (3.11) and from the feasibility for $\texttt{gen}(\mathcal{Z}_x^\star(t_0))$.

Monotonically decreasing cost: Because the optimization problem in (3.15) is convex and recursively feasible, the cost in (3.15a) is monotonically decreasing with an increasing number of generators of $\mathcal{Z}_x^\star(t_0)$. □

Based on Proposition 3.10, obtaining a small sampled-data RCI set can be achieved by increasing the number of generators $\texttt{gen}(\mathcal{Z}_x(t_0))$ of $\mathcal{Z}_x(t_0)$ starting from 1 until (3.15) is feasible. However, it is unclear how much $\texttt{gen}(\mathcal{Z}_x(t_0))$ must be increased to enable feasibility of (3.15), similar to [138, 145, 147]. Thus, the generator matrix of the RCI set obtained by solving (3.15) can have a lot of columns, which results in a large zonotope order and limits its use in efficient robust control approaches. In addition, using the constraint in (3.15e) instead of $\left\langle \begin{bmatrix} c_x(t_1) & G_x(t_1) \end{bmatrix} \right\rangle_Z \subseteq \left\langle \begin{bmatrix} c_x(t_0) & \mathbf{0} & G_x(t_0) \end{bmatrix} \right\rangle_Z$ enables convexity of the optimization problem on the one hand but also results in an increased conservativeness on the other hand. In the following subsection, we present an optimization-based approach using disturbance feedback control to compute a small safe set with a fixed zonotope order.

3.4.3 Disturbance Feedback Control

In this subsection, we use the disturbance feedback controller in (3.10) to synthesize small safe sets. By parameterizing the controller as an affine map of the past disturbances, we solve a COP to optimize over the disturbance feedback matrices [156]. Thus, we are more flexible when searching for a safety-preserving controller than the simplified controller in (3.13), which typically comes at the cost of increased computation times.

To compute a small safe set whose zonotope order is $o_{x,0} \in \mathbb{N}_{>0}$, we solve a COP that directly considers the shape of the state constraint set $\mathcal{X}$ and uses the generator scaling framework [143], i.e., we fix the arbitrary orientations of the generators of $\mathcal{Z}_x(t_0)$ and optimize only their scaling factors. In particular, let $s_{\mathcal{X}}^\star$, $s_{x,0}^\star$, $c_x^\star(t_0)$, $\mathcal{Z}_u^\star(\cdot)$ be the solution of the COP

$$\underset{s_{\mathcal{X}}, s_{x,0}, c_x(t_0), \mathcal{Z}_u(\cdot)}{\text{minimize}} \quad s_{\mathcal{X}} \tag{3.16a}$$

$$\text{subject to} \quad \mathbf{0} < s_{x,0} \tag{3.16b}$$

$$\mathcal{Z}_x(t_0) = \langle c_x(t_0), G_{\text{fixed}} \texttt{diag}(s_{x,0}) \rangle_Z \tag{3.16c}$$

$$\Pi_x \widetilde{\mathcal{R}}_{Lw}^{\mathbf{M}} \left(t_{k_{x,0}}, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \mathcal{Z}_x(t_0) \tag{3.16d}$$

$$0 < s_{\mathcal{X}} \tag{3.16e}$$

$$\mathcal{Z}_x(t_0) \subseteq \texttt{scalePolytope}(\mathcal{X}, s_{\mathcal{X}}, \mathring{x}) \tag{3.16f}$$

$$\Pi_x \widetilde{\mathcal{R}}_{Lw}^{\mathbf{M}} \left([t_k, t_{k+1}), \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \mathcal{X} \quad \text{for } k \in \mathbb{N}_{[0, k_{x,0}-1]} \tag{3.16g}$$

$$\Pi_u \widetilde{\mathcal{R}}_{Lw}^{\mathbf{M}} \left([t_k, t_{k+1}), \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \mathcal{U} \quad \text{for } k \in \mathbb{N}_{[0, k_{x,0}-1]}, \tag{3.16h}$$

where $\mathring{x} \in \mathcal{X}$ is a scaling point, $k_{x,0} \in \mathbb{N}_{>0}$ is the initial time step, $s_{x,0} \in \mathbb{R}_{>0}^{n_x o_{x,0}}$ is a generator scaling vector, and $G_{\text{fixed}} \in \mathbb{R}^{n_x \times n_x o_{x,0}}$ is a fixed generator matrix. Although any convex cost function can be used in (3.16a), we choose the linear cost for simplicity. Before presenting our heuristic for obtaining promising generator directions of the small safe set, i.e., promising columns of G_{fixed}, we prove in the following lemma that $\left\langle c_x^\star(t_0), G_{\text{fixed}} \texttt{diag}(s_{x,0}^\star) \right\rangle_Z \subset \mathbb{R}^{n_x}$ is a safe set.

Lemma 3.11 (Properties of (3.16)): Let $s_{\mathcal{X}}^\star$, $s_{x,0}^\star$, $c_x^\star(t_0)$, $\mathcal{Z}_u^\star(\cdot)$ be the solution of (3.16). Then, $\mathcal{Z}_x^\star(t_0) = \left\langle c_x^\star(t_0), G_{\text{fixed}} \texttt{diag}(s_{x,0}^\star) \right\rangle_Z$ is a safe set. If, in addition to the feasibility of (3.16), the initial time step $k_{x,0} = 1$, $\mathcal{Z}_x^\star(t_0)$ is an RCI set. ■

Proof. Safe set: The proof is similar to Lemma 3.8 and is presented subsequently for the sake of completeness. The constraint in (3.16d) ensures that the CT state trajectory $x(\cdot)$ starting in $\mathcal{Z}_x^\star(t_0)$ ends in $\mathcal{Z}_x^\star(t_0)$ at $t_{k_{x,0}}$. Because $x(\cdot)$ might leave $\mathcal{Z}_x^\star(t_0)$ during the time interval $[0, t_{k_{x,0}})$, the constraints in (3.16g) and (3.16h) enforce robust constraint satisfaction during $[0, t_{k_{x,0}})$. By induction, it follows that robust constraint satisfaction can be achieved for an infinite time horizon if $x(t_0) \in \mathcal{Z}_x^\star(t_0)$, which implies that $\mathcal{Z}_x^\star(t_0)$ is a safe set.

RCI set: If, in addition to the feasibility of (3.16), the initial time step $k_{x,0} = 1$, (3.16d) enforces the projected reachable set of $\mathcal{Z}_x^\star(t_0)$ to lie within $\mathcal{Z}_x^\star(t_0)$ at Δt. Thus, $\mathcal{Z}_x^\star(t_0)$ is an RCI set. □

To enable feasibility of (3.16), it is crucial to choose a suitable fixed generator matrix $G_{\text{fixed}} \in \mathbb{R}^{n_x \times n_x o_{x,0}}$. Subsequently, we present a simple two-step approach to obtain promising generator directions: First, for a given initial state set $\mathcal{Z}_x(t_0) \subset \mathbb{R}^{n_x}$, let $s_{\mathcal{X}}^\star$, $\mathcal{Z}_u^\star(\cdot)$ be the solution of the COP

$$
\begin{aligned}
\underset{s_{\mathcal{X}}, \mathcal{Z}_u(\cdot)}{\text{minimize}} \quad & s_{\mathcal{X}} && (3.17a)\\
\text{subject to} \quad & 0 < s_{\mathcal{X}} && (3.17b)\\
& \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}\left(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \texttt{scalePolytope}(\mathcal{X}, s_{\mathcal{X}}, \mathring{x}) \quad \text{for } k \in \mathbb{N}_{[0,k_{x,0}]} && (3.17c)\\
& \Pi_u \widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}\left(t_k, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \mathcal{U} \quad \text{for } k \in \mathbb{N}_{[0,k_{x,0}]}, && (3.17d)
\end{aligned}
$$

i.e., we minimize the expansion of the reachable sets around the scaling point $\mathring{x} \in \mathcal{X}$. We also want to mention that $k_{x,0}$ in (3.17) can be different compared to (3.16). Second, we compute the zonotope

$$
\langle c_{\text{fixed}}, G_{\text{fixed}} \rangle_Z = \texttt{reduce}\left(\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}\left(t_{k_{x,0}}, \mathcal{Z}_x(t_0), \mathcal{Z}_u^\star(\cdot)\right), o_{x,0}\right) \tag{3.18}
$$

of reduced order $o_{x,0} \in \mathbb{N}_{>0}$, whose generator matrix is the desired G_{fixed}.

Finally, we propose Alg. 3.2 to compute a small safe set by combining (3.16) to (3.18). Essentially, we iteratively update G_{fixed} until the feasibility of (3.16) is detected eventually. If feasibility is not obtained, the two inputs $o_{x,0} \in \mathbb{N}_{>0}$ and $k_{x,0} \in \mathbb{N}_{>0}$ of Alg. 3.2 can be increased or a more suitable scaling point $\mathring{x} \in \mathcal{X}$ can be selected. Subsequently, we describe the main steps of Alg. 3.2 in more detail.

Algorithm 3.2 Small safe set using disturbance feedback control

```
Input: $\mathbf{M}, \mathcal{X}, \mathcal{U}, o_{x,0}, k_{x,0}, \mathring{x}$
Output: $\mathcal{Z}_x^\star(t_0), \mathcal{Z}_u^\star(\cdot)$
1: $\langle c_{\text{fixed}}, G_{\text{fixed}} \rangle_Z \leftarrow \{\mathring{x}\}$
2: $\mathcal{Z}_x^\star(t_0) \leftarrow \emptyset$
3: while $\mathcal{Z}_x^\star(t_0) \equiv \emptyset$ do
4:     $\mathcal{Z}_x(t_0) \leftarrow \langle c_{\text{fixed}}, G_{\text{fixed}} \rangle_Z$
5:     $s_{\mathcal{X}}^\star, \mathcal{Z}_u^\star(\cdot) \leftarrow$ solve (3.17) for $k_{x,0}, \mathcal{Z}_x(t_0), \mathring{x}, \mathbf{M}, \mathcal{X}, \mathcal{U}$
6:     $\langle c_{\text{fixed}}, G_{\text{fixed}} \rangle_Z \leftarrow$ insert $k_{x,0}, \mathcal{Z}_x(t_0), \mathcal{Z}_u^\star(\cdot), o_{x,0}, \mathbf{M}$ into (3.18)
7:     $s_{\mathcal{X}}^\star, s_{x,0}^\star, c_x^\star(t_0), \mathcal{Z}_u^\star(\cdot) \leftarrow$ solve (3.16) for $k_{x,0}, \mathring{x}, G_{\text{fixed}}, \mathbf{M}, \mathcal{X}, \mathcal{U}$
8:     $\mathcal{Z}_x^\star(t_0) \leftarrow \left\langle c_x^\star(t_0), G_{\text{fixed}}\texttt{diag}(s_{x,0}^\star) \right\rangle_Z$
9: end while
```

In line 1 of Alg. 3.2, we initialize the initial state set with the scaling point $\mathring{x} \in \mathcal{X}$, which is given by, e.g., $\texttt{center}\,(\mathcal{X})$. In lines 3 to 9, we iteratively solve COPs for updated initial state sets until feasibility of (3.16) is detected eventually, i.e., until $\mathcal{Z}_x^\star(t_0)$ is nonempty. In particular, we solve the COP in (3.17) to find promising generator directions in line 5 of Alg. 3.2. In line 6, we perform a zonotope order reduction to keep the computational complexity of the COP solved in line 7 constant. Finally, we update $\mathcal{Z}_x^\star(t_0)$ in line 8.

In summary, if the initial state $x(t_0)$ lies within $\mathcal{Z}_x^\star(t_0) = \left\langle c_x^\star(t_0), G_{\text{fixed}}\texttt{diag}(s_{x,0}^\star)\right\rangle_Z$, the disturbance feedback controller in (3.10) with optimal correction input zonotope sequence $\mathcal{Z}_u^\star(\cdot)$ ensures robust constraint satisfaction for an infinite time horizon. After presenting our three approaches for computing safe sets that are as small as possible, we aim to maximize the volume of safe sets in the following section.

3.5 Large Safe Sets

In this section, we present three scalable approaches to compute safe sets for $\mathbf{M} = (A, B, \mathcal{W}, \text{CT})$ that are as large as possible. Based on the notion of robust control invariance for sampled-data systems [102], the largest safe set is the MRCI set. Because the shape of the MRCI set can be arbitrarily complex, our three approaches aim for a tight zonotopic under-approximation of this set.

In Subsection 3.5.1, we determine a large zonotopic safe set by uniformly scaling the small safe set $\mathcal{S}_{Kx}$ obtained by executing Alg. 3.1. This approach is very efficient because the optimal scaling can be determined without solving an optimization problem. In Subsection 3.5.2, we exploit the superposition principle to propose a COP with zero terminal constraint for obtaining a tight under-approximation of the MRCI set. Computing large safe sets in Subsection 3.5.3 is achieved by ensuring that the last reachable set is contained within another safe set. This approach is also closely related to the small safe set COPs proposed in Section 3.4, where we aim for the minimization instead of the maximization of the volume of the safe set.

3.5.1 Scaling of Safe Set

Typically, the safe set $\mathcal{S}_{Kx} = \langle c_{\mathcal{S}_{Kx}}, G_{\mathcal{S}_{Kx}}\rangle_Z \subseteq \mathcal{X}$ obtained by executing Alg. 3.1 is a relatively small safe set and its zonotope order is one, as it is a tight over-approximation of the DT mRPI set [102]. To maximize the region of operation for our system, we want to find a safe set that is as large as possible. Subsequently, we propose a simple approach for enlarging the volume of $\mathcal{S}_{Kx}$ in a straightforward and scalable way.

We use the simplified controller in (3.13) and solve a COP that determines the maximum scaling factor $\alpha^\star \geq 1$ that ensures $\langle c_{\mathcal{S}_{Kx}}, \alpha^\star G_{\mathcal{S}_{Kx}}\rangle_Z$ being a safe set. In particular, let $\alpha^\star$ be the solution of the COP

$$\underset{\alpha}{\text{maximize}} \quad \alpha \tag{3.19a}$$

$$\text{subject to} \quad \alpha \geq 1 \tag{3.19b}$$

$$\text{(3.14b) and (3.14c) are satisfied for } \langle c_{\mathcal{S}_{Kx}}, \alpha G_{\mathcal{S}_{Kx}}\rangle_Z, k_{Kx}, \mathbf{M}, K, \mathcal{X}, \mathcal{U}, \tag{3.19c}$$

where $\mathcal{S}_{Kx} = \langle c_{\mathcal{S}_{Kx}}, G_{\mathcal{S}_{Kx}}\rangle_Z$ and k_{Kx} are the two outputs of Alg. 3.1. Because $\mathcal{S}_{Kx}$ is assumed to be nonempty, (3.19) is always feasible by construction. In addition, $\mathcal{S}_{Kx}^\star = \langle c_{\mathcal{S}_{Kx}}, \alpha^\star G_{\mathcal{S}_{Kx}}\rangle_Z$ is a safe set based on Proposition 3.9 and $\mathcal{S}_{Kx} \subseteq \mathcal{S}_{Kx}^\star \subseteq \mathcal{X}$.

Because α is scalar and $\mathcal{X}$ is bounded, the solution of (3.19) can be alternatively obtained without solving an optimization problem. Instead, we perform a simple binary search, which is also known as logarithmic search [161]. We now present the corresponding safe set scaling method in Alg. 3.3, which has the following six inputs: the model $\mathbf{M}$, the state feedback matrix K used in (3.13), the safe set $\langle c_{\mathcal{S}_{Kx}}, G_{\mathcal{S}_{Kx}} \rangle_Z$ with corresponding k_{Kx}, the state constraint set $\mathcal{X}$, and the maximum interval radius $\epsilon \in \mathbb{R}_{>0}$, which is usually chosen close to 0. Subsequently, we describe the main steps of Alg. 3.3 in more detail.

Algorithm 3.3 Scaling of the safe set using binary search

```
Input: $\mathbf{M}, K, \langle c_{\mathcal{S}_{Kx}}, G_{\mathcal{S}_{Kx}} \rangle_Z, k_{Kx}, \mathcal{X}, \epsilon$
Output: $\mathcal{S}^\star_{Kx}$
1: $\mathcal{I} \leftarrow \left[1, \max\left\{\alpha \in \mathbb{R}_{>0} \,\middle|\, \langle c_{\mathcal{S}_{Kx}}, \alpha G_{\mathcal{S}_{Kx}} \rangle_Z \subseteq \mathcal{X}\right\}\right]$
2: while $\epsilon < \texttt{radius}(\mathcal{I})$ do
3:     if (3.14b) and (3.14c) are satisfied for $\langle c_{\mathcal{S}_{Kx}}, \texttt{center}(\mathcal{I})\, G_{\mathcal{S}_{Kx}} \rangle_Z, k_{Kx}, \mathbf{M}, K, \mathcal{X}, \mathcal{U}$ then
4:         $\mathcal{I} \leftarrow [\texttt{center}(\mathcal{I}), \texttt{max}(\mathcal{I})]$
5:     else
6:         $\mathcal{I} \leftarrow [\texttt{min}(\mathcal{I}), \texttt{center}(\mathcal{I})]$
7:     end if
8: end while
9: $\mathcal{S}^\star_{Kx} \leftarrow \langle c_{\mathcal{S}_{Kx}}, \texttt{min}(\mathcal{I})\, G_{\mathcal{S}_{Kx}} \rangle_Z$
```

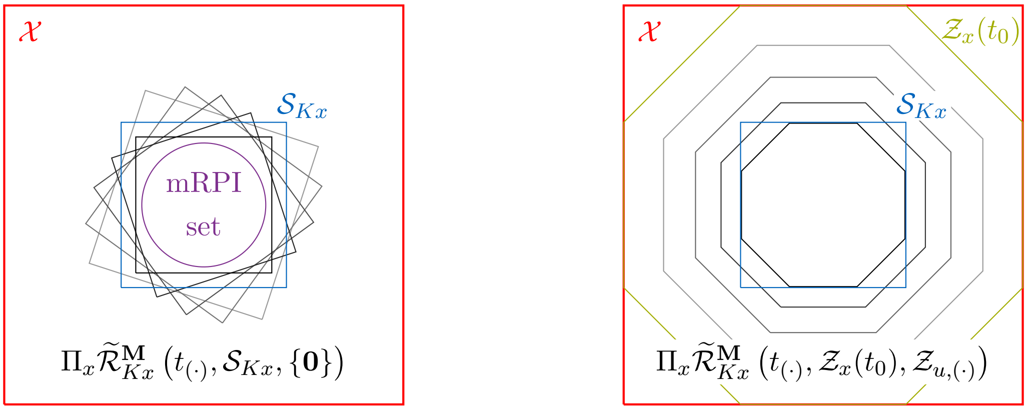
In line 1 of Alg. 3.3, we initialize an admissible one-dimensional interval $\mathcal{I} \subset \mathbb{R}$ such that (3.14) is satisfied for $\langle c_{\mathcal{S}_{Kx}}, \texttt{min}(\mathcal{I})\, G_{\mathcal{S}_{Kx}} \rangle_Z$, whereas it is generally violated for $\langle c_{\mathcal{S}_{Kx}}, \texttt{max}(\mathcal{I})\, G_{\mathcal{S}_{Kx}} \rangle_Z$. In lines 2 to 8, we perform a binary search to find an admissible scaling interval, whose maximum radius is ϵ. Finally, in line 9 of Alg. 3.3, the scaled safe set is computed.

Because the scaling factor is scalar, the shape of the scaled safe set is unchanged compared to $\mathcal{S}_{Kx}$, which can produce conservative results. In the following subsection, we present an approach to simultaneously optimize the shape of a large safe set and the controller, providing more flexibility.

3.5.2 Zero Terminal Constraint

The simplified controller in (3.13) guarantees robust constraint satisfaction for an infinite time horizon if the state of the system lies within the safe set $\mathcal{S}_{Kx} \subset \mathbb{R}^{n_x}$, which is obtained by Alg. 3.1 and Alg. 3.3, respectively. Thus, efficiently increasing the region of operation can be achieved by ensuring the initial state set $\mathcal{Z}_x(t_0) \subset \mathbb{R}^{n_x}$ to be an under-approximation of the sampled-data $k_{x,0}$-step robust backward-reachable set for the terminal set $\mathcal{S}_{Kx}$ with $k_{x,0} \in \mathbb{N}_{>0}$, also known as the robust sampled-data capture basin [129]. Thus, if the initial state $x(t_0) \in \mathbb{R}^{n_x}$ lies within $\mathcal{Z}_x(t_0)$, it can be safely steered into $\mathcal{S}_{Kx}$ in $k_{x,0}$ steps, which implies $\mathcal{Z}_x(t_0)$ being a safe set. Therefore, the satisfaction of the safety constraints in (3.1) is also ensured if $x(t_0) \in \mathcal{Z}_x(t_0)$ despite disturbances. This two-step safe set approach is illustrated in Fig. 3.2.

Ideally, we want to maximize the volume of the initial state set $\mathcal{Z}_x(t_0)$. However, computing the volume of a general zonotope is combinatorially complex with respect to the number of columns of the generator matrix [162]. Nevertheless, in the special case of $\mathcal{Z}_x(t_0)$ being a


$\mathcal{X}$
$\mathcal{S}_{Kx}$
mRPI
set
$\Pi_x\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{(\cdot)},\mathcal{S}_{Kx},\{\mathbf{0}\}\right)$
$\mathcal{X}$
$\mathcal{Z}_x(t_0)$
$\mathcal{S}_{Kx}$
$\Pi_x\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{(\cdot)},\mathcal{Z}_x(t_0),\mathcal{Z}_{u,(\cdot)}\right)$

(a) The small safe set $\mathcal{S}_{Kx}$ can be safely steered into itself. Thus, robust constraint satisfaction for an infinite time horizon can be ensured.

(b) The large safe initial set $\mathcal{Z}_x(t_0)$ can be safely steered into $\mathcal{S}_{Kx}$ in 4 steps.

Figure 3.2: Two-step safe set approach. Projections of reachable sets $\Pi_x\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{(\cdot)},\cdot,\cdot\right)$ are shown, where a lighter gray tone corresponds to a smaller prediction horizon.

parallelotope, maximizing the determinant of the generator matrix results in the maximum volume. When constraining this generator matrix to be symmetric positive definite, the maximization can be cast and efficiently solved as a COP [55, 163]. However, restricting $\mathcal{Z}_x(t_0)$ to be a parallelotope can be conservative.

To cast the optimization of the zonotopic initial state set $\mathcal{Z}_x(t_0)$ as a COP, our approach is based on the generator scaling framework [143], which has already been used in (3.16). In particular, let $s^\star_{x,0}$, $c^\star_x(t_0)$, $\mathcal{Z}^\star_u(\cdot)$ be the solution of the COP

$$\begin{aligned}
&\underset{s_{x,0},c_x(t_0),\mathcal{Z}_u(\cdot)}{\text{maximize}} && J_{\mathcal{Z}_x(t_0)}(\mathcal{Z}_x(t_0)) && \text{(3.20a)}\\
&\text{subject to} && \mathcal{Z}_x(t_0)=\langle c_x(t_0),G_{\text{fixed}}\mathtt{diag}(s_{x,0})\rangle_Z && \text{(3.20b)}\\
& && \Pi_x\widetilde{\mathcal{R}}^{(A,B,\{\mathbf{0}\},\text{CT})}_{Kx}\left(t_{k_{x,0}},\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot)\right)=\{\mathbf{0}\} && \text{(3.20c)}\\
& && \Pi_x\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_k,t_{k+1}),\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot)\right)\subseteq\mathcal{X}\quad\text{for } k\in\mathbb{N}_{[0,k_{x,0}-1]} && \text{(3.20d)}\\
& && \Pi_u\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_k,t_{k+1}),\mathcal{Z}_x(t_0),\mathcal{Z}_u(\cdot)\right)\subseteq\mathcal{U}\quad\text{for } k\in\mathbb{N}_{[0,k_{x,0}-1]}, && \text{(3.20e)}
\end{aligned}$$

where $J_{\mathcal{Z}_x(t_0)}$ is a concave cost function, $s_{x,0}\in\mathbb{R}^{\mathtt{gen}(\mathcal{Z}_x(t_0))}_{\geq 0}$ is a generator scaling vector, $G_{\text{fixed}}\in\mathbb{R}^{n_x\times\mathtt{gen}(\mathcal{Z}_x(t_0))}$ is a fixed generator matrix, and $k_{x,0}\in\mathbb{N}_{>0}$ is the initial time step when $\mathcal{S}_{Kx}$ is reached. To choose suitable parameters $J_{\mathcal{Z}_x(t_0)}$, G_{fixed}, and $k_{x,0}$, we give some recommendations in Subsection 3.5.4. Based on $\mathcal{Z}^\star_u(\cdot)$, we define the correction input zonotope sequence

$$\mathcal{Z}^{\star,\circ}_u(t_k)=\begin{cases}\mathcal{Z}^\star_u(t_k) & \text{for } k\in\mathbb{N}_{[0,k_{x,0}-1]}\\ \{\mathbf{0}\} & \text{for } k\geq k_{x,0}\end{cases}, \tag{3.21}$$

which combines both steps of our two-step safe set approach. $\mathcal{Z}^{\star,\circ}_u(t_k)$ is used in the following proposition to prove that $\mathcal{Z}^\star_x(t_0)=\left\langle c^\star_x(t_0),G_{\text{fixed}}\mathtt{diag}(s^\star_{x,0})\right\rangle_Z$ is a safe set.

Proposition 3.12 (Safe Set from (3.20)): Let $\mathbf{0} \in \mathcal{X}$, $\mathbf{0} \in \mathcal{U}$, $\mathcal{S}_{Kx} \subseteq \mathcal{X}$ be a safe set, and let $s^\star_{x,0}$, $c^\star_x(t_0)$, $\mathcal{Z}^\star_u(\cdot)$ be the solution of (3.20) for any $k_{x,0} \in \mathbb{N}_{>0}$. Then, $\mathcal{Z}^\star_x(t_0) = \langle c^\star_x(t_0), G_{\text{fixed}}\texttt{diag}(s^\star_{x,0})\rangle_Z$ is a safe set with corresponding safety-preserving controller in (3.4) and correction input zonotope sequence $\mathcal{Z}^{\star,\circ}_u(\cdot)$. ■

Proof. Control into $\mathcal{S}_{Kx}$: We prove that all $x(t_0) \in \mathcal{Z}^\star_x(t_0)$ can be steered into $\mathcal{S}_{Kx}$ at $t_{k_{x,0}}$ by

$$\begin{aligned}
\Pi_x\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{k_{x,0}}, \mathcal{Z}^\star_x(t_0), \mathcal{Z}^\star_u(\cdot)\right) &\overset{\text{Theorem 3.6}}{=} \Pi_x\widetilde{\mathcal{R}}^{(A,B,\{\mathbf{0}\},\text{CT})}_{Kx}\left(t_{k_{x,0}}, \mathcal{Z}^\star_x(t_0), \mathcal{Z}^\star_u(\cdot)\right) \\
&\qquad \oplus \Pi_x\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{k_{x,0}}, \{\mathbf{0}\}, \{\mathbf{0}\}\right) \\
&\overset{(3.20c)}{=} \Pi_x\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{k_{x,0}}, \{\mathbf{0}\}, \{\mathbf{0}\}\right) \\
&\overset{[38,102]}{\subseteq} \Pi_x\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(\infty, \{\mathbf{0}\}, \{\mathbf{0}\}\right) \\
&\overset{\text{Lemma 3.8}}{\subseteq} \mathcal{S}_{Kx},
\end{aligned}$$

where the second last step follows from $\Pi_x\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{(\cdot)}, \{\mathbf{0}\}, \{\mathbf{0}\}\right)$ being a monotonically increasing sequence that converges to the DT mRPI set [38, 102]. In addition, the constraints in (3.20d) and (3.20e) ensure $x(t) \in \mathcal{X}$ and $u(t) \in \mathcal{U}$ for $t \in [t_0, t_{k_{x,0}})$. Thus, the controller in (3.4) with correction input zonotope sequence $\mathcal{Z}^\star_u(\cdot)$ steers all $x(t_0) \in \mathcal{Z}^\star_x(t_0)$ safely into $\mathcal{S}_{Kx}$ at $t_{k_{x,0}}$.

Control within $\mathcal{S}_{Kx}$: For $k \geq k_{x,0}$, $\mathcal{Z}^{\star,\circ}_u(t_k) = \{\mathbf{0}\}$, which corresponds to the simplified controller in (3.13). Because $\mathcal{S}_{Kx}$ is a safe set, Lemma 3.8 ensures robust constraint satisfaction at all times $t \geq t_{k_{x,0}}$. Thus, $\mathcal{Z}^\star_x(t_0)$ is a safe set. □

In addition to being independent of the safe set $\mathcal{S}_{Kx}$, the COP in (3.20) offers other important properties, as shown in the following theorem.

Theorem 3.13 (Properties of (3.20)): Let $\mathbf{0} \in \mathcal{X}$, $\mathbf{0} \in \mathcal{U}$, and $\mathcal{S}_{Kx} \subseteq \mathcal{X}$ be a safe set. Then, the COP in (3.20) is always feasible, and the cost in (3.20a) is monotonically increasing with increasing $k_{x,0} \in \mathbb{N}_{>0}$. ■

Proof. Feasibility: When choosing $s_{x,0} = \mathbf{0}$, $c_x(t_0) = \mathbf{0}$, $\mathcal{Z}_u(\cdot) = \{\mathbf{0}\}$, we always obtain $\mathcal{Z}_x(t_0) = \{\mathbf{0}\}$ in (3.20b). Because $\Pi_x\widetilde{\mathcal{R}}^{(A,B,\{\mathbf{0}\},\text{CT})}_{Kx}(t_k, \{\mathbf{0}\}, \{\mathbf{0}\}) = \{\mathbf{0}\}$ for any $k \in \mathbb{N}$, the constraint in (3.20c) is satisfied for any $k_{x,0} \in \mathbb{N}_{>0}$. In addition, the satisfaction of (3.20d) and (3.20e) for any $k_{x,0}$ follows from $\mathbf{0} \in \mathcal{S}_{Kx}$ and Lemma 3.8. Thus, the COP in (3.20) is always feasible.

Monotonically increasing cost: Let $s^\star_{x,0}$, $c^\star_x(t_0)$, $\mathcal{Z}^\star_u(\cdot)$ be the solution of (3.20) for any $k_{x,0} \in \mathbb{N}_{>0}$. Subsequently, we show that $s^{\star,+}_{x,0}$, $c^{\star,+}_x(t_0)$, $\mathcal{Z}^{\star,+}_u(\cdot)$ is feasible for $k_{x,0}+1$, where $s^{\star,+}_{x,0} = s^\star_{x,0}$, $c^{\star,+}_x(t_0) = c^\star_x(t_0)$, and $\mathcal{Z}^{\star,+}_u(\cdot)$ is obtained by appending $\{\mathbf{0}\}$ to $\mathcal{Z}^\star_u(\cdot)$. When the previous solution is reused, the cost in (3.20a) of both optimization problems is the same. Thus, when optimizing over all feasible $s_{x,0}$, $c_x(t_0)$, $\mathcal{Z}_u(\cdot)$, the cost in (3.20a) for $k_{x,0}+1$ is always at least as high as for $k_{x,0}$, which implies that the cost is a monotonically increasing function. Subsequently, we prove that $s^\star_{x,0}$, $c^\star_x(t_0)$, $\mathcal{Z}^{\star,+}_u(\cdot)$ is actually feasible for $k_{x,0}+1$. By reusing $s^\star_{x,0}$ and $c^\star_x(t_0)$, the same $\mathcal{Z}^\star_x(t_0) = \langle c^\star_x(t_0), G_{\text{fixed}}\texttt{diag}(s^\star_{x,0})\rangle_Z$ is obtained in (3.20b) for both $k_{x,0}$ and $k_{x,0}+1$. Because $\mathcal{Z}^{\star,+}_u(t_{k_{x,0}}) = \{\mathbf{0}\}$ and $\Pi_x\widetilde{\mathcal{R}}^{(A,B,\{\mathbf{0}\},\text{CT})}_{Kx}(\Delta t, \{\mathbf{0}\}, \{\mathbf{0}\}) = \{\mathbf{0}\}$, the constraint in (3.20c) is also satisfied. We prove fulfillment of (3.20d) and (3.20e) for the last time interval $[t_{k_{x,0}}, t_{k_{x,0}+1})$ by

$$\begin{aligned}\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_{k_{x,0}}, t_{k_{x,0}+1}), \mathcal{Z}^{\star}_{x}(t_0), \mathcal{Z}^{\star,+}_{u}(\cdot)\right) &\overset{\text{Theorem 3.6}}{\subseteq} \widetilde{\mathcal{R}}^{(A,B,\{\mathbf{0}\},\mathrm{CT})}_{Kx}\left([t_{k_{x,0}}, t_{k_{x,0}+1}), \mathcal{Z}^{\star}_{x}(t_0), \mathcal{Z}^{\star,+}_{u}(\cdot)\right)\\ &\qquad \oplus \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_{k_{x,0}}, t_{k_{x,0}+1}), \{\mathbf{0}\}, \{\mathbf{0}\}\right)\\ &\overset{(3.20\mathrm{c}),\mathcal{Z}^{\star,+}_{u}(t_{k_{x,0}})=\{\mathbf{0}\}}{\subseteq} \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_{k_{x,0}}, t_{k_{x,0}+1}), \{\mathbf{0}\}, \{\mathbf{0}\}\right)\\ &\overset{\{\mathbf{0}\}\subseteq\mathcal{S}_{Kx}}{\subseteq} \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left([t_{k_{x,0}}, t_{k_{x,0}+1}), \mathcal{S}_{Kx}, \{\mathbf{0}\}\right)\\ &\overset{\text{Lemma 3.8}}{\subseteq} \mathcal{X}\times\mathcal{U}.\end{aligned}$$

Thus, $s^{\star}_{x,0}$, $c^{\star}_{x}(t_0)$, $\mathcal{Z}^{\star,+}_{u}(\cdot)$ is actually feasible for $k_{x,0}+1$. □

In summary, we can efficiently compute large safe sets along with corresponding safety-preserving state feedback controllers. We have proposed a two-step safe set approach, as illustrated in Fig. 3.2. During $[0, t_{k_{x,0}})$, the general state feedback controller in (3.4) safely steers any initial state $x(t_0) \in \mathcal{Z}_x(t_0)$ into $\mathcal{S}_{Kx}$ and switches to the simplified controller in (3.13) at $t_{k_{x,0}}$. Thus, we are able to satisfy the state and input constraints in (3.1) while providing a large region of operation for our system.

Based on Proposition 3.12 and Theorem 3.6, the zero terminal constraint in (3.20c) ensures that $\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}\left(t_{k_{x,0}}, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \mathcal{S}_{Kx}$. Instead of using $\mathcal{S}_{Kx}$ as a terminal set, we propose COPs in the following subsection that use any safe set as a terminal set.

3.5.3 Safe Set Terminal Constraint

Because the structure of the subsequent COPs is independent of the controller choice, we introduce $\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx|Lw}(\cdot,\cdot,\cdot)$ to represent both reachability analysis operators $\widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx}(\cdot,\cdot,\cdot)$ and $\widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}(\cdot,\cdot,\cdot)$. Closely related to the zero terminal constraint COP in (3.20), we propose a COP that uses any safe set $\mathcal{S}_{\text{safe}} \subseteq \mathcal{X}$ as terminal set, which could be obtained by any of the previously presented approaches for computing safe sets. In particular, let $s^{\star}_{x,0}$, $c^{\star}_{x}(t_0)$, $\mathcal{Z}^{\star}_{u}(\cdot)$ be the solution of the COP

$$\begin{aligned}\underset{s_{x,0},c_x(t_0),\mathcal{Z}_u(\cdot)}{\text{maximize}}\quad & J_{\mathcal{Z}_x(t_0)}(\mathcal{Z}_x(t_0)) && (3.22\mathrm{a})\\ \text{subject to}\quad & \mathcal{Z}_x(t_0) = \langle c_x(t_0), G_{\text{fixed}}\mathtt{diag}(s_{x,0})\rangle_Z && (3.22\mathrm{b})\\ & \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx|Lw}\left(t_{k_{x,0}}, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \mathcal{S}_{\text{safe}} && (3.22\mathrm{c})\\ & \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx|Lw}\left([t_k, t_{k+1}), \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \mathcal{X} \quad \text{for } k \in \mathbb{N}_{[0,k_{x,0}-1]} && (3.22\mathrm{d})\\ & \Pi_u \widetilde{\mathcal{R}}^{\mathbf{M}}_{Kx|Lw}\left([t_k, t_{k+1}), \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \mathcal{U} \quad \text{for } k \in \mathbb{N}_{[0,k_{x,0}-1]}, && (3.22\mathrm{e})\end{aligned}$$

where $J_{\mathcal{Z}_x(t_0)}$ is a concave cost function, $s_{x,0} \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z}_x(t_0))}_{\geq 0}$ is a generator scaling vector, $G_{\text{fixed}} \in \mathbb{R}^{n_x \times \mathtt{gen}(\mathcal{Z}_x(t_0))}$ is a fixed generator matrix, and $k_{x,0} \in \mathbb{N}_{>0}$ is the initial time step when $\mathcal{S}_{\text{safe}}$ is reached. Then, $\mathcal{Z}^{\star}_{x}(t_0) = \left\langle c^{\star}_{x}(t_0), G_{\text{fixed}}\mathtt{diag}(s^{\star}_{x,0})\right\rangle_Z$ is a safe set, as shown in the following proposition that is closely related to Proposition 3.12.

Proposition 3.14 (Properties of (3.22)): Let $s^\star_{x,0}$, $c^\star_x(t_0)$, $\mathcal{Z}^\star_u(\cdot)$ be the solution of (3.22) for $k_{x,0} \in \mathbb{N}_{>0}$ and $\mathcal{S}_\text{safe} \subseteq \mathcal{X}$. Then, $\mathcal{Z}^\star_x(t_0) = \left\langle c^\star_x(t_0), G_\text{fixed}\mathtt{diag}(s^\star_{x,0})\right\rangle_Z$ is a safe set. ■

Proof. The constraints in (3.22d) and (3.22e) enforce robust constraint satisfaction during the time interval $[0, t_{k_{x,0}})$. In addition, the constraint in (3.22c) ensures that the CT state trajectory $x(\cdot)$ starting in $\mathcal{Z}^\star_x(t_0)$ ends in $\mathcal{S}_\text{safe}$ at $t_{k_{x,0}}$. Because $\mathcal{S}_\text{safe}$ is a safe set by assumption, the corresponding safety-preserving controller ensures robust constraint satisfaction at all times $t \geq t_{k_{x,0}}$. Therefore, the state and input constraints in (3.1) are satisfied, which implies $\mathcal{Z}^\star_x(t_0)$ being a safe set. □

The only difference between the two COPs in (3.20) and (3.22) is the terminal constraint in (3.20c) and (3.22c), when choosing the same parameters, the same reachability analysis operator $\widetilde{\mathcal{R}}^\mathbf{M}_{Kx}(\cdot,\cdot,\cdot)$, and $\mathcal{S}_\text{safe} = \mathcal{S}_{Kx}$. If $k_{x,0} \in \mathbb{N}_{>0}$ is small, (3.22) is typically preferred over (3.20) because the zero terminal constraint is somewhat restrictive. Nevertheless, if $k_{x,0}$ is large, many additional optimization variables and constraints must be introduced in (3.22c) compared to (3.20c) for encoding the zonotope containment condition in (2.15), because the number of constraints and optimization variables in (3.20) is independent of $\mathcal{S}_{Kx}$.

The COP in (3.22) assumes that a safe set $\mathcal{S}_\text{safe} \subseteq \mathcal{X}$ is given. By slightly modifying (3.22), we present a one-step approach to compute a large safe set without requiring this assumption, similar to [52]. In particular, let $s^\star_{x,0}$, $c^\star_x(t_0)$, $\mathcal{Z}^\star_u(\cdot)$ be the solution of the COP

$$\begin{aligned}
\underset{s_{x,0},c_x(t_0),\mathcal{Z}_u(\cdot)}{\text{maximize}} \quad & J_{\mathcal{Z}_x(t_0)}(\mathcal{Z}_x(t_0)) && \text{(3.23a)}\\
\text{subject to} \quad & \mathcal{Z}_x(t_0) = \langle c_x(t_0), G_\text{fixed}\mathtt{diag}(s_{x,0})\rangle_Z && \text{(3.23b)}\\
& \Pi_x \widetilde{\mathcal{R}}^\mathbf{M}_{Kx|Lw}\left(t_{k_{x,0}}, \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \mathcal{Z}_x(t_0) && \text{(3.23c)}\\
& \Pi_x \widetilde{\mathcal{R}}^\mathbf{M}_{Kx|Lw}\left([t_k, t_{k+1}), \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \mathcal{X} \quad \text{for } k \in \mathbb{N}_{[0,k_{x,0}-1]} && \text{(3.23d)}\\
& \Pi_u \widetilde{\mathcal{R}}^\mathbf{M}_{Kx|Lw}\left([t_k, t_{k+1}), \mathcal{Z}_x(t_0), \mathcal{Z}_u(\cdot)\right) \subseteq \mathcal{U} \quad \text{for } k \in \mathbb{N}_{[0,k_{x,0}-1]}, && \text{(3.23e)}
\end{aligned}$$

where $s_{x,0} \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z}_x(t_0))}_{>0}$ is a positive generator scaling vector. There are only two differences between (3.22) and (3.23): First, to use the zonotope containment constraint in (2.15), the elements of the generator scaling vector $s_{x,0}$ must be greater than zero in (3.23). Second, instead of using a given safe set $\mathcal{S}_\text{safe}$ as terminal set in (3.22c), the terminal set in (3.23c) is the initial state set $\mathcal{Z}_x(t_0)$ itself. Thus, this slightly modified approach is similar to the COP in (3.16), which aims at finding a small instead of a large safe set. In the following proposition, we show important properties of the COP in (3.23).

Proposition 3.15 (Properties of (3.23)): Let $s^\star_{x,0}$, $c^\star_x(t_0)$, $\mathcal{Z}^\star_u(\cdot)$ be the solution of (3.23) for $k_{x,0} \in \mathbb{N}_{>0}$. Then, $\mathcal{Z}^\star_x(t_0) = \left\langle c^\star_x(t_0), G_\text{fixed}\mathtt{diag}(s^\star_{x,0})\right\rangle_Z$ is a safe set. If, in addition to the feasibility of (3.23), the initial time step $k_{x,0} = 1$, $\mathcal{Z}^\star_x(t_0)$ is an RCI set. ■

Proof. The proof is similar to Lemma 3.11 and is presented subsequently for the sake of completeness.

Safe set: The constraint in (3.23c) ensures that the CT state trajectory $x(\cdot)$ starting in $\mathcal{Z}^\star_x(t_0)$ ends in $\mathcal{Z}^\star_x(t_0)$ at $t_{k_{x,0}}$. Because $x(\cdot)$ might leave $\mathcal{Z}^\star_x(t_0)$ during $[0, t_{k_{x,0}})$, the constraints in (3.23d) and (3.23e) enforce robust constraint satisfaction during the time interval $[0, t_{k_{x,0}})$.

By induction, it follows that robust constraint satisfaction can be achieved for an infinite time horizon if $x(t_0) \in \mathcal{Z}_x^\star(t_0)$, which implies that $\mathcal{Z}_x^\star(t_0)$ is a safe set.

RCI set: If, in addition to the feasibility of (3.23), the initial time step $k_{x,0} = 1$, (3.23c) enforces the projected reachable set of $\mathcal{Z}_x^\star(t_0)$ to lie within $\mathcal{Z}_x^\star(t_0)$ at Δt. Thus, $\mathcal{Z}_x^\star(t_0)$ is an RCI set in this case. □

To construct a large RCI set, we can alternatively slightly modify the COP in (3.15). In particular, the cost in (3.15a) can be changed to aim for a volume maximization of the safe set, and the constraints in (3.15b) and (3.15c) can be removed.

In summary, we can efficiently determine zonotopic safe sets for $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$ that are as large as possible to maximize the region of operation for our system. In the following subsection, we present recommendations to choose suitable parameters of the presented COPs.

3.5.4 Choice of Parameters

To obtain large safe sets by solving the COPs proposed in Subsections 3.5.2 and 3.5.3, it is crucial to select suitable parameters. Thus, we recommend suitable $J_{\mathcal{Z}_x(t_0)}$, G_{fixed}, and $k_{x,0}$ in this subsection.

Ideally, we want to maximize the volume of the safe set $\mathcal{Z}_x(t_0) \subseteq \mathcal{X}$. However, computing the volume of a general zonotope is combinatorially complex with respect to the number of columns of the generator matrix [162]. A commonly used heuristic for maximizing the volume of polytopic RCI sets is the volume maximization of a contained ellipsoid [132, 134]. Inspired by this approach, we propose to maximize the volume of an ellipsoid $\mathcal{E} = \langle c, S \rangle_E \subset \mathbb{R}^{n_x}$ that suitably approximates $\mathcal{Z}_x(t_0)$, which results in an SDP problem [54, 163, 164]. The volume of $\mathcal{E}$ is proportional to $\det(S)$, and the determinant is logarithmically concave on the set of symmetric positive definite matrices [55]. Thus, maximizing the volume of $\mathcal{E}$ can be achieved by choosing $\log(\det(S))$ for the concave cost function $J_{\mathcal{Z}_x(t_0)}$ [55, Sec. 8.4.2]. To ensure that $\mathcal{E}$ is a suitable approximation of $\mathcal{Z}_x(t_0)$, we also add the cost-dependent constraint

$$c + S^{1/2} e_i \in \mathcal{Z}_x(t_0) \quad \text{for } i \in \mathbb{N}_{[1,n_x]}$$

to the constraints of the original COPs, where e_i denotes the i^{th} unit vector of the Euclidean space in $\mathbb{R}^{n_x}$. Thus, instead of enforcing $\mathcal{E} \subseteq \mathcal{Z}_x(t_0)$, we opt for this simple approximation to keep the amount of additional constraints small [165]. Instead of using an ellipsoid, we can also aim at maximizing the volume of a multidimensional interval $\mathcal{I} = \langle c_{\mathcal{I}}, \mathtt{diag}(s_{\mathcal{I}}) \rangle_Z \subset \mathbb{R}^{n_x}$ contained in $\mathcal{Z}_x(t_0)$. In this case, we choose the geometric mean of $s_{\mathcal{I}} \in \mathbb{R}^{n_x}_{>0}$ for $J_{\mathcal{Z}_x(t_0)}$ because it is a monotonic function of the volume of $\mathcal{I}$. To ensure that $\mathcal{I} \subseteq \mathcal{Z}_x(t_0)$, we use the zonotope containment condition in (2.15) and add this cost-dependent constraint to the constraints of the original COP. Alternatively, using the sum or the geometric mean of the generator scaling vector $s_{x,0}$ for $J_{\mathcal{Z}_x(t_0)}$ are reasonable heuristics to obtain large safe sets. For instance, the geometric mean of $s_{x,0}$ is a monotonic function of the volume of $\mathcal{Z}_x(t_0)$ if the fixed generator matrix $G_{\text{fixed}} \in \mathbb{R}^{n_x \times \mathtt{gen}(\mathcal{Z}_x(t_0))}$ is an identity matrix.

To cover the state constraint set $\mathcal{X}$, we can uniformly sample from the unit hypersphere and use the obtained points as columns of G_{fixed}. Because uniform sampling in high-dimensional spaces is a complex task, it is worthwhile to examine the sparsity of the system and input matrices [35, 143]. Alternatively, a good choice for G_{fixed} can be the generator matrix of any small safe set because it already incorporates some effects of the disturbance set $\mathcal{W} \subset \mathbb{R}^{n_x}$.

The parameter $k_{x,0}$ corresponds to the time step when all states starting in $\mathcal{Z}_x(t_0)$ reach the other safe set. Thus, this parameter is used to balance accuracy and computational complexity. Based on Theorem 3.13, we can also increase $k_{x,0}$ until the cost in (3.20a) has converged if the COP in (3.20) is solved. In this case, usually $\mathcal{S}_{Kx} \nsubseteq \mathcal{Z}_x^\star(t_0)$ or even $\mathcal{Z}_x^\star(t_0) \subseteq \mathcal{S}_{Kx}$ when choosing $k_{x,0} = 1$.

3.6 Numerical Examples

In this section, we demonstrate the effectiveness of our proposed safe set approaches using four numerical examples taken from the literature: We consider a CT double-integrator system in Subsection 3.6.1, a DT double-integrator system in Subsection 3.6.2, a vehicle platooning system in Subsection 3.6.3, and a chain of mass-spring-damper (MSD) systems in Subsection 3.6.4. When simulating these systems to generate random trajectories, we use the convex cost $J_\lambda = \|\lambda\|_\infty$ in (2.1) to obtain the not necessarily unique parameter vectors that are required for the control laws in (3.4) and (3.10), respectively.

3.6.1 Continuous-Time Double-Integrator System

To compare the performance of our presented safe set approaches, we consider the simple double-integrator system

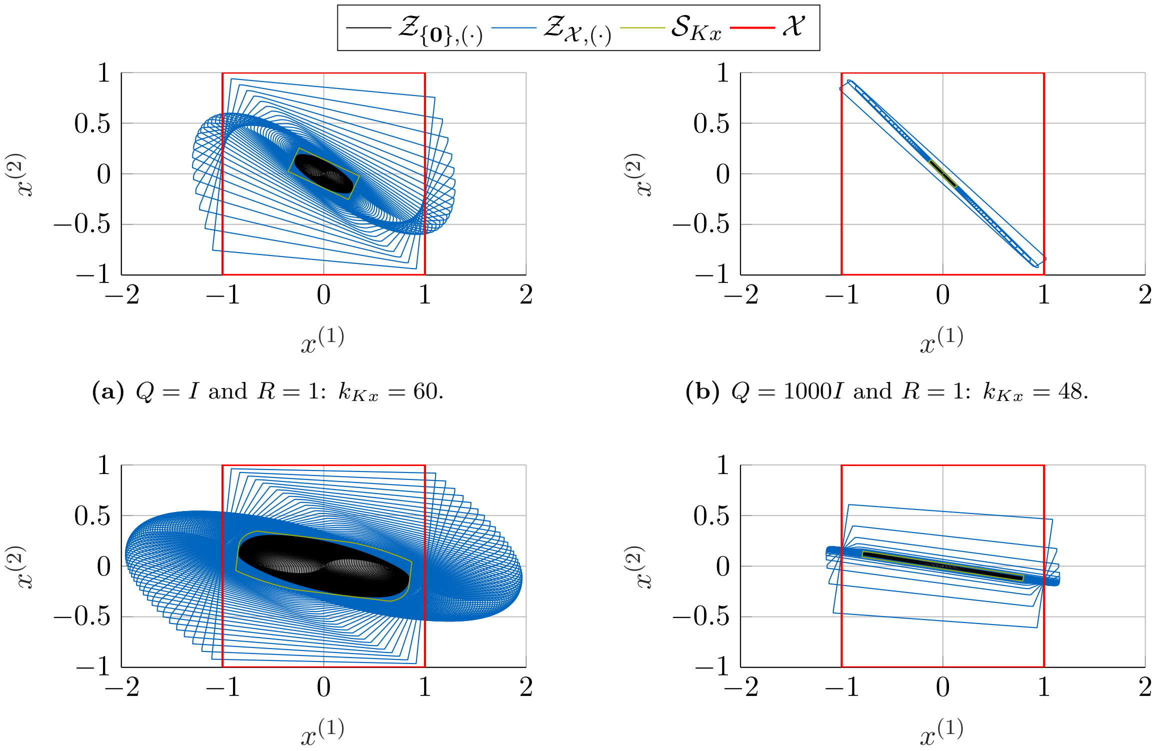
$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) + w(t)$$

with disturbance set $\mathcal{W} = [-0.1, 0.1]^2$ and state and input constraint sets $\mathcal{X} = [-1, 1]^2$ and $\mathcal{U} = [-1, 1]$, respectively. In addition, the sampling period is $\Delta t = 0.1\,\text{s}$. Subsequently, we show the small and large safe sets obtained by following our proposed robust control approaches.

Small Safe Sets

First, we compute small safe sets by executing Alg. 3.1. We choose the convergence tolerance to be $\epsilon = 0.01$. The stabilizing state feedback matrix $K \in \mathbb{R}^{1\times 2}$ is obtained using LQR-based controller synthesis [155]. In Fig. 3.3, we visualize the computed small safe sets when choosing different state and input weighting matrices for the LQR-based controller. Thus, Alg. 3.1 obtains small safe sets for a wide range of state feedback matrices.

Second, we solve the COP in (3.15) to construct an RCI set. For the scaling point $\mathring{x} = \texttt{center}\,(\mathcal{X})$, ensuring the feasibility of (3.15) requires the number of generators $\texttt{gen}\,(\mathcal{Z}_x(t_0))$ to be at least 8. The corresponding RCI set is shown in Fig. 3.4a. In addition, we visualize the safe sets that are obtained by executing Alg. 3.2 with $o_{x,0} \in \mathbb{N}_{>0}$ and $k_{x,0} = 1$, i.e., these safe sets are also RCI sets based on Lemma 3.11. In particular, Alg. 3.2 determines a small RCI set for $o_{x,0} = 1$ and the same small RCI set with only 4 generators for all $o_{x,0} > 1$. Thus, fewer generators are needed for computing an RCI set compared to (3.15), which shows the conservativeness of the robust control invariance approach in Subsection 3.4.2. In Fig. 3.4b,



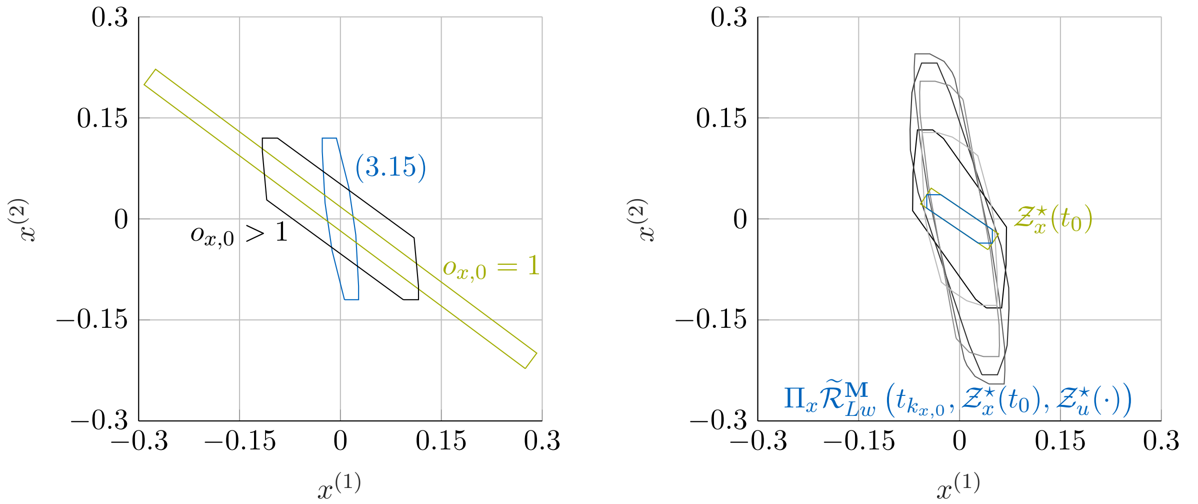
(a) $Q = I$ and $R = 1$: $k_{Kx} = 60$.

(b) $Q = 1000I$ and $R = 1$: $k_{Kx} = 48$.

(c) $Q = I$ and $R = 20$: $k_{Kx} = 165$.

(d) $Q = \texttt{diag}\left(\begin{bmatrix}1 & 40\end{bmatrix}^T\right)$ and $R = 1$: $k_{Kx} = 303$.

Figure 3.3: Small safe sets of CT double-integrator system with corresponding time step k_{Kx} obtained by executing Alg. 3.1. The stabilizing state feedback matrix $K \in \mathbb{R}^{1\times 2}$ is obtained using LQR-based controller synthesis with different state and input weighting matrices $Q \in \mathbb{R}^{2\times 2}$ and $R \in \mathbb{R}$, respectively. In addition, both zonotope sequences $\mathcal{Z}_{\{\mathbf{0}\},(\cdot)}$ and $\mathcal{Z}_{\mathcal{X},(\cdot)}$ of Alg. 3.1 are shown.



(a) RCI sets obtained by solving the COP in (3.15) with $\texttt{gen}\,(\mathcal{Z}_x(t_0)) = 8$ and by executing Alg. 3.2 with $k_{x,0} = 1$ and different $o_{x,0} \in \mathbb{N}_{>0}$.

(b) Small safe set obtained by executing Alg. 3.2 with $k_{x,0} = 5$ and $o_{x,0} = 1$. In addition, reachable sets $\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{Lw}\left([t_k, t_{k+1}), \mathcal{Z}_x^\star(t_0), \mathcal{Z}_u^\star(\cdot)\right)$ with $k \in \mathbb{N}_{[0,k_{x,0}-1]}$ are shown, where a lighter gray tone corresponds to a larger prediction horizon.

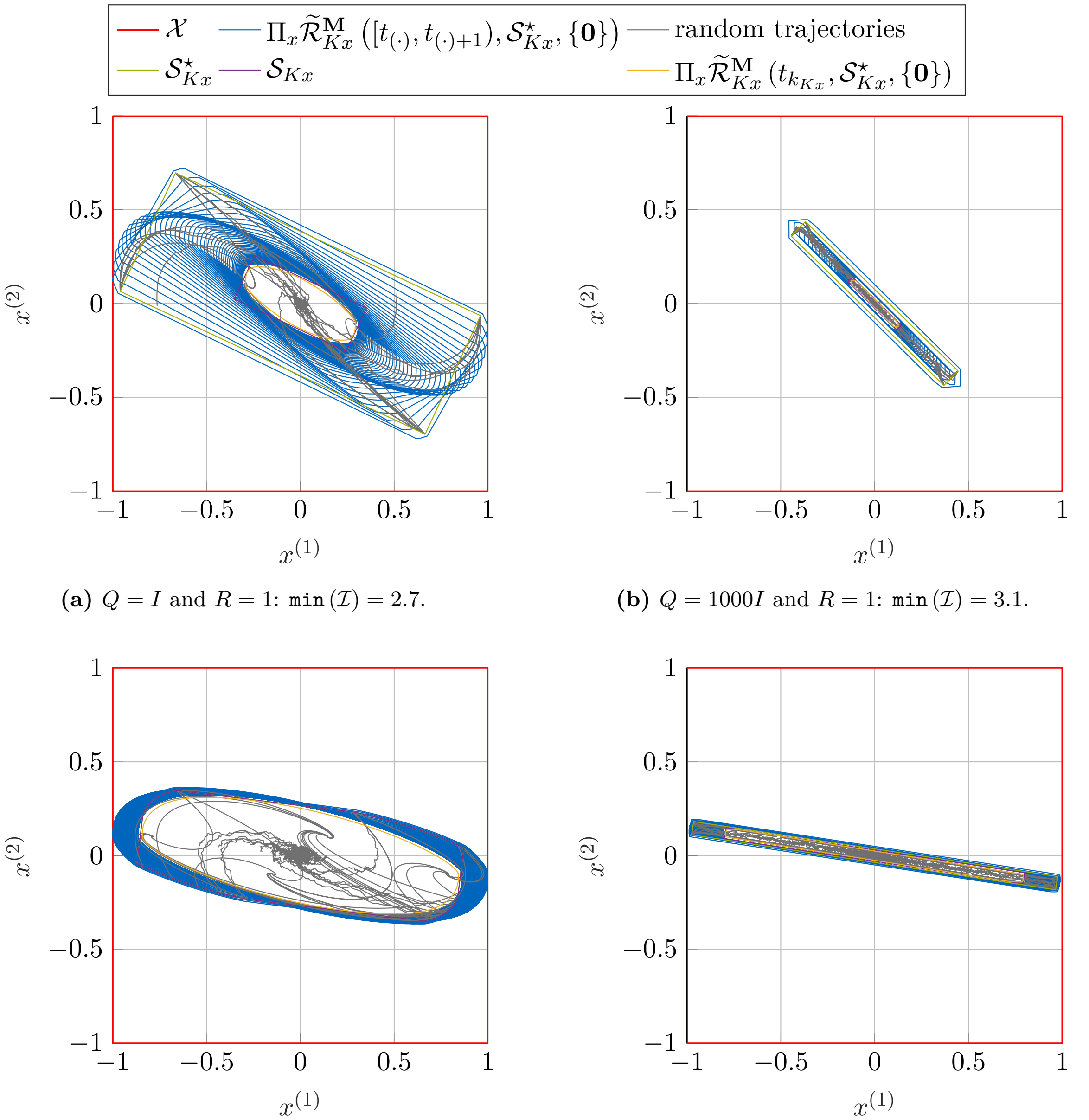
Figure 3.4: Small safe sets of CT double-integrator system with scaling point $\mathring{x} = \texttt{center}\,(\mathcal{X})$.

we show the small safe parallelotope that is obtained by executing Alg. 3.2 with $o_{x,0} = 1$ and $k_{x,0} = 5$. Because robust control invariance is not enforced in this case, more flexibility is available for optimizing a smaller safe set compared to the ones in Fig. 3.4a.

Large Safe Sets

First, we compute large safe sets using the safe set scaling method in Alg. 3.3. We choose the corresponding maximum interval radius to be $\epsilon = 0.01$. In Fig. 3.5, we visualize the computed scaled safe sets when choosing the same state and input weighting matrices of the LQR-based controller as in Fig. 3.3. As can be seen, the optimal scaling factors significantly depend on the chosen weighting matrices $Q \in \mathbb{R}^{2\times 2}$ and $R \in \mathbb{R}$. Because the reachable sets of $\mathcal{S}_{Kx}$ touch the bounds of $\mathcal{X}$ for $Q = I$ and $R = 20$, the corresponding optimal scaling factor obtained by executing Alg. 3.3 is only 1.0, i.e., the small safe set is not enlarged.

Second, we solve the COP in (3.20) to construct large safe sets. To cover $\mathcal{X} \subset \mathbb{R}^2$, we choose the columns of the fixed generator matrix $G_{\text{fixed}} \in \mathbb{R}^{2\times 10}$ to be 10 uniformly distributed points around the top half unit circle. In addition, we use the sum of the generator scaling vector $s_{x,0}$ for the linear cost function $J_{\mathcal{Z}_x(t_0)} = \mathbf{1}^T s_{x,0}$ in (3.20a), rendering the COP in (3.20) a simple linear programming problem. Based on Theorem 3.13, the cost in (3.20a) is monotonically increasing with increasing initial time step $k_{x,0} \in \mathbb{N}_{>0}$. Thus, we increment $k_{x,0}$ starting from 1 until the difference between two consecutive optimal costs is smaller than 10^{-5}. In Fig. 3.6, we visualize the evolution of the corresponding large safe sets when choosing the same state and input weighting matrices of the LQR-based controller as in Fig. 3.3. As can be observed, the choice of the state and input weighting matrices has only a small influence on the rate of



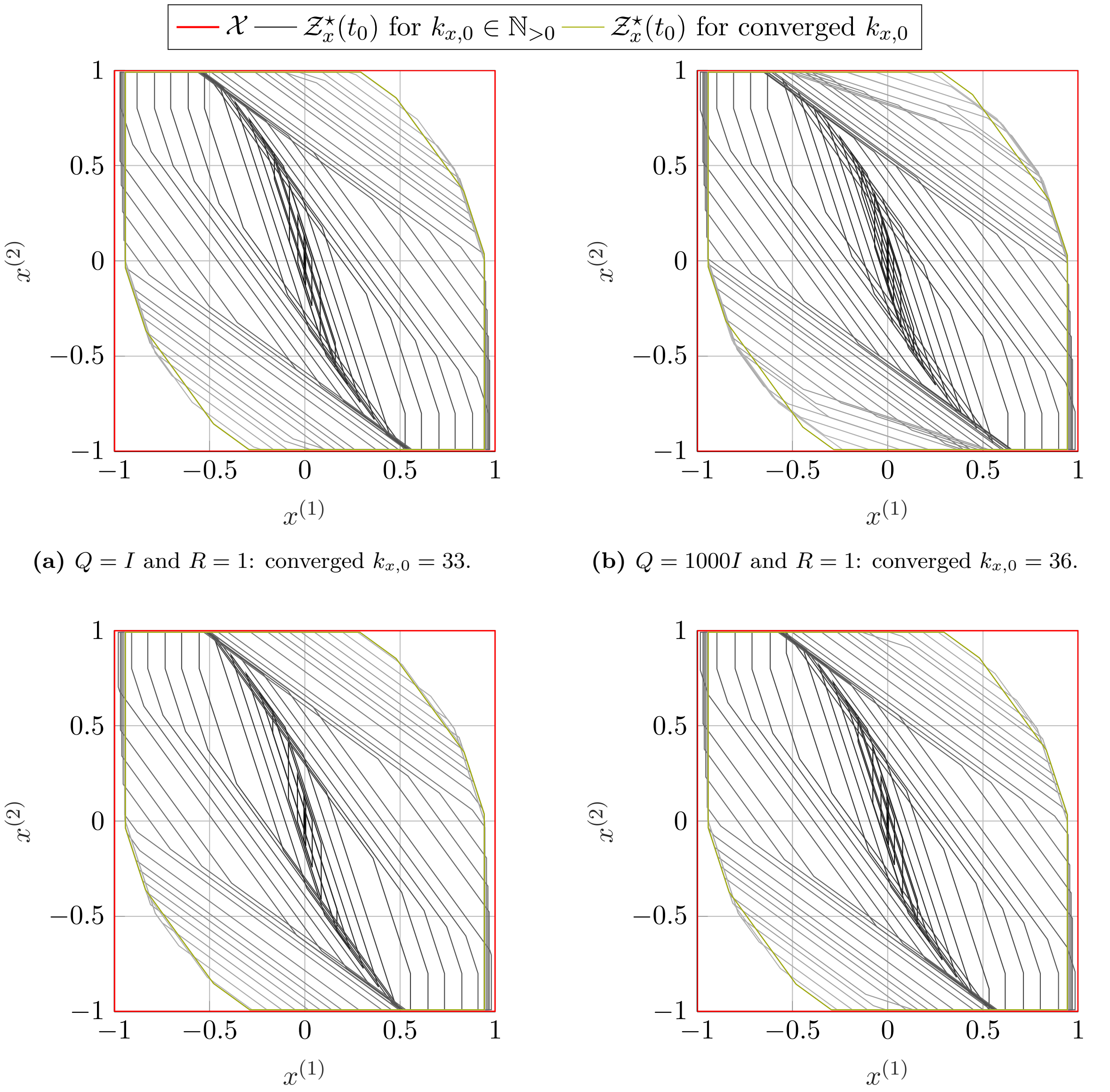
(a) $Q = I$ and $R = 1$: $\texttt{min}\,(\mathcal{I}) = 2.7$.

(b) $Q = 1000I$ and $R = 1$: $\texttt{min}\,(\mathcal{I}) = 3.1$.

(c) $Q = I$ and $R = 20$: $\texttt{min}\,(\mathcal{I}) = 1.0$.

(d) $Q = \texttt{diag}\left(\begin{bmatrix}1 & 40\end{bmatrix}^T\right)$ and $R = 1$: $\texttt{min}\,(\mathcal{I}) = 1.2$.

Figure 3.5: Scaled safe sets of CT double-integrator system with optimal scaling factor $\texttt{min}\,(\mathcal{I})$ obtained by executing Alg. 3.3. The state and input weighting matrices $Q \in \mathbb{R}^{2\times 2}$ and $R \in \mathbb{R}$ of the LQR-based controller are chosen analogously to Fig. 3.3. In addition, reachable sets and random trajectories are shown.



(a) $Q = I$ and $R = 1$: converged $k_{x,0} = 33$.

(b) $Q = 1000I$ and $R = 1$: converged $k_{x,0} = 36$.

(c) $Q = I$ and $R = 20$: converged $k_{x,0} = 32$.

(d) $Q = \texttt{diag}\left(\begin{bmatrix}1 & 40\end{bmatrix}^T\right)$ and $R = 1$: converged $k_{x,0} = 33$.

Figure 3.6: Large safe sets of CT double-integrator system obtained by solving (3.20) with increasing initial time step $k_{x,0}$. The state and input weighting matrices $Q \in \mathbb{R}^{2\times 2}$ and $R \in \mathbb{R}$ of the LQR-based controller are chosen analogously to Fig. 3.3. The large safe sets with a lighter gray tone correspond to a larger $k_{x,0}$.

convergence and the shape of the optimal large safe sets. This observation indicates that a wide range of stabilizing state feedback matrices can be chosen without affecting the large safe sets. In Fig. 3.7, we show the converged large safe sets along with their reachable sets and random trajectories. As can be seen, $\Pi_x \widetilde{\mathcal{R}}_{Kx}^{\mathbf{M}}\left(t_{k_{x,0}}, \mathcal{Z}_x^\star(t_0), \mathcal{Z}_u^\star(\cdot)\right) \subseteq \mathcal{S}_{Kx}$ in all cases although (3.20) only implicitly considers the small safe set $\mathcal{S}_{Kx}$.

Third, we demonstrate that our safe set approach is not overly conservative. To achieve this, we compute a tight RCI under-approximation of the DT MRCI set, which is obtained by executing Alg. 2.2 with convergence tolerance $\epsilon = 10^{-10}$. In addition, we would like to solve a DT version of the COP in (3.20). Thus, to ensure a fair comparison, we enforce satisfaction of the state and input constraints in (3.20d) and (3.20e) only at sampling times but not between them. Moreover, we choose the fixed generator matrix $G_{\text{fixed}} \in \mathbb{R}^{2\times 10}$, linear cost function $J_{\mathcal{Z}_x(t_0)} = \mathbf{1}^T s_{x,0}$, and stabilizing state feedback matrix $K \in \mathbb{R}^{1\times 2}$ analogously to the previous paragraph. Similar to Fig. 3.6, we visualize in Fig. 3.8 the evolution of the large safe sets when incrementing the initial time step $k_{x,0}$ starting from 1 until the difference between two consecutive optimal costs is smaller than 10^{-5}. As can be observed, our safe sets are tight under-approximations of the DT MRCI set, which shows that our approach is not conservative. Similar to Fig. 3.6, the choice of the state and input weighting matrices has only a small influence on the converged large safe set.

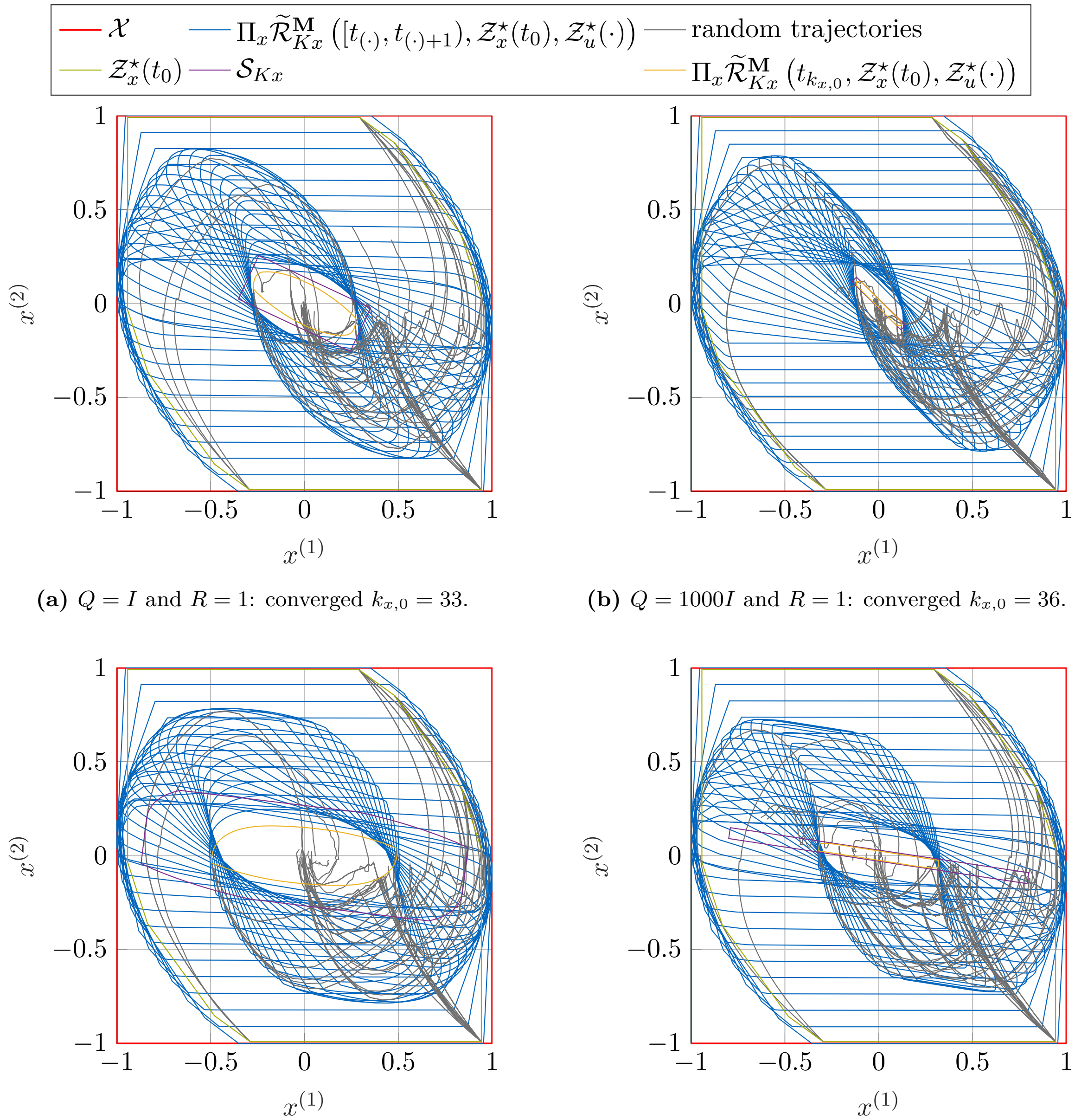
Fourth, we solve the COP in (3.22) to construct large safe sets when choosing the same $G_{\text{fixed}} \in \mathbb{R}^{2\times 10}$ as before. In addition, the stabilizing state feedback matrix $K \in \mathbb{R}^{1\times 2}$ is obtained using LQR-based controller synthesis with state and input weighting matrices $Q = I$ and $R = 1$, respectively. For the terminal constraint in (3.22c), we use the safe set $\mathcal{S}_{\text{safe}} \subseteq \mathcal{X}$ that is obtained by executing the safe set scaling method in Alg. 3.3, which is shown in Fig. 3.5a. In contrast to (3.20), the cost in (3.22) is not necessarily monotonically increasing with increasing $k_{x,0} \in \mathbb{N}_{>0}$. Nevertheless, we increment $k_{x,0}$ starting from 1 until the absolute difference between two consecutive optimal costs is smaller than 10^{-5}, which typically implies that the terminal constraint in (3.22c) becomes inactive. In Fig. 3.9, we visualize the evolution of the corresponding large safe sets when choosing the presented cost functions in Subsection 3.5.4. As can be observed, the volume of the safe set can be significantly enlarged irrespective of the chosen cost function, which mainly influences the shape of the safe sets.

Fifth, we solve the COP in (3.23) to construct large RCI sets when choosing the same $G_{\text{fixed}} \in \mathbb{R}^{2\times 10}$ as before. Based on Proposition 3.15, the obtained large safe set is an RCI set if (3.23) is feasible for the initial time step $k_{x,0} = 1$. In Fig. 3.10, we visualize such large RCI sets when choosing the presented cost functions in Subsection 3.5.4. Compared to the large safe sets in Fig. 3.9, all computed RCI sets result in smaller costs, i.e., in worse performance. Nevertheless, these safe sets are RCI, a beneficial property that can be exploited.

3.6.2 Discrete-Time Double-Integrator System

To compare our safe set approaches with two other established techniques [132,134], we consider a DT double-integrator system [166], which is used in [132, Sec. VI-A] and in [134, Sec. V-B]. The corresponding DT dynamics is

$$x(t_{k+1}) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t_k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t_k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w_D(t_k)$$



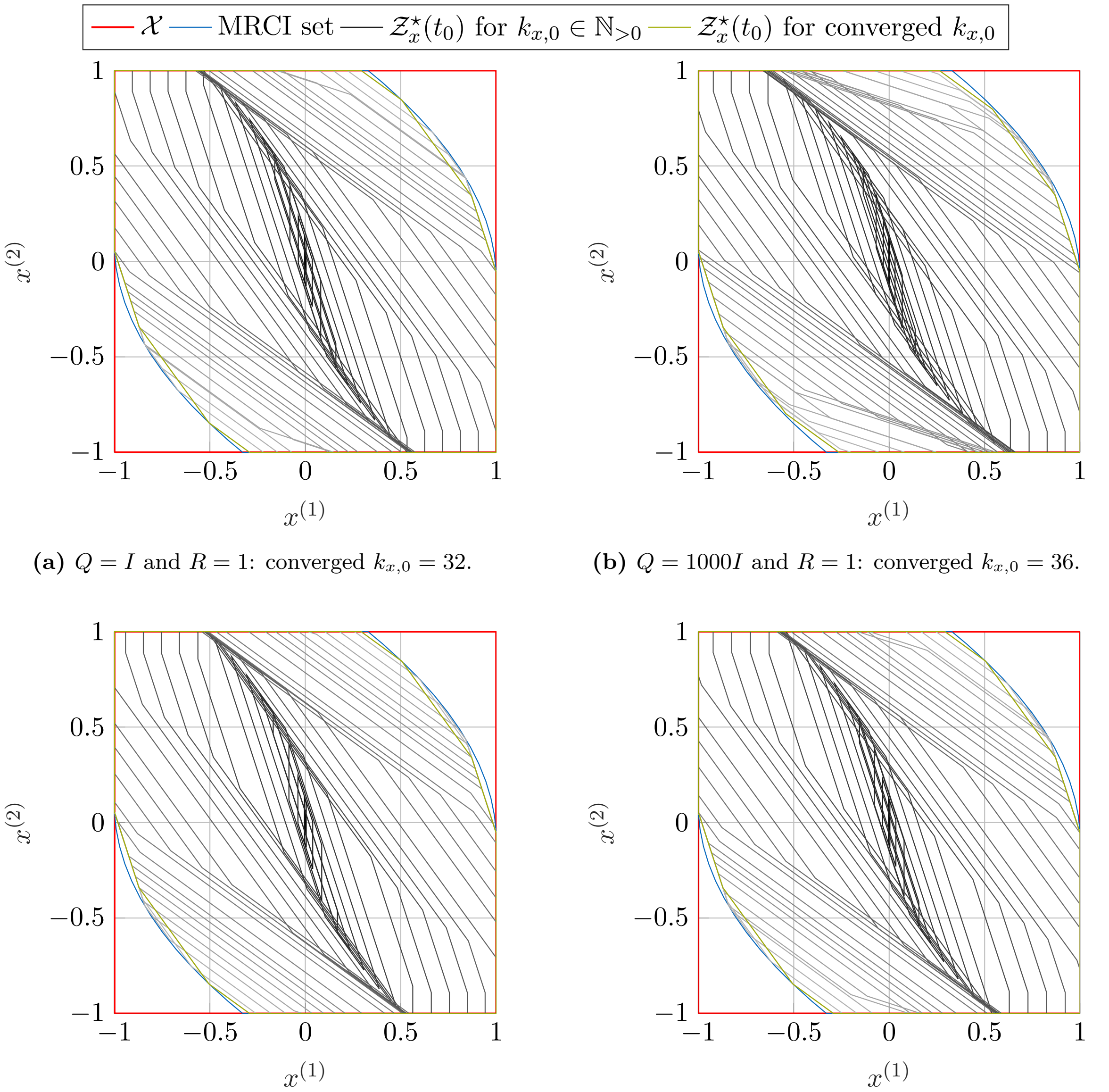
(a) $Q = I$ and $R = 1$: converged $k_{x,0} = 33$.

(b) $Q = 1000I$ and $R = 1$: converged $k_{x,0} = 36$.

(c) $Q = I$ and $R = 20$: converged $k_{x,0} = 32$.

(d) $Q = \texttt{diag}\left(\begin{bmatrix}1 & 40\end{bmatrix}^T\right)$ and $R = 1$: converged $k_{x,0} = 33$.

Figure 3.7: Large safe sets of CT double-integrator system obtained by solving (3.20) with converged initial time step $k_{x,0}$, which correspond to the green large safe sets in Fig. 3.6. The state and input weighting matrices $Q \in \mathbb{R}^{2\times 2}$ and $R \in \mathbb{R}$ of the LQR-based controller are chosen analogously to Fig. 3.6. In addition, reachable sets and random trajectories are shown.



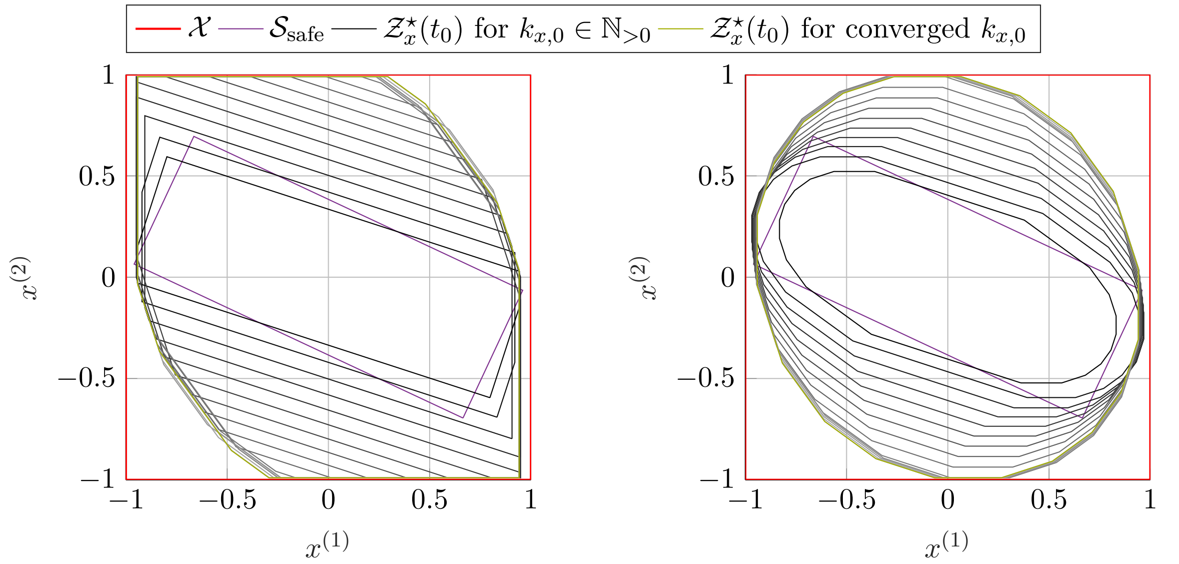
(a) $Q = I$ and $R = 1$: converged $k_{x,0} = 32$.

(b) $Q = 1000I$ and $R = 1$: converged $k_{x,0} = 36$.

(c) $Q = I$ and $R = 20$: converged $k_{x,0} = 31$.

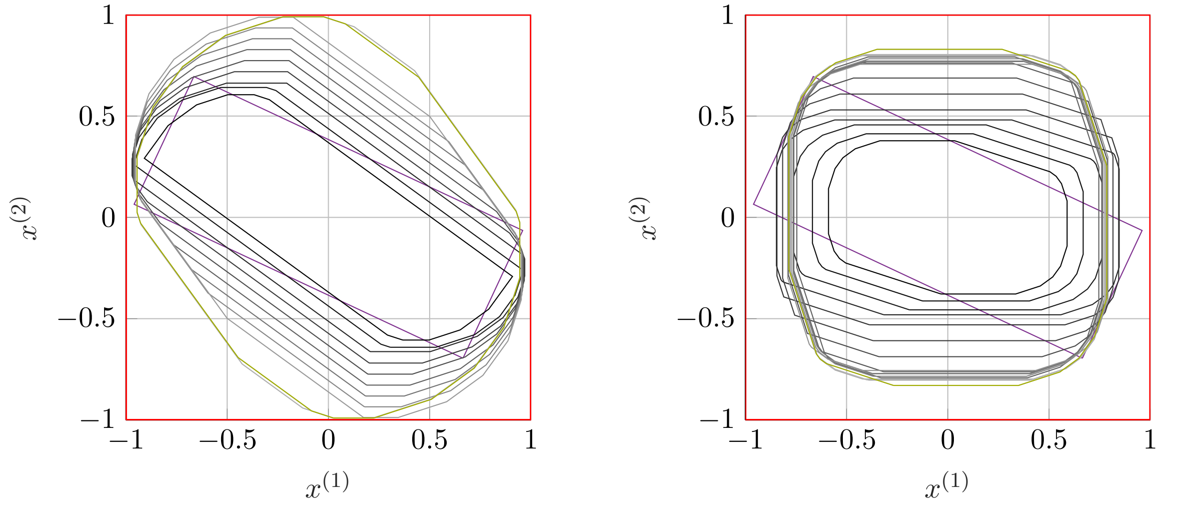
(d) $Q = \texttt{diag}\left(\begin{bmatrix}1 & 40\end{bmatrix}^T\right)$ and $R = 1$: converged $k_{x,0} = 32$.

Figure 3.8: Comparison of DT, large safe sets of CT double-integrator system and RCI under-approximation of MRCI set, which is obtained by executing Alg. 2.2 with $\epsilon = 10^{-10}$. The large safe sets are obtained by solving a DT version of (3.20). Moreover, the large safe sets with a lighter gray tone correspond to a larger initial time step $k_{x,0}$. In addition, the state and input weighting matrices $Q \in \mathbb{R}^{2\times 2}$ and $R \in \mathbb{R}$ of the LQR-based controller are chosen analogously to Fig. 3.6.



(a) $\mathbf{1}^T s_{x,0}$: converged $k_{x,0} = 18$.

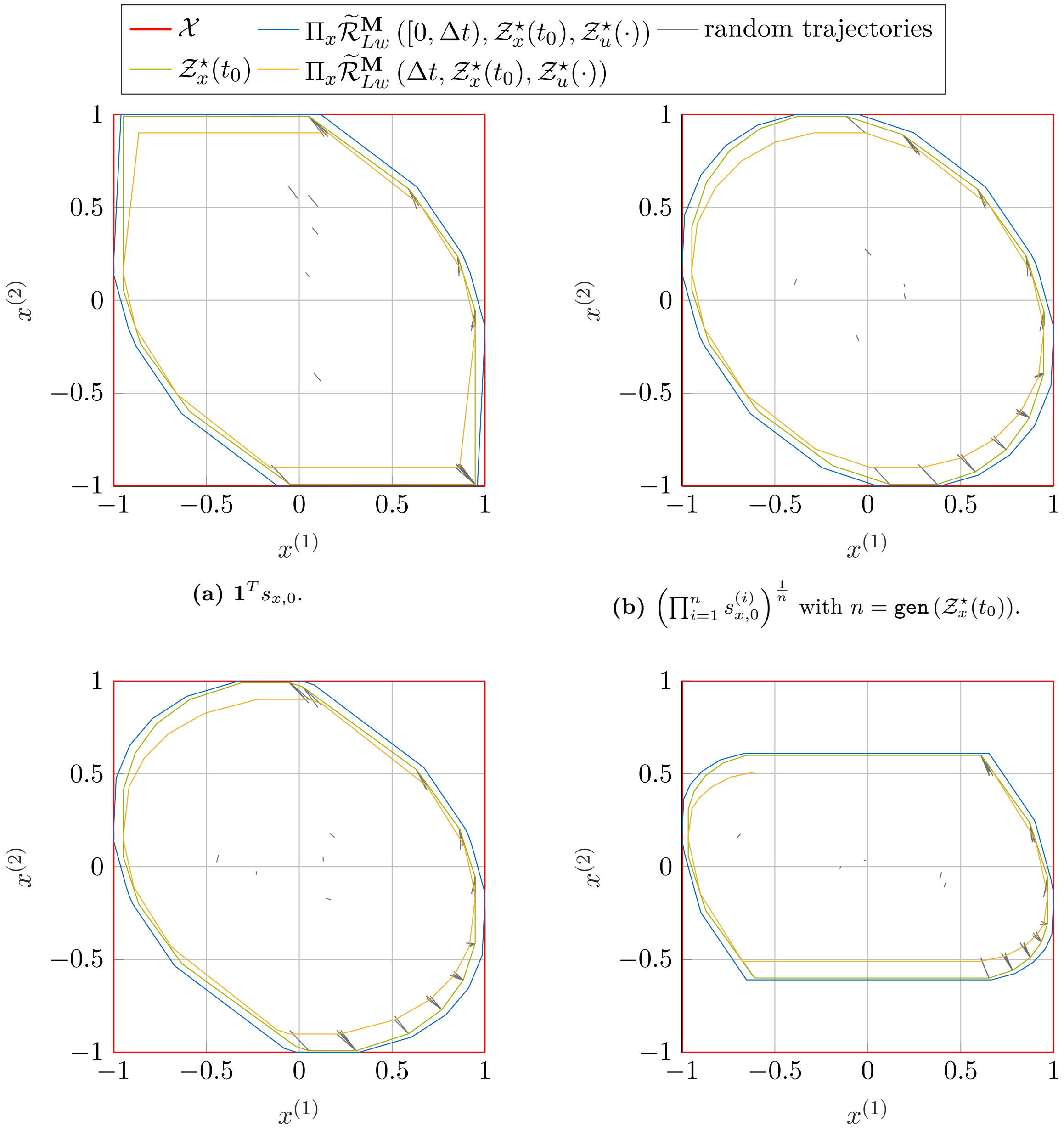
(b) $\left(\prod_{i=1}^{n} s_{x,0}^{(i)}\right)^{\frac{1}{n}}$ with $n = \texttt{gen}\left(\mathcal{Z}_x^\star(t_0)\right)$: converged $k_{x,0} = 15$.



(c) Volume of approximative ellipsoid (see Subsection 3.5.4): converged $k_{x,0} = 11$.

(d) Volume of enclosed multidimensional interval (see Subsection 3.5.4): converged $k_{x,0} = 17$.

Figure 3.9: Large safe sets of CT double-integrator system obtained by solving (3.22) with increasing initial time step $k_{x,0}$ using different cost functions in (3.22a). The large safe sets with a lighter gray tone correspond to a larger $k_{x,0}$.



(a) $\mathbf{1}^T s_{x,0}$.

(b) $\left(\prod_{i=1}^{n} s_{x,0}^{(i)}\right)^{\frac{1}{n}}$ with $n = \texttt{gen}\left(\mathcal{Z}_x^\star(t_0)\right)$.

(c) Volume of approximative ellipsoid (see Subsection 3.5.4).

(d) Volume of enclosed multidimensional interval (see Subsection 3.5.4).

Figure 3.10: RCI sets of CT double-integrator system obtained by solving (3.23) using different cost functions in (3.23a). In addition, reachable sets and random trajectories are shown.

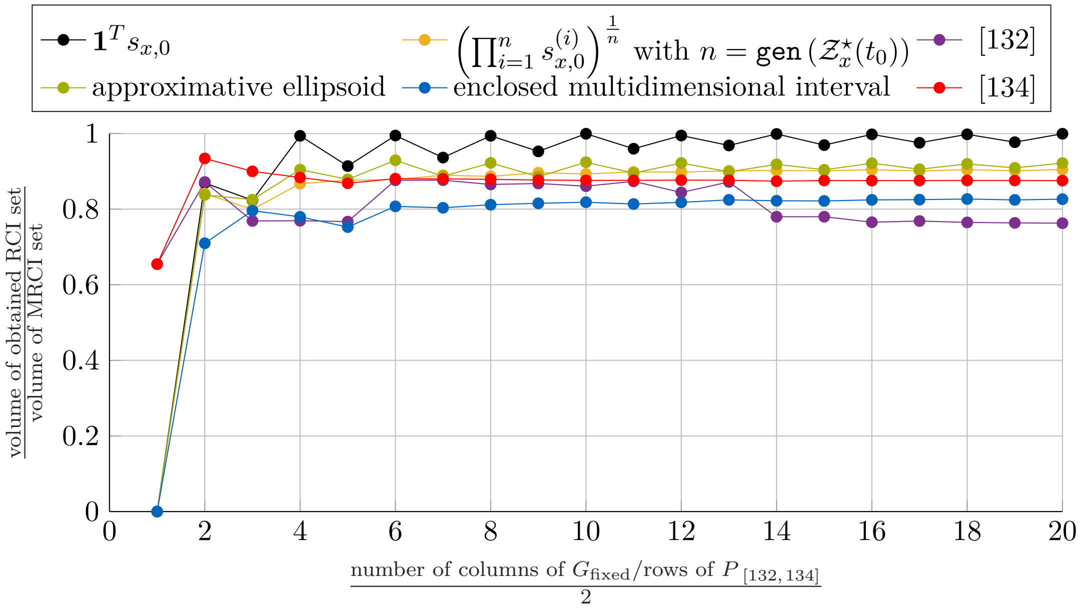


Figure 3.11: Evolution of the RCI set volumes of DT double-integrator system. The sets are computed by solving (3.23) using different cost functions in (3.23a) and by following the established approaches in [132, 134].

with DT disturbance set $\mathcal{W}_D = [-0.1, 0.1]$ and state and input constraint sets $\mathcal{X} = [-1, 1]^2$ and $\mathcal{U} = [-1, 1]$, respectively.

Both established methods require an initial guess for a matrix $P_{[132,134]}$ that is similar to the transpose of the fixed generator matrix G_{fixed} in our presented safe set approaches. In [132], it is proposed to choose $P_{[132,134]}$ such that a regular polytope is described. Similarly, the first author of [134] suggests in his doctoral dissertation to sample uniformly distributed points around the top half unit circle in the two-dimensional case and to use the sampled points as rows of $P_{[132,134]}$ [167, Rmk. 4]. To provide a fair comparison of the three considered approaches, we follow this suggestion for constructing $P_{[132,134]}$ and G_{fixed}, respectively.

Because both established approaches aim at computing large RCI under-approximations of the MRCI set, we solve the COP in (3.23) with initial time step $k_{x,0} = 1$ to construct large RCI sets as well. In particular, we use the presented cost functions in Subsection 3.5.4, which have also been used for generating the plots in Fig. 3.10. Moreover, both established methods solve a sequence of SDP problems to maximize the size of the RCI set. To detect convergence, we terminate the algorithms as soon as the difference between two consecutive cost function values is smaller than 10^{-5}. In addition, to obtain an initial feasible solution for the approach in [134], we choose the two scalar parameters ψ_1 and ψ_2 to be 100.

In Fig. 3.11, we show the evolution of the RCI set volumes with respect to the volume of a tight RCI under-approximation of the MRCI set, which is obtained by executing Alg. 2.2 with convergence tolerance $\epsilon = 10^{-10}$. When choosing $G_{\text{fixed}} = I \in \mathbb{R}^{2\times 2}$, the COP in (3.23) is infeasible and, thus, the volumes of our RCI sets are zero. In contrast to our generator-scaling-based approach, both methods in [132, 134] achieve feasibility for $G_{\text{fixed}} = I$ by optimizing the orientations. Nevertheless, when increasing the number of columns of G_{fixed} and rows of

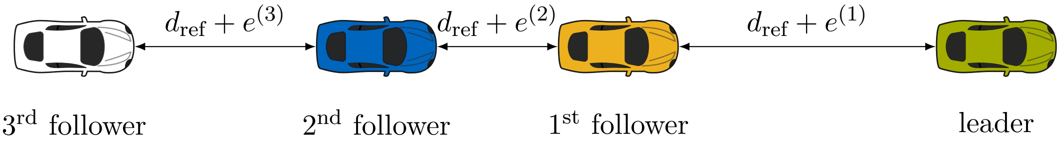

$d_{\text{ref}} + e^{(3)}$
$d_{\text{ref}} + e^{(2)}$
$d_{\text{ref}} + e^{(1)}$
3rd follower
2nd follower
1st follower
leader

Figure 3.12: Vehicle platooning system. The safe reference distance d_{ref} and the relative position errors between two vehicles are shown.

$P_{[132,134]}$, respectively, our approach outperforms the other two unless the simple enclosed multidimensional interval approximation is chosen as cost function. When adding more columns to an existing G_{fixed}, e.g., by doubling the number of uniformly distributed samples, the volumes of our approach are typically increasing. This increase shows that our cost function heuristics are reasonable. In contrast to this increase in volume, the approach in [134] achieves its highest volume when $P_{[132,134]}$ has only four rows, which might be caused by the used linearization techniques [134]. Thus, increasing the complexity of the RCI set does not necessarily result in an increase in volume, which can also be observed in Fig. 3.11 for the approach in [132].

3.6.3 Vehicle Platooning System

To demonstrate the applicability of our safe set approaches to a larger system, we consider a vehicle platooning system with nine states and three inputs [168], which is briefly summarized subsequently. The dynamics corresponding to the relative motion of the i^{th} following vehicle with $i \in \mathbb{N}_{[1,3]}$ and its vehicle ahead is

$$\ddot{e}^{(i)}(t) = a^{(i-1)}(t) - a^{(i)}(t), \tag{3.24}$$

where the relative position error $e^{(i)} \in \mathbb{R}$ denotes the difference between the two vehicles and a given safe reference distance $d_{\text{ref}} \in \mathbb{R}_{>0}$, as illustrated in Fig. 3.12. In addition, $a^{(i)} \in \mathbb{R}$ corresponds to the i^{th} effective acceleration described by the drivetrain dynamics

$$\dot{a}^{(i)}(t) = -\frac{1}{T_i} a^{(i)}(t) + \frac{1}{T_i} u^{(i)}(t), \tag{3.25}$$

where $T_i \in \mathbb{R}_{>0}$ represents a time constant that is assumed to be 0.5 s for all $i \in \mathbb{N}_{[1,3]}$ and $u^{(i)} \in \mathbb{R}$ is the i^{th} control input. In addition, the acceleration of the leading vehicle $a^{(0)}$ is assumed to be an unknown but bounded disturbance. In summary, the state of the platoon is described by $x = \begin{bmatrix} e^{(1)} & \dot{e}^{(1)} & a^{(1)} & e^{(2)} & \dot{e}^{(2)} & a^{(2)} & e^{(3)} & \dot{e}^{(3)} & a^{(3)} \end{bmatrix}^T$, the control input is $u = \begin{bmatrix} u^{(1)} & u^{(2)} & u^{(3)} \end{bmatrix}^T$, and the state disturbance is $w = \begin{bmatrix} 0 & a^{(0)} & 0 & \dots & 0 \end{bmatrix}^T$.

The sampling period is $\Delta t = 0.1\,\text{s}$, and the state, input, and disturbance bounds are presented in Table 3.1. Because all relative position errors are bounded by $[-10, 10]\,\text{m}$, the distance between two following vehicles is guaranteed to be within $[0, 20]\,\text{m}$ when setting $d_{\text{ref}} = 10\,\text{m}$. Then, compared to a nonplatooning scenario, the air drag of the following vehicles is reduced and the fuel consumption is decreased [169].

Because vehicle-to-vehicle communication is assumed [170, 171], a central controller with stabilizing feedback matrix $K \in \mathbb{R}^{3\times 9}$ can be designed using a linear matrix inequality (LMI)-

Table 3.1: State, input, and disturbance bounds of vehicle platooning system.

Variables	Bounds
$e^{(1)}, e^{(2)}, e^{(3)}$	$[-10,\, 10]\,\mathrm{m}$
$\dot{e}^{(1)}, \dot{e}^{(2)}, \dot{e}^{(3)}$	$[-5,\, 5]\,\frac{\mathrm{m}}{\mathrm{s}}$
$a^{(1)}, a^{(2)}, a^{(3)}$	$[-8,\, 8]\,\frac{\mathrm{m}}{\mathrm{s}^2}$
$u^{(1)}, u^{(2)}, u^{(3)}$	$[-8,\, 8]\,\frac{\mathrm{m}}{\mathrm{s}^2}$
$a^{(0)}$	$[-2,\, 2]\,\frac{\mathrm{m}}{\mathrm{s}^2}$

based controller synthesis [168], which results in

$$K = \begin{bmatrix} 0.8025 & 2.4340 & -0.7877 & -0.4099 & 0.2135 & -0.0225 & -0.0971 & 0.1813 & -0.0473 \\ 0.4359 & 1.9070 & -0.0377 & 0.5968 & 1.8129 & -0.6198 & -0.2975 & 0.0647 & -0.0398 \\ 0.3566 & 1.7865 & -0.0482 & 0.4236 & 1.6284 & -0.0438 & 0.6363 & 1.5360 & -0.5678 \end{bmatrix}. \tag{3.26}$$

We also want to mention that the overall system matrix $(A + BK) \in \mathbb{R}^{9\times 9}$ is a sparse matrix. However, all entries of the matrix exponential $e^{(A+BK)\Delta t}$ are nonzero, making the exploitation of the sparsity challenging. In addition, we ignore the underlying structure and consider the system as a black box because we want to show the applicability of our robust control approach to large-scale systems.

In Fig. 3.13, we visualize the small safe sets that are obtained by following the approaches presented in Section 3.4 for the scaling point $\mathring{x} = \texttt{center}\,(\mathcal{X})$. In particular, we execute Alg. 3.1 with convergence tolerance $\epsilon = 0.01$ to obtain a small safe set $\mathcal{S}_{Kx}$, which takes 0.4 s. In addition, we execute Alg. 3.2 with $o_{x,0} = 1$ to obtain a small safe parallelotope, which takes 3.8 s. Moreover, we solve the COP in (3.15) to construct a small RCI set, which takes 0.2 s. Ensuring feasibility of (3.15) requires the number of generators $\texttt{gen}\,(\mathcal{Z}_x(t_0))$ to be at least 26, i.e., the order of this small RCI set is similar to the one in Subsection 3.6.1. As hardly visible in the projection onto the $e^{(2)}$-$\dot{e}^{(2)}$-plane, the expansion of the RCI set is smaller than 10^{-3} in both of these dimensions. This small expansion might cause numerical issues when using this small RCI set in other robust control applications.

In Fig. 3.14, we visualize the large safe sets that are obtained by following some approaches presented in Section 3.5. In particular, we execute Alg. 3.3 with maximum interval radius $\epsilon = 0.01$ to obtain the scaled safe set $\mathcal{S}^\star_{Kx} \subseteq \mathcal{X}$, which takes 0.3 s. In addition, we use $\mathcal{S}^\star_{Kx}$ as safe terminal set $\mathcal{S}_{\text{safe}}$ when solving the COP in (3.22), which takes 30 s. In particular, we use the geometric mean of the generator scaling vector $s_{x,0} \in \mathbb{R}^{27}_{\geq 0}$ for the cost function $J_{\mathcal{Z}_x(t_0)}$ in (3.22a). Moreover, the fixed generator matrix $G_{\text{fixed}} \in \mathbb{R}^{9\times 27}$ is chosen as the generator matrix of $\mathcal{S}^\star_{Kx}$, and the initial time step is $k_{x,0} = 30$. We also want to mention that the COP in (3.23) is infeasible when choosing the same G_{fixed} and $k_{x,0} = 1$, i.e., an RCI set is not computed successfully based on Proposition 3.15. This infeasibility indicates that enforcing robust control invariance is more sensitive to parameter choices than guaranteeing safety.

Similarly, we are unable to compute a DT RCI set using established techniques [40, 132, 134]. In particular, the exponential computational complexity of polytopic set operations prevents

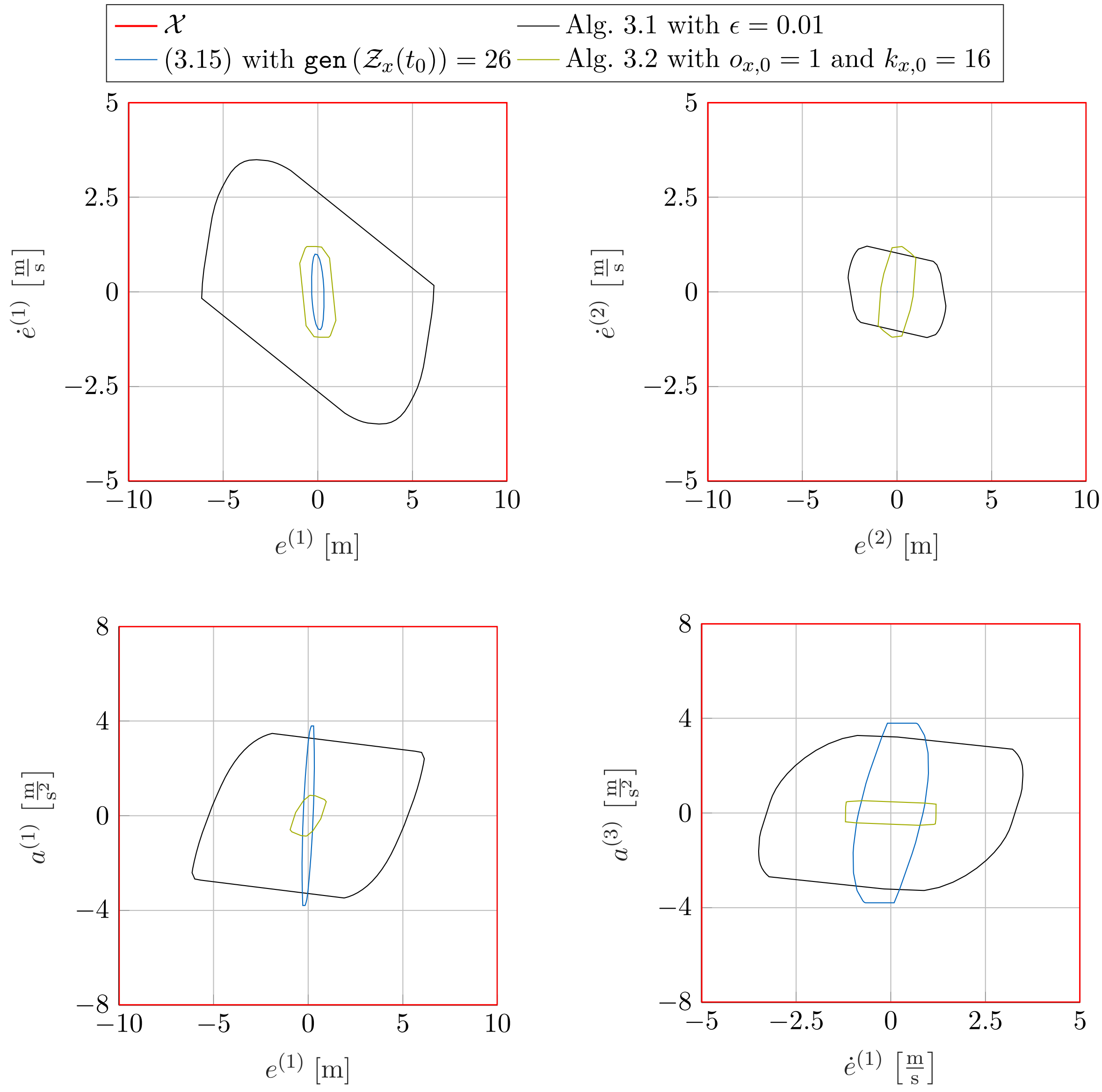


Figure 3.13: Two-dimensional projections of small safe sets of vehicle platooning system with scaling point $\mathring{x} = \texttt{center}\left(\mathcal{X}\right)$.

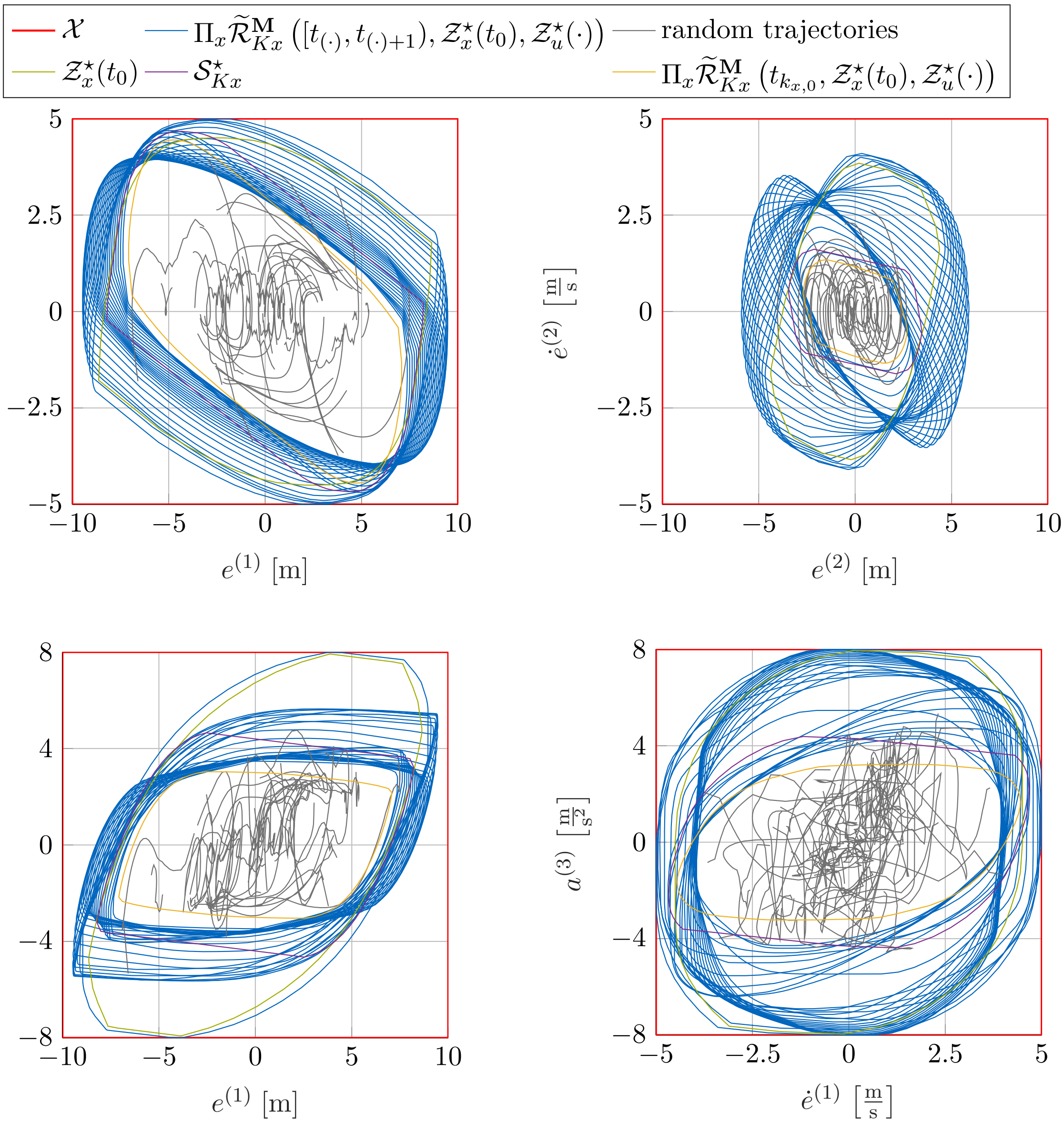


Figure 3.14: Two-dimensional projections of large safe sets of vehicle platooning system, which are obtained by executing Alg. 3.3 and by solving the COP in (3.22) with safe terminal set $\mathcal{S}^\star_{Kx}$. In addition, reachable sets and random trajectories are shown.

the synthesis of an RCI under-approximation of the MRCI set [40], which is implemented by Alg. 2.2. To use the methods in [132, 134], we proceed analogously to Subsection 3.6.2, i.e., we use the transpose of G_{fixed} as $P_{[132,134]} \in \mathbb{R}^{27\times 9}$. Because a nine-dimensional set must be converted from half-space to vertex representation in [134], this method also suffers from a high computational complexity and an error is thrown by the used MATLAB toolbox [77]. In addition, the number of scalar inequalities scales exponentially with the dimension of the state space. Similarly, following the approach in [132] led to an initial infeasible solution, which can also be observed for smaller dimensional systems.

3.6.4 Chain of Mass-Spring-Damper Systems

To demonstrate the scalability of our safe set approaches, we consider a chain of $N \in \mathbb{N}_{>0}$ MSD systems [172, 173], which is briefly summarized subsequently. The dynamics corresponding to the i^{th} MSD system with $i \in \mathbb{N}_{[2,N-1]}$ is

$$\begin{aligned}\begin{bmatrix}\dot{x}^{(2i-1)}(t)\\ \dot{x}^{(2i)}(t)\end{bmatrix} &= \begin{bmatrix}0 & 1\\ -2k & -2d\end{bmatrix}\begin{bmatrix}x^{(2i-1)}(t)\\ x^{(2i)}(t)\end{bmatrix} + \begin{bmatrix}0\\ u^{(i)}(t)\end{bmatrix}\\ &+ \begin{bmatrix}0 & 0\\ k & d\end{bmatrix}\begin{bmatrix}x^{(2(i+1)-1)}(t) + x^{(2(i-1)-1)}(t)\\ x^{(2(i+1))}(t) + x^{(2(i-1))}(t)\end{bmatrix} + \begin{bmatrix}w^{(2i-1)}(t)\\ w^{(2i)}(t)\end{bmatrix},\end{aligned}$$

where $k = 3$ and $d = 3$ are the spring and damper parameters of all systems. Similarly, the dynamics corresponding to the first and the last MSD systems are

$$\begin{aligned}\begin{bmatrix}\dot{x}^{(1)}(t)\\ \dot{x}^{(2)}(t)\end{bmatrix} &= \begin{bmatrix}0 & 1\\ -k & -d\end{bmatrix}\begin{bmatrix}x^{(1)}(t)\\ x^{(2)}(t)\end{bmatrix} + \begin{bmatrix}0\\ u^{(1)}(t)\end{bmatrix} + \begin{bmatrix}0 & 0\\ k & d\end{bmatrix}\begin{bmatrix}x^{(3)}(t)\\ x^{(4)}(t)\end{bmatrix} + \begin{bmatrix}w^{(1)}(t)\\ w^{(2)}(t)\end{bmatrix}\\ \begin{bmatrix}\dot{x}^{(2N-1)}(t)\\ \dot{x}^{(2N)}(t)\end{bmatrix} &= \begin{bmatrix}0 & 1\\ -k & -d\end{bmatrix}\begin{bmatrix}x^{(2N-1)}(t)\\ x^{(2N)}(t)\end{bmatrix} + \begin{bmatrix}0\\ u^{(N)}(t)\end{bmatrix}\\ &+ \begin{bmatrix}0 & 0\\ k & d\end{bmatrix}\begin{bmatrix}x^{(2(N-1)-1)}(t)\\ x^{(2(N-1))}(t)\end{bmatrix} + \begin{bmatrix}w^{(2N-1)}(t)\\ w^{(2N)}(t)\end{bmatrix}.\end{aligned}$$

The sampling period is $\Delta t = 0.1\,\text{s}$, and the disturbance, state, and input constraint sets are $\mathcal{W} = [-0.1, 0.1]^{2N}$, $\mathcal{X} = [-1, 1]^{2N}$, and $\mathcal{U} = [-10, 10]^{N}$, respectively. Moreover, the stabilizing state feedback matrix $K \in \mathbb{R}^{N\times 2N}$ is obtained using LQR-based controller synthesis [155], where both state and input weighting matrices are identity matrices. Subsequently, we report the computation times of our small and large safe set approaches presented in Sections 3.4 and 3.5, respectively. As mentioned in Section 2.5, we always exclude the time for modeling a COP with YALMIP [66] when reporting the computation time for solving it.

Small Safe Sets

Because no time-consuming optimization problem is solved in Alg. 3.1, the corresponding computation times in Table 3.2 are always smaller than 1 s. In addition, when decreasing the convergence tolerance $\epsilon \in \mathbb{R}_{>0}$, the computation time and the corresponding time step k_{Kx} usually increase while the zonotope order $\texttt{order}\,(\mathcal{S}_{Kx})$ of the small safe set $\mathcal{S}_{Kx} \subseteq \mathcal{X}$ decreases.

Table 3.2: Small safe sets obtained by executing Alg. 3.1 with varying convergence tolerance ϵ for a chain of N MSD systems.

N	ϵ	computation time [s]	$\texttt{order}\,(\mathcal{S}_{Kx})$	k_{Kx}
2	10^{-1}	0.079	1	42
2	10^{-2}	0.029	1	60
2	10^{-3}	0.041	1	96
3	10^{-1}	0.026	1	42
3	10^{-2}	0.045	1	60
3	10^{-3}	0.120	1	96
4	10^{-1}	0.039	3	42
4	10^{-2}	0.070	2	60
4	10^{-3}	0.173	2	96
5	10^{-1}	0.061	10	42
5	10^{-2}	0.135	5	66
5	10^{-3}	0.280	3	97
6	10^{-1}	0.122	18	43
6	10^{-2}	0.251	13	71
6	10^{-3}	0.567	12	102
7	10^{-1}	0.207	21	43
7	10^{-2}	0.407	17	74
7	10^{-3}	0.747	16	104

This relationship holds because $\mathcal{S}_{Kx}$ becomes a tighter over-approximation of the DT mRPI set based on Lemma 3.8.

The results for computing small RCI sets obtained by solving (3.15) with scaling point $\mathring{x} = \texttt{center}\,(\mathcal{X})$ are shown in Table 3.3. As the dimension of the system increases, the number of generators $\texttt{gen}\,(\mathcal{Z}_x(t_0))$ of $\mathcal{Z}_x(t_0)$ must also be increased to ensure the feasibility of (3.15) based on Proposition 3.10. Nevertheless, the computation time for solving (3.15) scales moderately with the dimension of the state space $n_x = 2N$ when keeping $\texttt{gen}\,(\mathcal{Z}_x(t_0))$ constant, as can be seen in the penultimate column of Table 3.3. Moreover, the optimal cost $s^{\star}_{\mathcal{X}}$, which we minimize in (3.15), increases only slightly with the system dimension and seems to converge, as shown in the last column.

The results for computing small safe sets obtained by executing Alg. 3.2 with scaling point $\mathring{x} = \texttt{center}\,(\mathcal{X})$, initial time step $k_{x,0} = 3$, and varying zonotope order $o_{x,0} \in \mathbb{N}_{>0}$ are shown in Table 3.4. For all systems, it is possible to compute a small safe set whose zonotope order $o_{x,0}$ is only 1. In contrast to these simple parallelotopes, the zonotope order of the small RCI sets in Table 3.3 is always equal or greater than 4. This observation indicates that enforcing

Table 3.3: Small RCI sets obtained by solving (3.15) with scaling point $\mathring{x} = \texttt{center}(\mathcal{X})$ for a chain of N MSD systems.

N	min $\texttt{gen}(\mathcal{Z}_x(t_0))$ for feasibility	$\texttt{gen}(\mathcal{Z}_x(t_0)) = 100$	
		computation time [s]	$s^\star_{\mathcal{X}}$
2	16	0.184	0.130
3	38	0.229	0.142
4	56	0.419	0.144
5	68	0.634	0.145
6	84	7.837	0.145
7	98	2.974	0.146

Table 3.4: Small safe sets obtained by executing Alg. 3.2 with scaling point $\mathring{x} = \texttt{center}(\mathcal{X})$, initial time step $k_{x,0} = 3$, and varying zonotope order $o_{x,0}$ for a chain of N MSD systems.

N	$o_{x,0}$	computation time [s]	$s^\star_{\mathcal{X}}$
2	1	0.109	0.053
2	2	0.205	0.045
2	3	0.118	0.043
3	1	0.170	0.057
3	2	0.285	0.053
3	3	0.365	0.051
4	1	0.349	0.066
4	2	0.738	0.058
4	3	1.028	0.050
5	1	1.306	0.083
5	2	1.775	0.049
5	3	4.857	0.046
6	1	4.887	0.094
6	2	7.684	0.067
6	3	11.475	0.067
7	1	3.109	0.116
7	2	6.345	0.075
7	3	27.862	0.066

Table 3.5: Large safe sets obtained by executing Alg. 3.3 with varying maximum interval radius ϵ and small safe set $\mathcal{S}_{Kx}$ from Table 3.2 for a chain of N MSD systems.

N	ϵ	$\texttt{order}\,(\mathcal{S}_{Kx})$	k_{Kx}	computation time [s]	$\texttt{min}\,(\mathcal{I})$
2	10^{-1}	1	42	0.017	1.475
2	10^{-2}	1	60	0.052	1.826
2	10^{-3}	1	96	0.155	1.859
3	10^{-1}	1	42	0.016	1.165
3	10^{-2}	1	60	0.094	1.420
3	10^{-3}	1	96	0.380	1.451
4	10^{-1}	3	42	0.001	1.000
4	10^{-2}	2	60	0.115	1.134
4	10^{-3}	2	96	0.549	1.161
5	10^{-1}	10	42	0.001	1.000
5	10^{-2}	5	66	0.086	1.018
5	10^{-3}	3	97	0.239	1.005
6	10^{-1}	18	43	0.002	1.000
6	10^{-2}	13	71	0.003	1.000
6	10^{-3}	12	102	1.119	1.025
7	10^{-1}	21	43	0.004	1.000
7	10^{-2}	17	74	0.313	1.013
7	10^{-3}	16	104	1.219	1.015

robust control invariance is more complicated than guaranteeing safety, similar to the small safe set results in Subsection 3.6.1. Similar to the computation times in Table 3.3, the ones in Table 3.4 also scale moderately with the state space dimension while the optimal cost $s^\star_{\mathcal{X}}$ decreases when increasing the zonotope order $o_{x,0}$.

Large Safe Sets

The results for computing large safe sets obtained by executing Alg. 3.3 are shown in Table 3.5, where the maximum interval radii are identical to the convergence tolerances in Table 3.2. Thus, the third and fourth columns of Table 3.5 are identical to the last two columns of Table 3.2. Because a simple binary search is performed in Alg. 3.3, the reported computation times in the fifth column of Table 3.5 are always smaller than 1.3 s. Thus, Alg. 3.3 is well suited for quickly computing safe sets of large-scale systems.

For the following large safe set approaches that solve an optimization problem, we choose the concave cost function $J_{\mathcal{Z}_x(t_0)}$ to be $\log(\det(S))$ with $S \in \mathbb{R}^{2N\times 2N}$ being the symmetric positive definite shape matrix of the approximative ellipsoid, as explained in Subsection 3.5.4. In addition, $\mathcal{Z}^{\star}_{x,(3.15)}(t_0)$ denotes the small RCI set obtained by solving (3.15) with scaling point $\mathring{x} = \texttt{center}(\mathcal{X})$ and $\texttt{gen}\left(\mathcal{Z}^{\star}_{x,(3.15)}(t_0)\right) = 100$. Thus, the construction of $\mathcal{Z}^{\star}_{x,(3.15)}(t_0)$ corresponds to the last two columns of Table 3.3.

The results for computing large safe parallelotopes obtained by solving (3.20) are shown in Table 3.6, where the square fixed generator matrix $G_{\text{fixed}} \in \mathbb{R}^{2N \times 2N}$ equals the generator matrix of a safe set of reduced zonotope order. As shown in the fourth and last columns of Table 3.6 and proven in Theorem 3.13, the optimal cost $\log(\det(S^{\star}))$ is monotonically increasing with increasing initial time step $k_{x,0} \in \mathbb{N}_{>0}$. For instance, the optimal cost has converged for $N = 2$ with $k_{x,0} = 15$. In addition, the choice of the fixed generator matrix G_{fixed} influences the optimal cost only slightly, and the computation time moderately increases when increasing N or $k_{x,0}$.

The results for computing large RCI sets obtained by solving (3.23) with initial time step $k_{x,0} = 1$ are shown in Table 3.7. In contrast to the small RCI set results in Table 3.3, the minimum zonotope order that is required to ensure feasibility of (3.23) is equal or smaller than 2 for all systems, as shown in the second column of Table 3.7. This minimum zonotope order indicates that the large RCI set approach is less conservative than the small RCI set one, although a fixed generator matrix is used. As shown in the penultimate column of Table 3.7, the computation time for solving (3.23) to obtain large safe sets scales moderately with the state space dimension $n_x = 2N$. Moreover, the optimal costs $\log(\det(S^{\star}))$ in the last column of Table 3.7 are larger than the ones in Table 3.6, which is enabled, inter alia, by the bigger zonotope order of 3 instead of 1.

3.7 Summary

In this chapter, we have presented efficient methods for synthesizing zonotopic safe sets along with corresponding safety-preserving controllers, which ensure robust constraint satisfaction for an infinite time horizon. First, we have computed reachable sets when using a simple state feedback controller and a slightly more sophisticated disturbance feedback controller.

Based on the reachable set computations, we have proposed three different approaches for computing safe sets that are as small as possible: The first one determines a small safe set without solving an optimization problem but assumes a stabilizing state feedback matrix to be given. This assumption is not required for our second method, which computes a small RCI set whose number of generators must be large enough to ensure the feasibility of the corresponding COP. To prevent the number of generators of a small safe set from becoming too large, the third approach limits this number while using the disturbance feedback controller.

In addition to these small safe set methods, we have also proposed three different approaches for computing safe sets that are as large as possible: The first one determines a large safe set by uniformly scaling a small safe set without solving an optimization problem. By exploiting the superposition principle, the second method solves a COP whose cost monotonically increases with increasing horizon. Moreover, the third approach computes large safe sets by solving a COP that ensures the last reachable set is contained within a safe set.

Finally, we have demonstrated the effectiveness of our proposed safe set approaches using multiple numerical examples. In particular, to show the performance of all proposed approaches,

Table 3.6: Large safe sets obtained by solving (3.20) with varying initial time step $k_{x,0}$ for a chain of N MSD systems. The square fixed generator matrix $G_{\text{fixed}} \in \mathbb{R}^{2N \times 2N}$ equals the generator matrix of a safe set of reduced zonotope order.

N	$k_{x,0}$	$\texttt{reduce}\,(\mathcal{S}^{\star}_{Kx}, 1)$ with $\epsilon = 10^{-2}$		$\texttt{reduce}\left(\mathcal{Z}^{\star}_{x,(3.15)}(t_0), 1\right)$	
		computation time [s]	log(det($S^{\star}$))	computation time [s]	log(det($S^{\star}$))
2	5	0.168	0.321	0.043	0.363
2	10	0.109	0.695	0.092	0.681
2	15	0.232	0.840	0.193	0.787
2	20	0.424	0.840	0.289	0.787
3	5	0.158	0.381	0.137	0.461
3	10	0.579	0.672	0.442	0.692
3	15	0.980	0.785	0.933	0.785
3	20	1.385	0.785	1.297	0.788
4	5	0.326	0.243	0.354	0.311
4	10	0.995	0.511	1.074	0.489
4	15	2.522	0.678	2.450	0.581
4	20	3.266	0.678	3.386	0.598
5	5	0.824	0.383	0.857	0.495
5	10	2.299	0.604	2.529	0.650
5	15	5.915	0.728	6.260	0.721
5	20	5.968	0.734	6.605	0.736
6	5	0.921	0.349	0.981	0.439
6	10	3.289	0.560	3.279	0.598
6	15	7.378	0.630	7.690	0.679
6	20	12.382	0.639	13.631	0.704
7	5	1.373	0.295	1.425	0.372
7	10	5.096	0.497	4.858	0.532
7	15	14.448	0.580	14.089	0.616
7	20	25.602	0.609	26.819	0.647

Table 3.7: Large RCI sets obtained by solving (3.23) for a chain of N MSD systems. The fixed generator matrix $G_{\text{fixed}} \in \mathbb{R}^{2N \times \texttt{gen}(\mathcal{Z}_x(t_0))}$ equals the generator matrix of the zonotope $\texttt{reduce}\left(\mathcal{Z}^{\star}_{x,(3.15)}(t_0), \texttt{order}\left(\mathcal{Z}_x(t_0)\right)\right)$.

N	min $\texttt{order}\left(\mathcal{Z}_x(t_0)\right)$ for feasibility	$\texttt{order}\left(\mathcal{Z}_x(t_0)\right) = 3$	
		computation time [s]	$\log(\det(S^{\star}))$
2	1	0.122	0.817
3	1	0.140	0.841
4	1	0.349	0.730
5	1	0.634	0.790
6	2	0.981	0.730
7	2	1.481	0.688

we have considered a simple two-dimensional system such that no projection is required to visualize them along with their reachable sets. To demonstrate the scalability of our robust control methods, we have also used a chain of MSD systems. Because all of our proposed approaches use zonotopes as set representations and are based on scalable reachability analysis and convex optimization, they scale moderately with the dimension of the system. In the following chapter, we incorporate our large safe sets into one of the most popular control methods, namely, MPC.

4 Model Predictive Control

In this chapter, which is based on [47, 50], we propose a real-time robust output feedback model predictive control (MPC) approach that uses our large safe sets presented in Section 3.5 as terminal sets. After introducing the concept of MPC and reviewing the relevant literature in Section 4.1, we formulate the control goal of this chapter in Section 4.2. In addition, we present some preliminaries in Section 4.3. In Section 4.4, we propose our scalable robust output feedback dual-mode MPC approach for sampled-data systems that explicitly considers the computation time of the online optimal MPC problem. To demonstrate the effectiveness of our approach, we consider two numerical examples in Section 4.5. Finally, we summarize this chapter in Section 4.6.

4.1 Introduction and State of the Art

Over the past few decades, MPC has become a very successful approach for controlling complex dynamical systems in industry [174, 175]. For instance, MPC is used in the automotive industry [176], in health care [177], in power electronics [178], in air conditioning systems [179], and in the field of finance [180]. Thus, MPC is also known as "most popular control". Its great popularity can be attributed to its simple concept and ability to effectively deal with state and input constraints [41–43, 128, 181], which is impossible with linear-quadratic regulator (LQR)-based controller synthesis [155].

Typically, (implicit) MPC follows a three-step iterative online scheme: At each sampling time $t_k = k\Delta t$, it

1. measures/estimates the current state of the system;
2. solves an online optimization problem on a moving horizon of $N \in \mathbb{N}_{>0}$ to obtain an optimal input sequence, as illustrated in Fig. 4.1; and
3. applies only the first input of this sequence to the system until the next sampling time.

Because of the moving prediction window, MPC is also known as moving or receding horizon control. Instead of iteratively solving an optimization problem online, explicit MPC solves a reformulation of this problem offline as a function of the state [182, 183]. By solving this offline multi-parametric programming problem once [77], the dependence of the controller on the state is given explicitly instead of defining it implicitly by the online optimization problem. Although explicit MPC is often used to achieve small sampling periods on embedded platforms, it is usually limited to small problems due to its high computational complexity [184].

When using MPC in safety-critical applications, it is crucial to formally guarantee robustness against disturbances. Thus, robust MPC approaches that ensure robust constraint satisfaction despite unknown but bounded disturbances are required [160, 185–187]. Initially, robust MPC

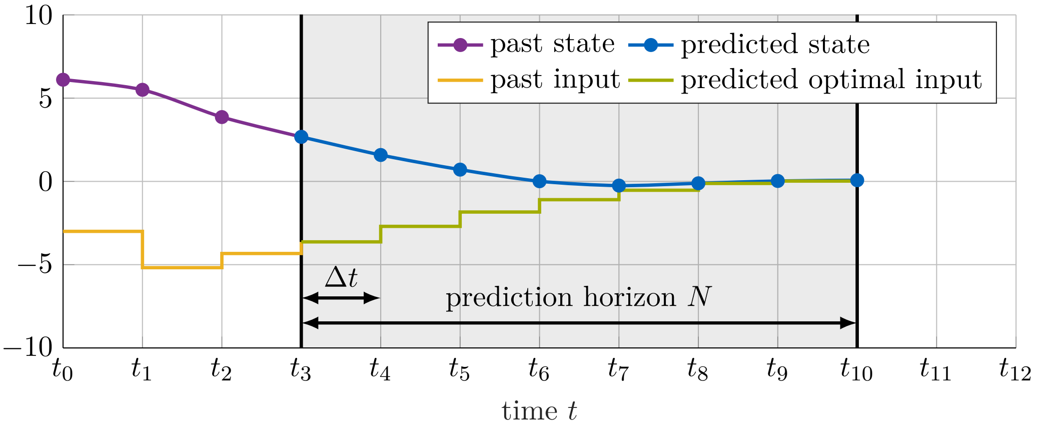


Figure 4.1: MPC concept. At current initial time t_3, an optimization problem is solved on a moving horizon of $N = 7$.

has been applied to linear systems with state feedback control. Because a min-max optimization over general feedback controllers easily becomes impractical due to its high computational complexity [188–191], tube-based MPC approaches have been proposed [192, 193]. The key idea is to ensure that the state of the system stays within a tube surrounding the nominal trajectory that satisfies the state and input constraints. Thus, by tightening the constraints appropriately, only the disturbance-free nominal prediction model is required for online computations, whereas computationally expensive set operations are performed offline. In addition, many extensions and generalizations of the traditional tube-based MPC approach exist [194–196].

Depending on the application, these robust MPC approaches have also been successfully applied in real time. For instance, a system of oscillating masses with sampling periods in the order of 1 ms and the trajectory tracking of a mobile robot with a sampling period of 350 ms are studied [197, 198]. To enable real-time computations, set-based reachability analysis can also be used in a second step to formally verify that the solution of an approximate optimal MPC problem robustly satisfies the state and input constraints [96].

When using MPC in real-world environments, the exact measurement of the system state is typically unavailable. Thus, in addition to state feedback MPC, output feedback MPC approaches that use noisy measurements of the system have been proposed. Based on these measurements, a simple linear Luenberger state observer is often used in robust output feedback MPC to estimate the inaccessible state of the system [199–202]. In addition, there also exist more sophisticated methods that incorporate set-membership and moving horizon estimation into MPC [203–207]. These methods use not only the *a priori* known error sets but also the noisy measurements obtained online to provide more accurate state estimates at the expense of increased computational complexity.

Most literature on robust MPC deals with discrete-time (DT) models. Thus, constraint satisfaction is only guaranteed at discrete time steps while assuming that the solution of the optimal MPC problem can be obtained instantaneously. However, most real-world control applications can be accurately modeled by sampled-data systems, where a continuous-time (CT) physical plant is controlled by a DT digital controller. Therefore, robust constraint satisfaction must also be guaranteed between two discrete sampling times. Because of its high practical relevance, the consideration of sampled-data robust MPC approaches has gained popularity over the past few years [208, 209]. For instance, the approach in [193] has been extended to deal with sampled-data systems in [210]. However, in this extension, the unknown but bounded disturbance is assumed to be constant between two sampling times, which is quite unrealistic.

Because most robust MPC approaches suffer from high computational complexity, centralized methods are usually unsuitable for handling large-scale systems. For instance, the computation of a polytopic invariant terminal set is difficult for large-scale systems because performing the required polytopic set operations becomes intractable [211], as shown in Table 2.1. Typically, these problems can only be overcome if the original control problem can be decomposed into simpler ones, as in distributed and decentralized MPC [146, 212, 213]. Instead of polytopes, ellipsoids are also widely used in MPC approaches to analyze the evolution of the disturbance based on scalable ellipsoidal reachability analysis [97,98]. However, ellipsoids under-approximate typical constraints represented by multidimensional intervals in high dimensions very poorly [129, 142], increasing conservativeness.

It is clear from the presented literature review that it is an unresolved issue to provide a scalable, real-time, robust output feedback MPC approach for sampled-data systems. In this chapter, we address this issue by incorporating our efficient large safe set computations proposed in Section 3.5 into robust output feedback MPC. In particular, we use our large safe sets as terminal sets in the optimal MPC problem, which is iteratively solved online on a moving horizon, as shown in Fig. 4.1. Because the computation time for solving this convex optimization problem (COP) is nonnegligible, we explicitly consider it in our reachability analysis to avoid invalidating the formal safety guarantees. In addition, we use a simple linear Luenberger state observer to estimate the inaccessible state of the system. Before we present our scalable, real-time, robust output feedback MPC approach, we formulate the control problem of this chapter in the following section.

4.2 Problem Formulation

In this chapter, we consider a CT, linear time-invariant (LTI) system that evolves according to (2.20), i.e., it is compactly represented by the model $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$. Instead of making the simplifying assumption that the state of the system is directly measurable, as done in Chapter 3, only noisy measurements

$$y_D(t_k) = C_D x(t_k) + v_D(t_k) \tag{4.1}$$

are available at periodic sampling times t_k, where $C_D \in \mathbb{R}^{n_y \times n_x}$ is the output matrix and $v_D(t_k) \in \mathbb{R}^{n_y}$ is the unknown output disturbance at t_k. Similar to the zonotopic state disturbance set $\mathcal{W}$, the output disturbance set $\mathcal{V}_D = \langle c_{\mathcal{V}_D}, G_{\mathcal{V}_D} \rangle_Z \subset \mathbb{R}^{n_y}$ contains the unknown output disturbance at all times, i.e., $v_D(\cdot) \in \mathcal{V}_D$. As for the state and input constraints in (3.1), in this chapter, the system in (2.20) is also constrained by

$$x(\cdot) \in \mathcal{X} \tag{4.2a}$$

$$u(\cdot) \in \mathcal{U}, \tag{4.2b}$$

where $\mathcal{X} = \langle H_{\mathcal{X}}, h_{\mathcal{X}} \rangle_P \subset \mathbb{R}^{n_x}$ and $\mathcal{U} = \langle H_{\mathcal{U}}, h_{\mathcal{U}} \rangle_P \subset \mathbb{R}^{n_u}$ are the state and input constraint sets, respectively. Because $\mathcal{W}$, $\mathcal{V}_D$, $\mathcal{X}$, and $\mathcal{U}$ are usually represented by multidimensional intervals, they can be easily expressed in both half-space and generator representations using (2.5) and (2.6). If other set representations are chosen, tight polytopic under-approximations of $\mathcal{X}$ and $\mathcal{U}$ as well as zonotopic over-approximations of $\mathcal{W}$ and $\mathcal{V}_D$ must be computed such that the formal safety guarantees remain valid.

To formulate a meaningful sampled-data control problem, we assume that (A_D, B_D, C_D) is stabilizable and detectable, where $A_D \in \mathbb{R}^{n_x \times n_x}$ and $B_D \in \mathbb{R}^{n_x \times n_u}$ are the DT system and input matrices defined in (2.27). Without loss of generality, we also assume that $\mathcal{W}$, $\mathcal{V}_D$, $\mathcal{X}$, and $\mathcal{U}$ contain the origin. Then, the control goal of this chapter is to steer the system in (2.20) to a neighborhood of the origin while satisfying the safety constraints in (4.2).

4.3 Preliminaries

In this section, we present our clocked digital state estimator and controller. In addition, we compute reachable sets for times $t_k + t$ with $t \in \mathbb{R}_{\geq 0}$ based on the noisy measurement $y_D(t_k)$ obtained at t_k. Finally, we determine large safe sets along with corresponding safety-preserving controllers by slightly modifying the approaches proposed in Section 3.5.

4.3.1 State Estimation and Control

The noisy measurements in (4.1) are used to compute state estimates at sampling times. To obtain a DT state estimator, we consider the exactly discretized system dynamics in (2.28) with corresponding model $(A_D, B_D, \mathcal{W}_D, \mathrm{DT})$. Instead of using a computationally demanding set-membership or strip-based observer [91, 214], we choose a simple linear Luenberger state estimator

$$\hat{x}(t_{k+1}) = A_D \hat{x}(t_k) + B_D u(t_k) + L\big(y_D(t_k) - C_D \hat{x}(t_k)\big),$$

where $L \in \mathbb{R}^{n_x \times n_y}$ is a stabilizing output injection matrix such that all eigenvalues of $A_D - LC_D$ are contained in the open complex unit disc [215]. As a result, the dynamics of the state estimation error $\xi(t_k) = x(t_k) - \hat{x}(t_k)$ and the corresponding error sets are

$$\xi(t_{k+1}) = (A_D - LC_D)\xi(t_k) + w_D(t_k) - Lv_D(t_k) \tag{4.3a}$$

$$\mathcal{E}(t_{k+1}) = (A_D - LC_D)\mathcal{E}(t_k) \oplus \mathcal{W}_D \oplus (-L\mathcal{V}_D), \tag{4.3b}$$

where $w_D(t_k) \in \mathcal{W}_D$ is the discretized state disturbance at t_k, and the initial error $\xi(t_0) \in \mathbb{R}^{n_x}$ lies within the zonotopic initial error set $\mathcal{E}(t_0) \subset \mathbb{R}^{n_x}$ containing the origin. Thus, $\xi(t_k) \in \mathcal{E}(t_k)$ for any $k \in \mathbb{N}$, resulting in

$$x(t_k) \in \{\hat{x}(t_k)\} \oplus \mathcal{E}(t_k) \tag{4.4a}$$

$$\hat{x}(t_k) \in \{x(t_k)\} \oplus \big(-\mathcal{E}(t_k)\big). \tag{4.4b}$$

We also want to emphasize that we do not require $\mathcal{E}(t_0)$ to be a robust positively invariant (RPI) set [37, 39], which would imply $\mathcal{E}(t_{k+1}) \subseteq \mathcal{E}(t_k)$ and significantly simplify the reachability analysis [200, 202]. Subsequently, we use the state estimate $\hat{x}(t_k)$ for computing the output feedback control input.

If $\hat{x}(t_k)$ is guaranteed to lie within $\mathcal{Z}_{\hat{x}}(t_k) = \langle c_{\hat{x}}(t_k), G_{\hat{x}}(t_k)\rangle_Z \subset \mathbb{R}^{n_x}$, it can be expressed by

$$\hat{x}(t_k) = c_{\hat{x}}(t_k) + G_{\hat{x}}(t_k)\lambda_{\hat{x},k},$$

similar to (3.2). The not necessarily unique parameter vector $\lambda_{\hat{x},k} \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z}_{\hat{x}}(t_k))}$ with $|\lambda_{\hat{x},k}| \leq \mathbf{1}$ can be obtained by solving (2.1) for $\hat{x}(t_k)$ and $\mathcal{Z}_{\hat{x}}(t_k)$, similar to (3.3). By generalizing the piecewise constant state feedback controller in (3.4), we obtain the piecewise constant output feedback control law

$$u(t) = K\hat{x}(t_k) + c_u(t_k) + G_u(t_k)\lambda_{\hat{x},k} \quad \text{for } t \in [t_k, t_{k+1}), \tag{4.5}$$

where $\mathcal{Z}_u(t_k) = \langle c_u(t_k), G_u(t_k)\rangle_Z \subset \mathbb{R}^{n_u}$ with generator matrix $G_u(t_k) \in \mathbb{R}^{n_u \times \mathtt{gen}(\mathcal{Z}_{\hat{x}}(t_k))}$ is the correction input zonotope at sampling time t_k, and $K \in \mathbb{R}^{n_u \times n_x}$ is a stabilizing feedback matrix such that all eigenvalues of $A_D + B_D K$ are contained in the open complex unit disc. Because (A_D, B_D) is assumed to be stabilizable, a stabilizing K can be easily obtained, e.g., by LQR-based controller synthesis [155].

In MPC, we iteratively solve an optimal control problem on a moving horizon at each sampling time t_k, as illustrated in Fig. 4.1 for t_3. To extend our notation to account for these iterations, we use $(t_i|t_k)$ with $i \in \mathbb{N}$ to refer to the prediction for the time $t_k + t_i$ made at t_k. In particular, we denote the prediction of the correction input zonotope sequence that is optimized online based on the state estimate $\hat{x}(t_k)$ at t_k by $\mathcal{Z}_u(\cdot|t_k) = \langle c_u(\cdot|t_k), G_u(\cdot|t_k)\rangle_Z$. Analogously, we denote the predictions of the future state and input trajectories based on $\hat{x}(t_k)$ by $x(\cdot|t_k)$ and $u(\cdot|t_k)$, respectively, where $x(t_0|t_k) = x(t_k)$. In Fig. 4.1, these state and input trajectory predictions are shown in blue and green, respectively.

4.3.2 Reachability Analysis

Based on the state estimate $\hat{x}(t_k) \in \mathcal{Z}_{\hat{x}}(t_k) = \langle c_{\hat{x}}(t_k), G_{\hat{x}}(t_k)\rangle_Z$ at t_k, we predict reachable sets when using the output feedback controller in (4.5) for arbitrary future sampling times t_i and time intervals $[t_i, t_{i+1})$ with $i \in \mathbb{N}$. To account for the piecewise constant control law in (4.5), we compute reachable sets for consecutive time steps of size Δt until the specified time t_i is reached. Similar to (3.5), we introduce the following recursively defined set sequence for $\mathcal{Z}_{\hat{x}}(t_k)$, $\mathcal{Z}_u(\cdot|t_k)$, and $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$:

$$\begin{aligned}
&\widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left(t_0, \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \mathcal{E}(\cdot)\right) \\
&\quad = \left\langle \begin{bmatrix} c_{\hat{x}}(t_k) \\ Kc_{\hat{x}}(t_k) + c_u(t_0|t_k) \end{bmatrix}, \begin{bmatrix} G_{\hat{x}}(t_k) \\ KG_{\hat{x}}(t_k) + G_u(t_0|t_k) \end{bmatrix} \right\rangle_Z \oplus \left\langle \begin{matrix} \mathcal{E}(t_k) \\ \{\mathbf{0}\} \end{matrix} \right\rangle_Z
\end{aligned} \tag{4.6a}$$

$$\begin{aligned}
&\langle c_x(t_i|t_k), G_x(t_i|t_k)\rangle_Z \\
&\quad = \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{over}}\left(\Delta t, \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left(t_{i-1}, \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \mathcal{E}(\cdot)\right)\right)
\end{aligned} \tag{4.6b}$$

$$\begin{aligned}
&\widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left(t_i, \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \mathcal{E}(\cdot)\right) \\
&\quad = \left\langle \begin{bmatrix} c_x(t_i|t_k) \\ Kc_x(t_i|t_k) + c_u(t_i|t_k) \end{bmatrix}, \begin{bmatrix} G_x(t_i|t_k) \\ KG_x(t_i|t_k) + \begin{bmatrix} G_u(t_i|t_k) & \mathbf{0} \end{bmatrix} \end{bmatrix} \right\rangle_Z \oplus \left\langle \begin{matrix} \{\mathbf{0}\} \\ -K\mathcal{E}(t_{k+i}) \end{matrix} \right\rangle_Z .
\end{aligned} \tag{4.6c}$$

Similar to Theorem 3.2, we prove that the sets in (4.6) are over-approximating the augmented reachable sets of $\mathbf{M}$ when using the controller in (4.5), as shown in the following proposition.

Proposition 4.1 (Set Propagation using Output Feedback Control): For all $\hat{x}(t_k) \in \mathcal{Z}_{\hat{x}}(t_k)$, applying the output feedback controller in (4.5) to $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$ results in

$$\begin{bmatrix} x(t_i|t_k) \\ u(t_i|t_k) \end{bmatrix} \in \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left(t_i, \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \mathcal{E}(\cdot)\right),$$

where $i \in \mathbb{N}$. ■

Proof. The structure of the output feedback controller in (4.5) is identical to the structure of the state feedback controller in (3.4). In addition, by using the relations in (4.4), the proof by induction follows the same line of thoughts as Theorem 3.2. □

We have focused on performing reachability analysis for discrete sampling times t_i when using the output feedback controller in (4.5). Nevertheless, the state and input constraints in (4.2) must be satisfied not only at but also between sampling times. Thus, based on Proposition 4.1 and (2.22b), we compute reachable sets for an arbitrary time interval $[t_i, t_{i+1})$ with $i \in \mathbb{N}$ according to

$$\widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left([t_i, t_{i+1}), \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \mathcal{E}(\cdot)\right) = \widetilde{\mathcal{R}}^{\mathbf{M}}_{\text{over}}\Big([0, \Delta t), \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left(t_i, \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \mathcal{E}(\cdot)\right)\Big). \quad (4.7)$$

Then, the projection of the over-approximative reachable set onto the original state and input space is obtained by $\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left(\cdot, \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \mathcal{E}(\cdot)\right)$ and $\Pi_u \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left(\cdot, \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \mathcal{E}(\cdot)\right)$, respectively. In summary, we can efficiently compute the set of states and inputs that are reachable for all state estimates $\hat{x}(t_k) \in \mathcal{Z}_{\hat{x}}(t_k)$ at t_k when applying the output feedback controller in (4.5) to $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$.

Similar to Theorem 3.6, we can also separate the reachable sets in (4.6) and (4.7) into controllable and uncontrollable parts based on the superposition principle. To enable formal safety guarantees, we separate all involved set operations in an over-approximative way, as presented in the following proposition.

Proposition 4.2 (Separation of Output Feedback Reachable Sets): The reachable sets in (4.6) and (4.7) for $\mathbf{M} = (A, B, \mathcal{W}, \mathrm{CT})$ can be separated by

$$\begin{aligned} \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left(t_i, \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \mathcal{E}(\cdot)\right) &= \widetilde{\mathcal{R}}^{(A,B,\{\mathbf{0}\},\mathrm{CT})}_{K\hat{x}}\left(t_i, \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \{\mathbf{0}\}\right) \\ &\quad \oplus \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left(t_i, \{\mathbf{0}\}, \{\mathbf{0}\}, \mathcal{E}(\cdot)\right) \\ \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left([t_i, t_{i+1}), \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \mathcal{E}(\cdot)\right) &\subseteq \widetilde{\mathcal{R}}^{(A,B,\{\mathbf{0}\},\mathrm{CT})}_{K\hat{x}}\left([t_i, t_{i+1}), \mathcal{Z}_{\hat{x}}(t_k), \mathcal{Z}_u(\cdot|t_k), \{\mathbf{0}\}\right) \\ &\quad \oplus \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left([t_i, t_{i+1}), \{\mathbf{0}\}, \{\mathbf{0}\}, \mathcal{E}(\cdot)\right). \end{aligned}$$ ■

Proof. The structure of the output feedback controller in (4.5) is identical to the structure of the state feedback controller in (3.4). In addition, all operations required to compute the reachable sets in (4.6) and (4.7) are identical to those in Theorem 3.6. Therefore, the proof follows the same line of thoughts as Theorem 3.6. □

In the following subsection, we use the presented reachable set computations to propose an efficient output feedback control method to construct large safe sets based on the approaches in Section 3.5.

4.3.3 Safe Sets

Ensuring the satisfaction of the safety constraints in (4.2) is typically achieved by using the controller $u(t) = K\hat{x}(t_k)$ for $t \in [t_k, t_{k+1})$ and constructing a polytopic RPI set [37,39]. However, the underlying set operations become intractable for large-scale systems, as shown in Table 2.1.

To overcome this scalability problem, we have proposed efficient approaches in Section 3.5 for computing large zonotopic safe sets based on the state feedback controller in (3.4). To incorporate the output feedback controller in (4.5) into these safe set computations, we need to consider the following main modifications:

- The reachable set sequence in (4.6) is used instead of (3.5).
- If $x(t_k) \in \mathcal{Z}_x(t_k)$ at t_k, $\mathcal{Z}_{\hat{x}}(t_k) = \mathcal{Z}_x(t_k) \oplus \big(-\mathcal{E}(t_k)\big)$ is guaranteed to contain $\hat{x}(t_k)$ based on (4.4b).
- Because we do not require the initial error set $\mathcal{E}(t_0) \subset \mathbb{R}^{n_x}$ to be an RPI set, we compute time-variant optimal safe sets $\mathcal{Z}_x^\star(t_k) \subseteq \mathcal{X}$ with corresponding optimal correction input zonotope sequence $\mathcal{Z}_u^{\star,\circ}(\cdot|t_k)$.

By performing these slight modifications, we compute optimal large safe sets $\mathcal{Z}_x^\star(t_k) \subseteq \mathcal{X}$ along with corresponding optimal correction input zonotope sequence $\mathcal{Z}_u^{\star,\circ}(\cdot|t_k)$, which guarantee satisfaction of the safety constraints in (4.2) if $x(t_k) \in \mathcal{Z}_x^\star(t_k)$ or $\hat{x}(t_k) \in \mathcal{Z}_x^\star(t_k) \oplus \big(-\mathcal{E}(t_k)\big)$. In the following section, these large safe sets are used as terminal sets in our robust MPC approach.

4.4 Robust Output Feedback Model Predictive Control

In this section, we present our scalable, real-time, robust output feedback dual-mode MPC algorithm considering a finite prediction horizon of $N \in \mathbb{N}_{>0}$. In the first mode of our algorithm, we iteratively solve an optimal MPC problem on a moving horizon, as shown in Fig. 4.1. We switch to the second mode when the state of the system is guaranteed to lie within a large safe set, which is efficiently computed based on Section 3.5. In this second mode, the corresponding safety-preserving controller ensures robust constraint satisfaction at all future times.

Because solving an optimal MPC problem on a moving horizon must be achieved in real time, it is usually too computationally expensive to optimize over general zonotopes online. To overcome this problem, we slightly simplify the controller in (4.5) that is used for the online optimal MPC optimizations. In particular, we set the correction input generator matrix $G_u(t_i|t_k)$ with $i \in \mathbb{N}_{[0,N-1]}$ to zero, i.e., we use $\mathcal{Z}_u(t_i|t_k) = \langle c_u(t_i|t_k), \mathbf{0}\rangle_Z = c_u(t_i|t_k)$ for all online optimal MPC computations.

In this section, we explicitly consider the computation time for solving the optimal MPC problem online, which is neglected by most existing robust MPC approaches. After introducing the contraction constraint and stating the optimal MPC problem, we present our robust output feedback dual-mode MPC algorithm that achieves the control goal of this chapter formulated in Section 4.2. Finally, we propose some simplifications that significantly reduce the online computational effort.

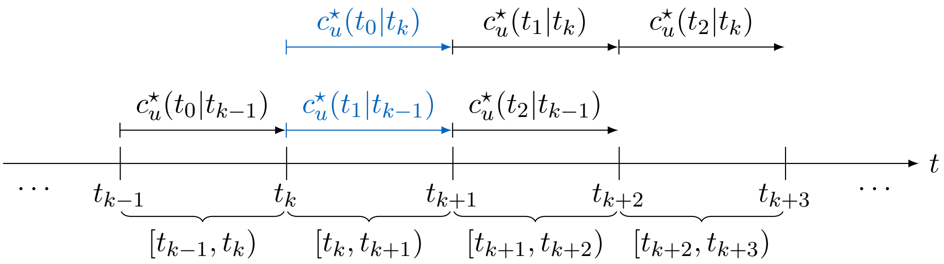


Figure 4.2: Optimal correction input sequences $c_u^\star(\cdot|t_{k-1})$ and $c_u^\star(\cdot|t_k)$ for $N = 3$. Based on (4.8), $c_u^\star(t_0|t_k) = c_u^\star(t_1|t_{k-1})$, which is shown in blue.

4.4.1 Computation Time Considerations

To ensure the satisfaction of the state and input constraints in (4.2), we explicitly consider the nonzero computation time for solving the optimal MPC problem online [216]. Thus, our formal safety guarantees remain valid despite such computational delays.

During the time interval $[t_{k-1}, t_k)$, we solve an optimal MPC problem to optimize the simplified correction input sequence $\mathcal{Z}_u(\cdot|t_{k-1}) = \langle c_u(\cdot|t_{k-1}), \mathbf{0}\rangle_Z = c_u(\cdot|t_{k-1})$, as shown in Fig. 4.2 for $N = 3$. At the next sampling time $t_k = k\Delta t$, the available computation time for optimizing $c_u(\cdot|t_{k-1})$ is elapsed. Then, we set

$$c_u^\star(t_0|t_k) = c_u^\star(t_1|t_{k-1}) \tag{4.8}$$

and apply the input $K\hat{x}(t_k) + c_u^\star(t_1|t_{k-1})$ to the system during $[t_k, t_{k+1})$ while we optimize $c_u(t_i|t_k)$ for $i \in \mathbb{N}_{[1,N-1]}$. If the optimization solver requires a longer time than the sampling period Δt to complete, we abort the optimization prematurely. If the result of this optimization is infeasible, we use the remainder of $c_u^\star(\cdot|t_{k-1})$ as a safe backup solution under the common assumption that an initial feasible solution $c_u^\star(\cdot|t_0)$ of the optimal MPC problem exists [203]. In this case, we set

$$c_u^\star(t_i|t_k) = \begin{cases} c_u^\star(t_1 + t_i|t_{k-1}) & \text{for } i \in \mathbb{N}_{[0,N-2]} \\ \emptyset & \text{for } i = N-1 \end{cases}. \tag{4.9}$$

If $c_u^\star(t_1|t_{k^{\star,\circ}-1})$ equals the empty set $\emptyset$ for some $k^{\star,\circ} \in \mathbb{N}$, we apply the inputs of (4.5) with optimal correction input zonotope sequence $\mathcal{Z}_u^{\star,\circ}(\cdot|t_{k^{\star,\circ}})$ that guarantee robust constraint satisfaction at all times $t \geq t_{k^{\star,\circ}}$ based on Subsection 4.3.3. However, this approach is only valid if $x(t_{k^{\star,\circ}}) \in \mathcal{Z}_x^\star(t_{k^{\star,\circ}})$. Thus, we add a terminal constraint to the optimal MPC problem that ensures the state at the end of the prediction horizon N to lie within a safe set, as visualized in Fig. 4.3. Before we propose the optimal MPC problem that incorporates this terminal constraint, we introduce a contraction constraint in the following subsection.

4.4.2 Contraction Constraint

Inspired by [95], we construct a simple contraction constraint such that the convergence of the state trajectory $x(\cdot)$ to the origin in finite time is ensured. Based on the contraction distances

$$d_x\big(t_i, \hat{x}(t_k), c_u(\cdot|t_k)\big) = \alpha + \mathtt{dist}\left(\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left(t_i, \{\hat{x}(t_k)\}, \langle c_u(\cdot|t_k), \mathbf{0}\rangle_Z, \mathcal{E}(\cdot)\right), \{\mathbf{0}\}\right), \tag{4.10}$$

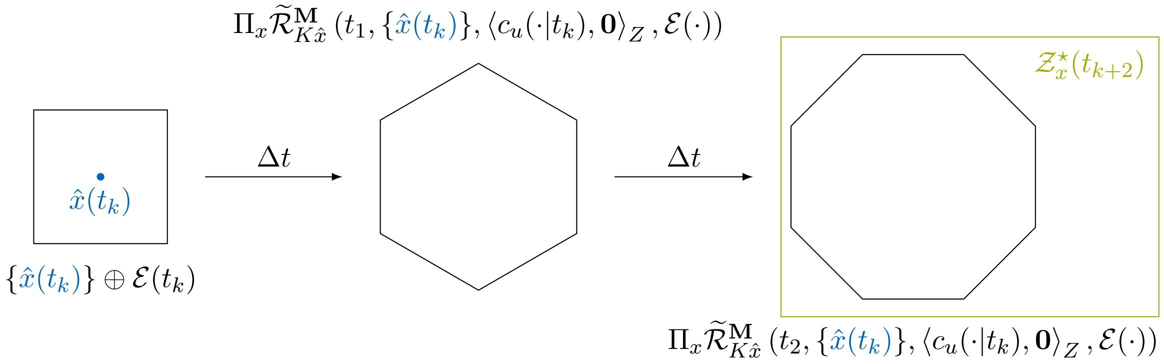

$\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}} \left(t_1, \{\hat{x}(t_k)\}, \langle c_u(\cdot|t_k), \mathbf{0}\rangle_Z, \mathcal{E}(\cdot)\right)$
$\mathcal{Z}^\star_x(t_{k+2})$
Δt
Δt
$\hat{x}(t_k)$
$\{\hat{x}(t_k)\} \oplus \mathcal{E}(t_k)$
$\Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}} \left(t_2, \{\hat{x}(t_k)\}, \langle c_u(\cdot|t_k), \mathbf{0}\rangle_Z, \mathcal{E}(\cdot)\right)$

Figure 4.3: Overview of optimal MPC problem for $N = 2$ based on the state estimate $\hat{x}(t_k)$ at t_k. Our reachability analysis guarantees that $x(t_{k+2}) \in \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}} \left(t_2, \{\hat{x}(t_k)\}, \langle c_u(\cdot|t_k), \mathbf{0}\rangle_Z, \mathcal{E}(\cdot)\right)$ lies within the safe set $\mathcal{Z}^\star_x(t_{k+2})$.

where $\alpha \in \mathbb{R}_{>0}$ is a contraction parameter that is typically chosen close to zero, we add the contraction constraint

$$\sum_{i=1}^{N-1} d_x\big(t_i, \hat{x}(t_k), c_u(\cdot|t_k)\big) - \sum_{i=1}^{N-1} d_x\big(t_i, \hat{x}(t_{k-1}), c_u(\cdot|t_{k-1})\big) < -\alpha \tag{4.11}$$

to the optimal MPC problem solved during $[t_k, t_{k+1})$. Thus, the constraint in (4.11) ensures that $\sum_{i=1}^{N-1} d_x\big(t_i, \hat{x}(t_k), c_u(\cdot|t_k)\big)$ is a strictly decreasing function with respect to consecutive time steps k, implying the convergence of the state trajectory to the origin. To disregard the contraction constraint in (4.11) for the initial time, we define $\sum_{i=1}^{N-1} d_x\big(t_i, \hat{x}(t_{-1}), c^\star_u(\cdot|t_{-1})\big) = \infty$.

As stated in Subsection 4.4.1, we use the remainder of the previous solution $c^\star_u(\cdot|t_{k-1})$ as a safe backup if the optimal MPC problem is infeasible. To be consistent with the update of $c^\star_u(\cdot|t_k)$ in (4.9), we set

$$d_x\big(t_i, \hat{x}(t_k), c^\star_u(\cdot|t_k)\big) = \begin{cases} d_x\big(t_1 + t_i, \hat{x}(t_{k-1}), c^\star_u(\cdot|t_{k-1})\big) & \text{for } i \in \mathbb{N}_{[1,N-2]} \\ 0 & \text{for } i = N-1 \end{cases} \tag{4.12}$$

if the optimal MPC problem solved during $[t_k, t_{k+1})$ is infeasible. Instead of reusing the previous contraction distances, we could also compute $d_x\big(t_i, \hat{x}(t_k), c^\star_u(\cdot|t_k)\big)$ for $i \in \mathbb{N}_{[1,N-2]}$ based on (4.9) and (4.10). Nevertheless, we choose (4.12) to be consistent with the update of $c^\star_u(\cdot|t_k)$ and to make the contraction constraint less restrictive for the optimal MPC problem that is solved during the following time interval.

4.4.3 Algorithm

By incorporating the computation time considerations, the terminal constraint, and the contraction constraint, the online optimal MPC problem that is solved during the time interval $[t_k, t_{k+1})$

is

$$
\begin{aligned}
\underset{c_u(\cdot|t_k)}{\text{minimize}} \quad & J_{\text{MPC}}\big(\hat{x}(t_k), c_u(\cdot|t_k)\big) && \text{(4.13a)}\\
\text{subject to} \quad & \text{(4.8) and (4.11) are satisfied} && \text{(4.13b)}\\
& \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left(t_N, \{\hat{x}(t_k)\}, \langle c_u(\cdot|t_k), \mathbf{0}\rangle_Z, \mathcal{E}(\cdot)\right) \subseteq \mathcal{Z}_x^\star(t_{k+N}) && \text{(4.13c)}\\
& \Pi_x \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left([t_i, t_{i+1}), \{\hat{x}(t_k)\}, \langle c_u(\cdot|t_k), \mathbf{0}\rangle_Z, \mathcal{E}(\cdot)\right) \subseteq \mathcal{X} \quad \text{for } i \in \mathbb{N}_{[0,N-1]} && \text{(4.13d)}\\
& \Pi_u \widetilde{\mathcal{R}}^{\mathbf{M}}_{K\hat{x}}\left([t_i, t_{i+1}), \{\hat{x}(t_k)\}, \langle c_u(\cdot|t_k), \mathbf{0}\rangle_Z, \mathcal{E}(\cdot)\right) \subseteq \mathcal{U} \quad \text{for } i \in \mathbb{N}_{[0,N-1]}, && \text{(4.13e)}
\end{aligned}
$$

where J_{MPC} is a convex cost function. When choosing J_{MPC} to be a quadratic function, the COP in (4.13) is a simple quadratic programming problem [55]. Finally, we present our robust output feedback dual-mode MPC approach in Alg. 4.1, where we iteratively solve the optimal MPC problem in (4.13) in a moving horizon fashion until the safety-preserving controller takes over. Subsequently, we describe this algorithm in more detail.

Algorithm 4.1 Robust output feedback dual-mode MPC

```
1: $k \leftarrow 1$
2: while $d_x\big(t_1, \hat{x}(t_{k-1}), c_u^\star(\cdot|t_{k-1})\big) \nleq \alpha$ do        ▷ 1. mode
3:     $u(t) \leftarrow K\hat{x}(t_k) + c_u^\star(t_1|t_{k-1})$ for $t \in [t_k, t_{k+1})$
4:     $c_u^\star(\cdot|t_k) \leftarrow$ solve (4.13) for $\hat{x}(t_k)$
5:     if $c_u^\star(t_0|t_k) \not\equiv \emptyset$ then
6:         $d_x\big(\cdot, \hat{x}(t_k), c_u^\star(\cdot|t_k)\big) \leftarrow$ apply $\hat{x}(t_k), c_u^\star(\cdot|t_k)$ to (4.10)
7:     else
8:         $c_u^\star(\cdot|t_k) \leftarrow$ apply $c_u^\star(\cdot|t_{k-1})$ to (4.9)
9:         $d_x\big(\cdot, \hat{x}(t_k), c_u^\star(\cdot|t_k)\big) \leftarrow$ apply $d_x\big(\cdot, \hat{x}(t_{k-1}), c_u^\star(\cdot|t_{k-1})\big)$ to (4.12)
10:    end if
11:    $k \leftarrow k + 1$
12: end while
13: $k^{\star,\circ} \leftarrow k$
14: $\lambda_{\hat{x},k} \leftarrow$ solve (2.1) for $\mathcal{Z}_x^\star(t_{k^{\star,\circ}}), \hat{x}(t_{k^{\star,\circ}})$
15: while true do        ▷ 2. mode
16:     $u(t) \leftarrow K\hat{x}(t_k) + c_u^{\star,\circ}(t_k - t_{k^{\star,\circ}}|t_{k^{\star,\circ}}) + G_u^{\star,\circ}(t_k - t_{k^{\star,\circ}}|t_{k^{\star,\circ}})\lambda_{\hat{x},k}$ for $t \in [t_k, t_{k+1})$
17:     $k \leftarrow k + 1$
18: end while
```

In line 2 of Alg. 4.1, we implicitly check if the unknown state $x(t_k) \in \mathbb{R}^{n_x}$ is guaranteed to lie within the optimal large safe set $\mathcal{Z}_x^\star(t_k) \subseteq \mathcal{X}$ at t_k. If this is the case, we switch to the second mode and apply the inputs of the safety-preserving controller with corresponding $\mathcal{Z}_u^{\star,\circ}(\cdot|t_k) = \big\langle c_u^{\star,\circ}(\cdot|t_k), G_u^{\star,\circ}(\cdot|t_k)\big\rangle_Z$ to the system in lines 15 to 18. Otherwise, the input that is applied to the system during $[t_k, t_{k+1})$ is updated in line 3 based on Subsection 4.4.1. In addition, the optimal MPC problem in (4.13) is solved until t_{k+1} in line 4. In case of an infeasible solution, $c_u^\star(\cdot|t_k)$ and $d_x\big(\cdot, \hat{x}(t_k), c_u^\star(\cdot|t_k)\big)$ are updated based on the safe backup solution of the previous time step in lines 8 and 9. In the following theorem, we show that Alg. 4.1 achieves the control goal of this chapter formulated in Section 4.2.

Theorem 4.3 (Properties of Alg. 4.1): If an initial feasible solution $c_u^\star(\cdot|t_0)$ with corresponding $d_x\big(\cdot, \hat{x}(t_0), c_u^\star(\cdot|t_0)\big)$ is given at t_1, Alg. 4.1 steers the disturbed system in (2.20) to a neighborhood of the origin while satisfying the safety constraints in (4.2). ■

Proof. We must show two things: (i) The safety constraints in (4.2) are satisfied, and (ii) the system reaches a neighborhood of the origin in finite time.

(i) This part of the proof is based on extending the optimized correction input sequence by the safety-preserving control sequence and using the previous solution as a safe backup in case of the infeasibility of the optimal MPC problem in (4.13). Because these ideas follow standard proof techniques in robust dual-mode MPC [95], this part of the proof is omitted.

(ii) Based on Subsection 4.3.3, the safety-preserving controller steers the system to a neighborhood of the origin if $x(t_k) \in \mathcal{Z}_x^\star(t_k)$ with $k \in \mathbb{N}$. If

$$d_x\big(t_1, \hat{x}(t_{k-1}), c_u^\star(\cdot|t_{k-1})\big) \leq \alpha, \tag{4.14}$$

i.e., if the condition in Alg. 4.1 is satisfied for switching to the safety-preserving controller, $x(t_k)$ is guaranteed to be the origin or lie within $\mathcal{Z}_x^\star(t_k)$ based on (4.10), (4.12), and (4.13c). Thus, it remains to show that (4.14) is satisfied for some finite k.

On the one hand, if the optimal MPC problem in (4.13) solved during $[t_k, t_{k+1})$ is feasible, we know that the contraction constraint in (4.11) is satisfied, i.e., the contraction rate of at least $\alpha \in \mathbb{R}_{>0}$ is guaranteed for $\sum_{i=1}^{N-1} d_x\big(t_i, \hat{x}(t_k), c_u(\cdot|t_k)\big)$. On the other hand, if it is infeasible, $d_x\big(\cdot, \hat{x}(t_k), c_u^\star(\cdot|t_k)\big)$ is updated according to (4.12) in line 9 of Alg. 4.1, i.e., it is set equal to the remainder of the previous contraction distance sequence $d_x\big(\cdot, \hat{x}(t_{k-1}), c_u^\star(\cdot|t_{k-1})\big)$. By construction of (4.12), the contraction constraint in (4.11) is simplified to $-d_x\big(t_1, \hat{x}(t_{k-1}), c_u^\star(\cdot|t_{k-1})\big) < -\alpha$. This inequality holds because the condition in line 2 has been previously verified. Thus, the contraction rate of α is also guaranteed in case of the infeasibility of (4.13). Therefore, (4.14) is satisfied for some finite k. □

4.4.4 Simplifications

Solving the optimal MPC problem in (4.13) for large-scale systems is often computationally too expensive for real-time applications that require the sampling period Δt to be in the order of 100 ms. Thus, we propose some simplifications that reduce the online computational effort while maintaining the formal safety guarantees of Theorem 4.3.

Terminal Constraint

Ideally, we want to use the terminal constraint in (4.13c), as visualized in Fig. 4.3. Although the zonotope containment condition in (2.15) provides a way to solve (4.13c) for large-scale systems, it is still computationally too expensive for most real-time applications. Nevertheless, we know that the reachable set at the end of the prediction horizon can be separated by

$$\begin{aligned}\Pi_x \widetilde{\mathcal{R}}_{K\hat{x}}^{\mathbf{M}}\left(t_N, \{\hat{x}(t_k)\}, \langle c_u(\cdot|t_k), \mathbf{0}\rangle_Z, \mathcal{E}(\cdot)\right) = \Pi_x \widetilde{\mathcal{R}}_{K\hat{x}}^{(A,B,\{\mathbf{0}\},\mathrm{CT})}\left(t_N, \{\hat{x}(t_k)\}, \langle c_u(\cdot|t_k), \mathbf{0}\rangle_Z, \{\mathbf{0}\}\right) \\ \oplus\, \Pi_x \widetilde{\mathcal{R}}_{K\hat{x}}^{\mathbf{M}}\left(t_N, \{\mathbf{0}\}, \{\mathbf{0}\}, \mathcal{E}(\cdot)\right)\end{aligned} \tag{4.15}$$

based on Proposition 4.2. Thus, to speed up the online computations, we also want to separate the terminal constraint in (4.13c) based on (4.15). To achieve this goal, we decompose the zonotope containment condition in (2.15) into two COPs, as shown in the following proposition.

Proposition 4.4 (Separation of Zonotope Containment Condition in (2.15)): Let $\mathcal{Z}_1 = \langle c_1, G_1 \rangle_Z \subset \mathbb{R}^n$, $\mathcal{Z}_2 = \langle c_2, G_2 \rangle_Z \subset \mathbb{R}^n$, and $c_3 \in \mathbb{R}^n$ be given. In addition, let $\Gamma^\star \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z}_2) \times \mathtt{gen}(\mathcal{Z}_1)}$ be the solution of the COP

$$\begin{aligned} \underset{\Gamma}{\text{minimize}} \quad & J_\Gamma(\Gamma) && \text{(4.16a)} \\ \text{subject to} \quad & G_1 = G_2 \Gamma && \text{(4.16b)} \\ & |\Gamma| \, \mathbf{1} \leq \mathbf{1}, && \text{(4.16c)} \end{aligned}$$

where J_Γ is a convex cost function. If a vector $\gamma \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z}_2)}$ exists such that

$$\begin{aligned} c_2 - (c_1 + c_3) &= G_2 \gamma && \text{(4.17a)} \\ \left| \begin{bmatrix} \Gamma^\star & \gamma \end{bmatrix} \right| \mathbf{1} &\leq \mathbf{1}, && \text{(4.17b)} \end{aligned}$$

the Minkowski addition of $\mathcal{Z}_1$ and $\{c_3\}$ is contained in $\mathcal{Z}_2$, i.e., $\mathcal{Z}_1 \oplus \{c_3\} \subseteq \mathcal{Z}_2$. ■

Proof. Based on (2.10a) and (2.15), $\mathcal{Z}_1 \oplus \{c_3\} \subseteq \mathcal{Z}_2$ if a matrix $\Gamma \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z}_2) \times \mathtt{gen}(\mathcal{Z}_1)}$ and a vector $\gamma \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z}_2)}$ exist such that

$$\begin{aligned} G_1 &= G_2 \Gamma && \text{(4.18a)} \\ c_2 - (c_1 + c_3) &= G_2 \gamma && \text{(4.18b)} \\ \left| \begin{bmatrix} \Gamma & \gamma \end{bmatrix} \right| \mathbf{1} &\leq \mathbf{1}. && \text{(4.18c)} \end{aligned}$$

Because $\Gamma^\star$ satisfies the constraint in (4.16b), which is identical to (4.18a), feasibility of (4.17) implies satisfaction of the zonotope containment condition in (4.18). □

Based on (4.15) and Proposition 4.4, we separate the terminal constraint in (4.13c) into two COPs: one COP corresponds to the uncontrollable part of the reachable set in (4.15) and is solved offline, while the other COP corresponds to the controllable part and replaces (4.13c) online. In particular, let $\Gamma_{k+N}^\star$ of appropriate dimensions be the solution of the COP

$$\begin{aligned} \underset{\Gamma_{k+N}}{\text{minimize}} \quad & \|\Gamma_{k+N}\|_\infty \\ \text{subject to} \quad & G_{\mathcal{R}_{k+N,1}} = G_{\mathcal{Z}_x^\star(t_{k+N})} \Gamma_{k+N} \\ & |\Gamma_{k+N}| \, \mathbf{1} \leq \mathbf{1}, \end{aligned}$$

where $\left\langle c_{\mathcal{Z}_x^\star(t_{k+N})}, G_{\mathcal{Z}_x^\star(t_{k+N})} \right\rangle_Z \subseteq \mathcal{X}$ is the large safe set at t_{k+N}, and $\left\langle c_{\mathcal{R}_{k+N,1}}, G_{\mathcal{R}_{k+N,1}} \right\rangle_Z = \Pi_x \widetilde{\mathcal{R}}_{K\hat{x}}^{\mathbf{M}} (t_N, \{\mathbf{0}\}, \{\mathbf{0}\}, \mathcal{E}(\cdot))$ is the uncontrollable part of the reachable set in (4.15). Then, we replace the original terminal constraint in (4.13c) by

$$\begin{aligned} c_{\mathcal{Z}_x^\star(t_{k+N})} - (c_{\mathcal{R}_{k+N,1}} + c_{\mathcal{R}_{k+N,3}}) &= G_{\mathcal{Z}_x^\star(t_{k+N})} \gamma_{k+N} && \text{(4.19a)} \\ \left| \begin{bmatrix} \Gamma_{k+N}^\star & \gamma_{k+N} \end{bmatrix} \right| \mathbf{1} &\leq \mathbf{1}, && \text{(4.19b)} \end{aligned}$$

where $\gamma_{k+N} \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z}_x^\star(t_{k+N}))}$ is an optimization vector and

$$\left\langle c_{\mathcal{R}_{k+N,3}}, \mathbf{0} \right\rangle_Z = \Pi_x \widetilde{\mathcal{R}}_{K\hat{x}}^{(A,B,\{\mathbf{0}\},\mathrm{CT})} \left(t_N, \{\hat{x}(t_k)\}, \langle c_u(\cdot|t_k), \mathbf{0} \rangle_Z, \{\mathbf{0}\} \right)$$

is the controllable part of the reachable set in (4.15). Based on Proposition 4.4, it follows that the original terminal constraint in (4.13c) is satisfied if (4.19) is feasible.

State and Input Constraints

Based on Proposition 4.2, we also want to tighten the state and input constraints in (4.13d) and (4.13e) offline to further reduce the online computational effort [192, 194]. The constraint tightening can be achieved by computing Minkowski differences, which are defined in (2.8d). Although the Minkowski differences of the polytopes $\mathcal{X}$ or $\mathcal{U}$ and a zonotope $\mathcal{Z}$ of appropriate dimension can be computed exactly [217, Thm. 1], its representing number of half-spaces grows exponentially with $\mathtt{gen}(\mathcal{Z})$. Alternatively, to maintain the scalability of our approach, we directly separate the zonotope-in-polytope containment condition in (2.14), as shown in the following proposition.

Proposition 4.5 (Separation of Zonotope Containment Condition in (2.14)): The Minkowski addition of $\mathcal{Z}_1 = \langle c_1, G_1 \rangle_Z \subset \mathbb{R}^n$ and $\mathcal{Z}_3 = \langle c_3, G_3 \rangle_Z \subset \mathbb{R}^n$ is contained in $\mathcal{Z}_2 = \langle H_2, h_2 \rangle_P \subset \mathbb{R}^n$, i.e., $\mathcal{Z}_1 \oplus \mathcal{Z}_3 \subseteq \mathcal{Z}_2$, if and only if

$$H_2 c_1 + |H_2 G_1| \, \mathbf{1} \le h_2 - (H_2 c_3 + |H_2 G_3| \, \mathbf{1}) \, . \qquad \blacksquare$$

Proof. Based on (2.10a) and (2.14), $\mathcal{Z}_1 \oplus \mathcal{Z}_3 \subseteq \mathcal{Z}_2$ if and only if

$$\begin{aligned} & H_2(c_1 + c_3) + \left| H_2 \begin{bmatrix} G_1 & G_3 \end{bmatrix} \right| \mathbf{1} \le h_2 \\ \Leftrightarrow\ & H_2 c_1 + H_2 c_3 + |H_2 G_1| \, \mathbf{1} + |H_2 G_3| \, \mathbf{1} \le h_2 \\ \Leftrightarrow\ & H_2 c_1 + |H_2 G_1| \, \mathbf{1} \le h_2 - (H_2 c_3 + |H_2 G_3| \, \mathbf{1}) \, . \qquad \square \end{aligned}$$

Based on Proposition 4.5, tightening $\mathcal{Z}_2 = \langle H_2, h_2 \rangle_P \subset \mathbb{R}^n$ by $\mathcal{Z}_3 = \langle c_3, G_3 \rangle_Z \subset \mathbb{R}^n$ is achieved by $\langle H_2, h_2 - (H_2 c_3 + |H_2 G_3| \, \mathbf{1}) \rangle_P$, which is an under-approximation of $\mathcal{Z}_2 \ominus \mathcal{Z}_3$. Advantageously, this under-approximation does not introduce more conservativeness because the zonotope-in-polytope containment condition is necessary and sufficient. Thus, we can accurately and efficiently tighten the state and input constraints in (4.13d) and (4.13e) offline by considering the uncontrollable part of the reachable sets based on Propositions 4.2 and 4.5.

Error Sets and Safe Sets

As mentioned in Subsection 4.3.1, we do not require the initial error set $\mathcal{E}(t_0) \subset \mathbb{R}^{n_x}$ to be an RPI set of the autonomous, DT system in (4.3a), which would imply $\mathcal{E}(t_{i+1}) \subseteq \mathcal{E}(t_i)$ for any $i \in \mathbb{N}$ [200]. To avoid computing an infinite number of error sets, we exploit the fact that the error set sequence $\mathcal{E}(\cdot)$ usually converges quickly to the minimal robust positively invariant (mRPI) set $\mathcal{E}(t_\infty) \subset \mathbb{R}^{n_x}$. Computing a tight RPI over-approximation $\mathcal{E}_\infty \supseteq \mathcal{E}(t_\infty)$ can be achieved by slightly modifying our invariant set approaches in Chapter 3 or by following standard methods [38, 120–122]. Based on $\mathcal{E}_\infty$, let $\beta^\star \in \mathbb{R}_{\geq 0}$ be the solution of the linear programming problem

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & \beta && (4.20a) \\ \text{subject to} \quad & 0 \leq \beta && (4.20b) \\ & \mathcal{E}\left(t_{k_\infty}\right) \subseteq (1+\beta)\mathcal{E}_\infty, && (4.20c) \end{aligned}$$

where a finite $k_\infty \in \mathbb{N}$ is given. Then, we can use $(1+\beta^\star)\mathcal{E}_\infty$ as RPI over-approximation for all $\mathcal{E}\left(t_{k_\infty+i}\right)$ with $i \in \mathbb{N}$. Therefore, only a finite number of error sets must be computed.

Similarly, it suffices to compute only a finite number of optimal large safe sets. In particular, we replace $\mathcal{Z}_x^\star\left(t_{k_\infty+i}\right) \subseteq \mathcal{X}$ with $\mathcal{Z}_x^\star\left(t_{k_\infty}\right) \subseteq \mathcal{X}$ for all $i \in \mathbb{N}$. It is even sufficient to compute only $\mathcal{Z}_x^\star(t_N) \subseteq \mathcal{X}$ when choosing $k_\infty = N$ and staying in the first mode for the first N time steps in Alg. 4.1. In the following section, we use the simplifications proposed in this subsection to enable the real-time capability of our robust output feedback MPC approach.

4.5 Numerical Examples

In this section, we demonstrate the effectiveness of our proposed robust MPC approach using two numerical examples. In Subsection 4.5.1, we slightly modify the vehicle platooning system considered in Subsection 3.6.3, where the state was assumed to be measurable. In addition, we consider an under-actuated quadrotor model in Subsection 4.5.2.

For both numerical examples, we make the following choices: The sampling period is $\Delta t = 150\,\text{ms}$, the prediction horizon is $N = 20$, the applied correction input during $[t_0, t_1)$ is $\mathbf{0}$, the contraction parameter used in (4.11) is $\alpha = 10^{-3}$, and the stabilizing output injection matrix L is obtained by [218, Eq. 16]. In addition, to enable short computation times, we use the quadratic cost function $J_{\text{MPC}}\big(\hat{x}(t_k), c_u(\cdot|t_k)\big) = \sum_{i=1}^{N-1} L_{\text{MPC}}\big(\bar{x}(t_i|t_k), c_u(t_i|t_k)\big) + V_{\text{MPC}}\big(\bar{x}(t_N|t_k)\big)$ in (4.13a), where $L_{\text{MPC}}(\bar{x}, c_u) = \bar{x}^T\bar{x} + 10c_u^T c_u$ is the stage cost, $V_{\text{MPC}}(\bar{x}) = \bar{x}^T\bar{x}$ is the terminal cost, and $\bar{x}(t_i|t_k)$ equals the undisturbed reachable set $\Pi_x \widetilde{\mathcal{R}}_{K\hat{x}}^{(A,B,\{\mathbf{0}\},\text{CT})}\left(t_i, \{\hat{x}(t_k)\}, c_u(\cdot|t_k), \{\mathbf{0}\}\right)$. Thus, the optimal MPC problem in (4.13) is a simple quadratic programming problem [55]. Moreover, we use $(1+\beta^\star)\mathcal{E}_\infty$ as RPI over-approximation for all $\mathcal{E}\left(t_{N+i}\right)$ with $i \in \mathbb{N}$, where $\mathcal{E}_\infty \subset \mathbb{R}^{n_x}$ is computed by Alg. 2.1 with convergence tolerance $\epsilon = 0.01$ and $\beta^\star$ is the solution of (4.20) for $k_\infty = N$.

4.5.1 Vehicle Platooning System

In Subsection 3.6.3, we assumed that the state of the vehicle platoon is measurable. To demonstrate the effectiveness of our output feedback approach, we extend the vehicle platoon dynamics in (3.24) and (3.25) by the output equation in (4.1). In particular, the corresponding

Table 4.1: State, input, disturbance, and initial state estimation error bounds of vehicle platooning system.

Variables	Bounds
$e^{(1)}, e^{(2)}, e^{(3)}$	$[-10,\ 10]\,\mathrm{m}$
$\dot{e}^{(1)}, \dot{e}^{(2)}, \dot{e}^{(3)}$	$[-5,\ 5]\,\frac{\mathrm{m}}{\mathrm{s}}$
$a^{(1)}, a^{(2)}, a^{(3)}$	$[-8,\ 8]\,\frac{\mathrm{m}}{\mathrm{s}^2}$
$u^{(1)}, u^{(2)}, u^{(3)}$	$[-8,\ 8]\,\frac{\mathrm{m}}{\mathrm{s}^2}$
$a^{(0)}$	$[-1,\ 1]\,\frac{\mathrm{m}}{\mathrm{s}^2}$
$v^{(1)}, v^{(3)}, v^{(5)}$	$[-0.05,\ 0.05]\,\mathrm{m}$
$v^{(2)}, v^{(4)}, v^{(6)}$	$[-0.05,\ 0.05]\,\frac{\mathrm{m}}{\mathrm{s}}$
$\xi^{(1)}(t_0), \xi^{(4)}(t_0), \xi^{(7)}(t_0)$	$[-0.5,\ 0.5]\,\mathrm{m}$
$\xi^{(2)}(t_0), \xi^{(5)}(t_0), \xi^{(8)}(t_0)$	$[-0.5,\ 0.5]\,\frac{\mathrm{m}}{\mathrm{s}}$
$\xi^{(3)}(t_0), \xi^{(6)}(t_0), \xi^{(9)}(t_0)$	$[-0.5,\ 0.5]\,\frac{\mathrm{m}}{\mathrm{s}^2}$

output matrix $C_D \in \mathbb{R}^{6\times 9}$ is chosen as

$$C_D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

i.e., the third, sixth, and ninth elements of the state vector cannot be measured directly, which correspond to the effective accelerations of all three following vehicles. Moreover, the state, input, state disturbance, output disturbance, and initial state estimation error bounds are presented in Table 4.1. The stabilizing feedback matrix $K \in \mathbb{R}^{3\times 9}$ is the same as in (3.26). Moreover, the unknown initial state of the system is given by $x(t_0) = \begin{bmatrix} -9\,\mathrm{m} & 4\,\frac{\mathrm{m}}{\mathrm{s}} & 7\,\frac{\mathrm{m}}{\mathrm{s}^2} & 9\,\mathrm{m} & -4\,\frac{\mathrm{m}}{\mathrm{s}} & 7\,\frac{\mathrm{m}}{\mathrm{s}^2} & 3\,\mathrm{m} & 3\,\frac{\mathrm{m}}{\mathrm{s}} & 0 \end{bmatrix}^T$.

As described in Subsection 4.3.3, we slightly modify our large safe set approaches in Section 3.5 to incorporate output feedback control. Then, we execute Alg. 3.1 with convergence tolerance $\epsilon = 0.01$ and Alg. 3.3 with maximum interval radius $\epsilon = 0.01$ to obtain the scaled safe set $\mathcal{S}^{\star}_{K\hat{x}}(t_N) \subseteq \mathcal{X}$. In addition, we use $\mathcal{S}^{\star}_{K\hat{x}}(t_N)$ as safe terminal set $\mathcal{S}_{\text{safe}}$ when solving the COP in (3.22). In (3.22a), we use the geometric mean of the generator scaling vector $s_{x,0} \in \mathbb{R}^{18}_{\geq 0}$ for the cost function $J_{\mathcal{Z}_x(t_0)}$. Moreover, the fixed generator matrix $G_{\text{fixed}} \in \mathbb{R}^{9\times 18}$ is chosen as the generator matrix of $\texttt{reduce}\,(\mathcal{S}^{\star}_{K\hat{x}}(t_N), 2)$ and the initial time step is $k_{x,0} = 35$. Then, solving this COP to obtain the optimal large safe set $\mathcal{Z}^{\star}_x(t_N) \subseteq \mathcal{X}$ takes 1 min.

In Fig. 4.4, we show projections of sets and trajectories when sampling the state and output disturbances randomly from the admissible bounds reported in Table 4.1. As can be observed, the large safe set $\mathcal{Z}^{\star}_x(t_N)$ covers the state constraint set $\mathcal{X}$ quite well, and the reachable sets

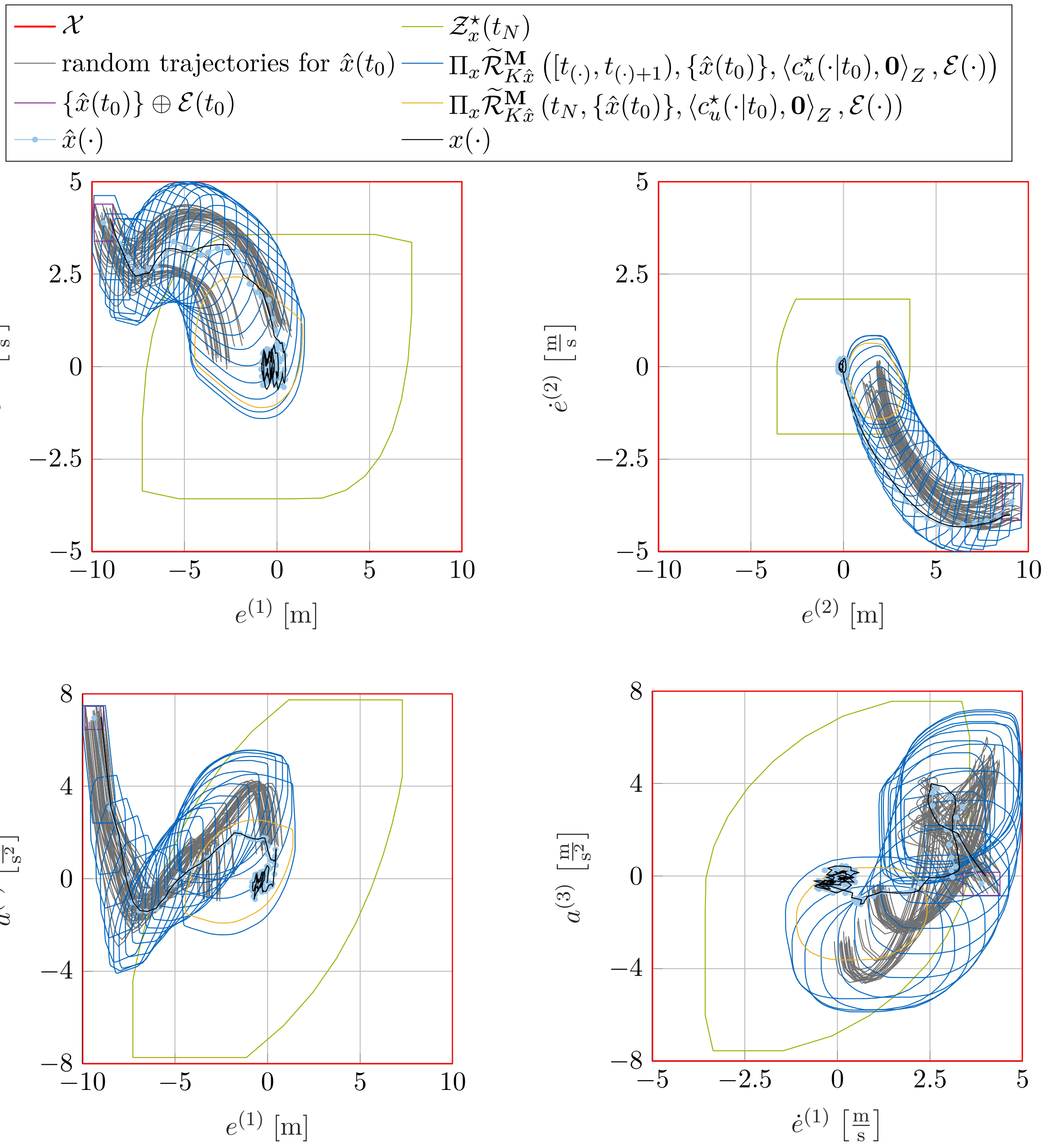


Figure 4.4: Two-dimensional projections of sets and trajectories. In particular, 50 random trajectories for the initial solution $c_u^\star(\cdot|t_0)$ along with the corresponding reachable sets are visualized. In addition, the unknown state trajectory $x(\cdot)$ and the corresponding state estimate trajectory $\hat{x}(\cdot)$ are shown for the first 20 s.

corresponding to the initial solution touch the bounds of $\mathcal{X}$. This observation indicates that our approach is not overly conservative. Because the maximum computation time for solving the optimal MPC problem in (4.13) is 111 ms, which is less than Δt, we never must abort the online optimization prematurely.

In Fig. 4.5, we visualize the corresponding state estimate, unknown state, and input trajectories for the first 20 s. As can be observed, the system is successfully steered to a neighborhood of the origin while satisfying the safety constraints in (4.2). In addition, the norm of $c_u(\cdot)$ diminishes within 1 s, resulting in a quick decrease of the cost L_{MPC}. Moreover, the switch to the second mode of our dual-mode MPC controller is barely visible, which results in a comfortable driving experience for all vehicles.

4.5.2 Quadrotor System

To demonstrate the computational efficiency of our approach, we also consider the twelve-dimensional, under-actuated quadrotor system proposed in [142, 219]. The system dynamics is linearized around the hover condition to obtain a linear model. The resulting twelve states are given by

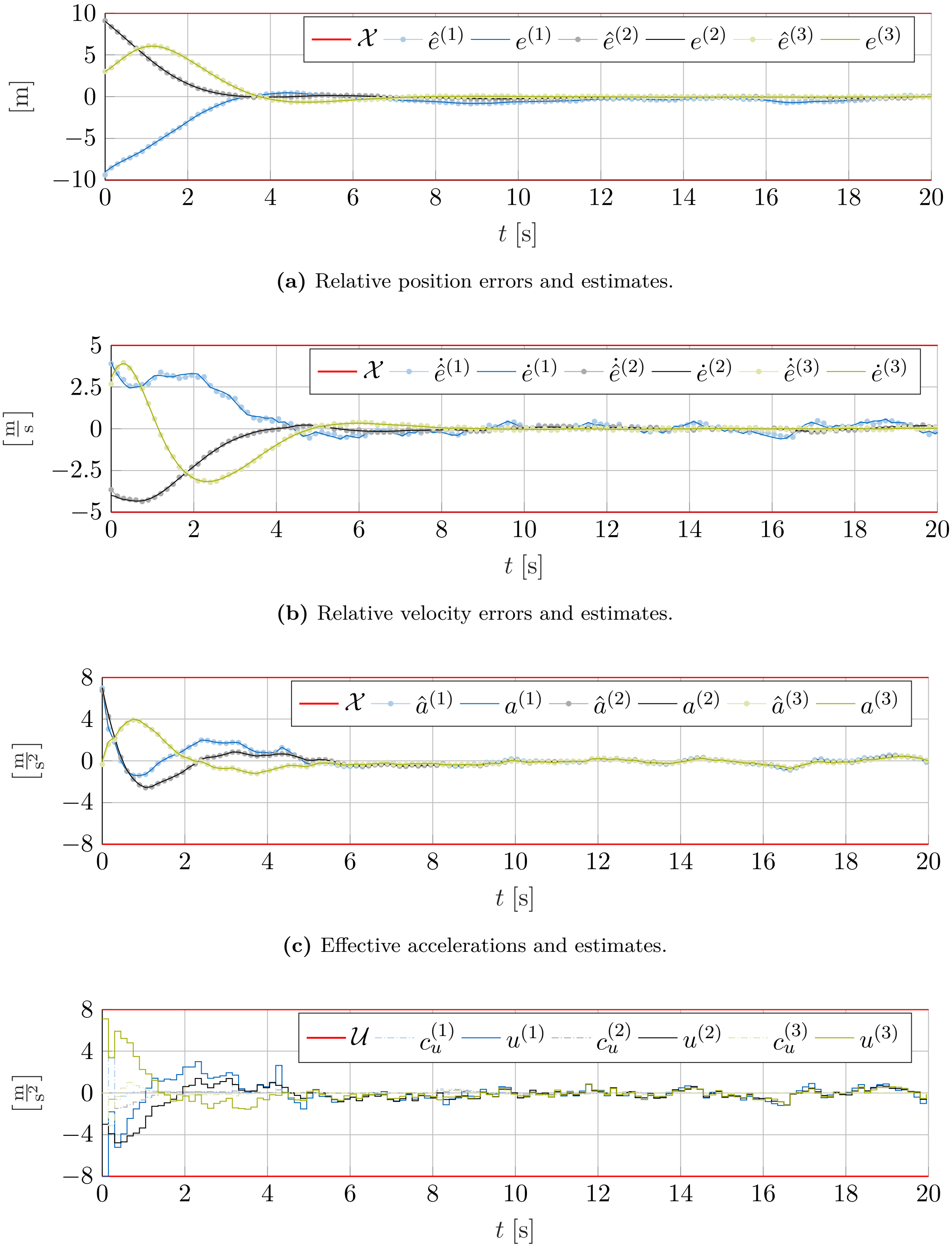
- the spatial positions $\begin{bmatrix} x^{(1)} & x^{(2)} & x^{(3)} \end{bmatrix}^T \in [-3, 3]^3$;
- the spatial velocities $\begin{bmatrix} x^{(4)} & x^{(5)} & x^{(6)} \end{bmatrix}^T \in [-3, 3]^3$;
- the angular positions $\begin{bmatrix} x^{(7)} & x^{(8)} \end{bmatrix}^T \in [-\pi/4, \pi/4]^2$, $x^{(9)} \in [-\pi, \pi]$; and
- the angular velocities $\begin{bmatrix} x^{(10)} & x^{(11)} & x^{(12)} \end{bmatrix}^T \in [-3, 3]^3$.

In addition, the four control inputs are given by

- the total normalized thrust $u^{(1)} \in [-9.81, 2.38]$; and
- the second-order derivatives of the angular positions $\begin{bmatrix} u^{(2)} & u^{(3)} & u^{(4)} \end{bmatrix}^T \in [-0.5, 0.5]^3$.

The uncertain wind is modeled by the unknown, bounded disturbance $\begin{bmatrix} w^{(4)} & w^{(5)} & w^{(6)} \end{bmatrix}^T \in [-0.01, 0.01]^3$ that affects only the three spatial velocities. In addition, the output matrix $C_D \in \mathbb{R}^{12\times 12}$ in (4.1) is chosen as identity matrix I, the output disturbance set is $\mathcal{V}_D = [-0.003, 0.003]^{12}$, and the initial error set is $\mathcal{E}(t_0) = [-0.03, 0.03]^{12}$. The unknown initial state of the system is given by $x(t_0) = \begin{bmatrix} 2 & -2 & 2 & -1 & 1 & -2 & 0 & 0 & 1 & 0.2 & 0.2 & -1 \end{bmatrix}^T$. Moreover, we compute the stabilizing feedback matrix $K \in \mathbb{R}^{4\times 12}$ using LQR-based controller synthesis [155], where the state and input weighting matrices are $Q = I$ and $R = 100I$.

Analogous to Subsection 4.5.1, we slightly modify our large safe set approaches in Section 3.5 to incorporate output feedback control. Then, we execute Alg. 3.1 with convergence tolerance $\epsilon = 0.1$ and Alg. 3.3 with maximum interval radius $\epsilon = 0.1$ to obtain the scaled safe set $\mathcal{S}^{\star}_{K\hat{x}}(t_N) \subseteq \mathcal{X}$, which we use as large safe set $\mathcal{Z}^{\star}_{x}(t_N)$ in our algorithm. Executing both algorithms takes 0.5 s.



(a) Relative position errors and estimates.

(b) Relative velocity errors and estimates.

(c) Effective accelerations and estimates.

(d) Control inputs.

Figure 4.5: State estimate, unknown state, and input trajectories. At sampling time step $k = 54$, i.e., at $t = 8.1\,\text{s}$, our robust output feedback dual-mode MPC controller switches to the second mode, which ensures robust constraint satisfaction at all future times.

Because the maximum computation time for solving the optimal MPC problem in (4.13) is 74 ms, which is less than Δt, we never must abort the online optimization prematurely. As the projections of sets and trajectories are qualitatively the same compared to Figs. 4.4 and 4.5, we omit these plots.

4.6 Summary

In this chapter, we have proposed a scalable robust output feedback dual-mode MPC algorithm for sampled-data LTI systems. In the first mode of this algorithm, we iteratively solve an optimal MPC problem that uses a large safe set as a terminal set. We switch to the second mode when the state of the system is guaranteed to lie within this safe set. Then, the corresponding safety-preserving controller formally guarantees robust constraint satisfaction at all future times.

Because the presented large safe set methods in Section 3.5 assume the state of the sampled-data system to be measurable, we have slightly modified these computations to incorporate output feedback control. Based on these modified safe sets, we have computed real-time controllers that steer the system to a neighborhood of the origin and explicitly consider the computation time of the online optimal MPC problem. Finally, we have demonstrated the effectiveness of our proposed robust MPC approach on a slight modification of the vehicle platooning system considered in Subsection 3.6.3, where the state was assumed to be measurable. Because our approach uses zonotopes as set representation and is based on scalable reachability analysis and convex optimization, a sampling period of 150 ms is achieved for this nine-dimensional, sampled-data system, which is impossible for existing approaches.

In the following chapter, we present our supervisory safety filter approach that iteratively solves a COP on a moving horizon to minimally modify a desired control input. Thus, the core of this safety filter is a robust MPC problem whose finite prediction horizon is typically only one. To formulate any robust MPC problem, the system model and its corresponding disturbance set are usually assumed to be given. Because we remove this assumption in the following chapter, we perform offline set membership identification to identify models that are conformant to a finite set of training data. As a new measurement obtained online might invalidate the model conformance, we quickly update this conformant model online to restore formal safety guarantees.

5 Safety Filter

In this chapter, which is based on [49, 51], we propose a minimally invasive supervisory safety filter approach that makes no assumptions about the availability of a model along with its corresponding disturbance set. After introducing the concept of safety filters and reviewing the relevant literature in Section 5.1, we formulate the control goal of this chapter in Section 5.2. In Section 5.3, we perform offline set membership identification to identify models that are conformant to a finite set of training data. Based on these conformant models, we present our safety filter algorithm and our online conformance updates in Section 5.4, followed by a demonstration of its effectiveness using four numerical examples in Section 5.5. Finally, we summarize this chapter in Section 5.6.

5.1 Introduction and State of the Art

Excellent control performance is typically achieved using sophisticated control methods with fine-tuned parameters. Due to the high complexity of these high-performance controllers, which are obtained, e.g., using machine learning techniques, it is usually cumbersome to formally verify safety. Nevertheless, providing formal safety guarantees for any controller can be accomplished using an additional supervisory safety filter along with a corresponding safety-preserving or safe backup controller. Such a filter aims at modifying the desired input of the unverified high-performance controller in a minimally invasive or least restrictive way so that safety is guaranteed at all times. Therefore, safety filters serve as supervisory mediators between a simple, safe backup controller and a sophisticated, unverified high-performance controller, as illustrated in Fig. 5.1.

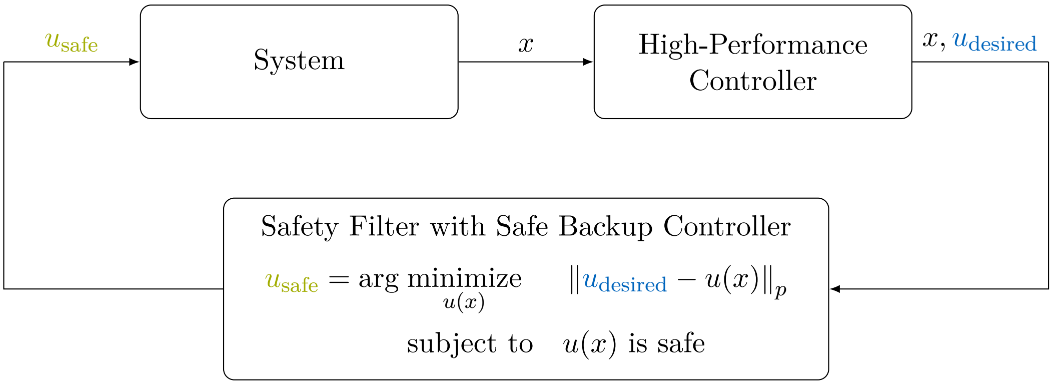

u_{safe}
System
x
High-Performance Controller
x, u_{desired}
Safety Filter with Safe Backup Controller
$u_{\text{safe}} = \underset{u(x)}{\text{arg minimize}} \quad \|u_{\text{desired}} - u(x)\|_p$
subject to $u(x)$ is safe

Figure 5.1: Safety filter concept. The safety filter can be seen as a supervisory mediator between performance and safety.

Because the simple concept of safety filters is compelling, they are used in a wide variety of areas, such as safe reinforcement learning [220, 221], human-in-the-loop control [129], medical systems [222], motion planning [17, 18, 46], collision avoidance [223], and fault tolerant systems [224]. Moreover, different naming conventions have been introduced in the literature because safety filters are widely used in several disciplines. For instance, they are closely related to safety shields [225], verified control envelopes [226], and sandboxing control [227].

Safety filters can be efficiently implemented using, e.g., reachability analysis [45, 228], invariance control [229, 230], barrier functions [231, 232], or command governors [44]. These implementations usually use model predictive control (MPC) techniques [41–43, 181], where an optimal control problem is iteratively solved online on a moving horizon of length one. By increasing this time horizon, the corresponding safe set of the safety filter can be enlarged [130], i.e., the region of operation of the safe backup controller can be enlarged. Because the size of its safe set mainly determines the conservativeness of a safety filter, we refer to Section 3.1 for a brief literature overview of large safe sets.

Because formal safety guarantees are model-based, they are only valid as long as the identified model of the unknown system is valid [233–235]. However, perfect models are typically unavailable. Thus, control approaches using a finite training data set have gained interest over the past few years. For instance, a conformant model and a robust control invariant (RCI) approximation of the minimal RCI set are simultaneously computed in [148]. However, there is no guarantee that an unseen measurement obtained online also lies within this RCI set because only a finite training data set was used for its construction. Thus, additional assumptions are required to provide formal safety guarantees for an infinite time horizon. For instance, the disturbance set is assumed to be known while the system dynamics is unknown [236, 237], which is quite unrealistic. The availability of such *a priori* disturbance sets is also a standard assumption in most robust MPC approaches [41–43, 160], including our robust output feedback MPC method in Chapter 4. Alternatively, to obtain a margin that also ensures safety online, the tightest estimate of the disturbance set can be multiplied by a safety factor greater than one [238, 239]. However, it is unclear how to choose this safety factor without introducing excessive conservativeness to ensure safety for an infinite time horizon.

It is clear from the literature review that minimally invasive supervisory safety filters are of great importance for many robust control applications. However, it is an unresolved issue to provide a scalable safety filter approach that makes no assumptions about the availability of a model along with its corresponding disturbance set. In this chapter, we address this issue by incorporating our efficient large safe set computations proposed in Section 3.5 into safety filters. In contrast to existing methods, our approach is scalable while making no assumptions about the availability of the disturbance set. In particular, we perform offline set membership identification to identify a conformant model based on a finite set of training data. Then, we use this identified model to compute a large safe set with a corresponding safe backup controller. Moreover, we quickly update the conformant model, the safe set, and the safe backup controller online because a new measurement might invalidate the identified model due to the unknown disturbance set. Thanks to the scalability of our safety filter algorithm, these updates are performed in real time, even for medium-sized problems. As in Chapter 4, we also consider all online computation times for solving optimization problems to guarantee safety despite such computational delays. Before we present our scalable safety filter approach, we formulate the control problem of this chapter in the following section.

5.2 Problem Formulation

Up to now, we have always assumed that a system and its corresponding disturbance set are given. In this chapter, we consider an unknown, discrete-time (DT), time-invariant system that evolves according to

$$x(t_{k+1}) = f\left(x(t_k), u(t_k), w_D(t_k)\right), \tag{5.1}$$

i.e., it is compactly represented by the model $\mathbf{M} = \left(f\left(x, u, w_D\right), \mathcal{W}_D, \mathrm{DT}\right)$ with $\mathcal{W}_D \subset \mathbb{R}^{n_w}$ being the unknown, DT disturbance set. Similar to the continuous-time (CT) state and input constraints in (3.1) and (4.2), the unknown, DT system in (5.1) is constrained by

$$x(\cdot) \in \mathcal{X} \tag{5.2a}$$

$$u(\cdot) \in \mathcal{U}, \tag{5.2b}$$

where $\mathcal{X} = \langle H_{\mathcal{X}}, h_{\mathcal{X}} \rangle_P \subset \mathbb{R}^{n_x}$ and $\mathcal{U} = \langle H_{\mathcal{U}}, h_{\mathcal{U}} \rangle_P \subset \mathbb{R}^{n_u}$ are the known state and input constraint sets, respectively, which contain the origin.

To gain some knowledge about the unknown system in (5.1), sufficiently exciting training data $\{x(t_k), u(t_k), x(t_{k+1})\}_{k=1}^{N}$ is available offline [236, 240], where $N \in \mathbb{N}_{>0}$ denotes the number of measurements. In particular, we assume that the state of the system is directly measurable in contrast to the problem in Chapter 4. To deal with unstable systems, the training data is not required to be recorded from a single run of the system but can be obtained by performing multiple short experiments. Then, the control goal of this chapter is the construction of a scalable safety filter, i.e., the goal is to determine the minimal modification $\left\|\tilde{u}(t_k) - u(t_k)\right\|_p$ of a desired input $\tilde{u}(t_k) \in \mathbb{R}^{n_u}$ with $k \in \mathbb{N}$ such that the safety constraints in (5.2) are satisfied.

5.3 Model Conformance

In this section, we construct linear models that are conformant to or consistent with the available offline training data. Throughout this section, we assume that the offline training data has captured all possible system behaviors. Because the system in (5.1) is unknown, we remove this unrealistic assumption in Section 5.4 by updating our conformant model online.

To enable formal safety guarantees, we first identify linear models that are conformant to the offline training data [235].

Definition 5.1 (Conformant Model): Let $\{x(t_k), u(t_k), x(t_{k+1})\}_{k=1}^{N}$ be a finite set of training data. Then, $\mathbf{M} = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \mathrm{DT}\right)$ is a conformant model if for all $k \in \mathbb{N}_{[1,N]}$

$$x(t_{k+1}) = \widehat{A}_D x(t_k) + \widehat{B}_D u(t_k) + \hat{w}_D(t_k) \tag{5.3a}$$

$$\hat{w}_D(t_k) \in \widehat{\mathcal{W}}_D, \tag{5.3b}$$

where $\widehat{A}_D \in \mathbb{R}^{n_x \times n_x}$, $\widehat{B}_D \in \mathbb{R}^{n_x \times n_u}$, and $\widehat{\mathcal{W}}_D \subset \mathbb{R}^{n_x}$ are the estimated system matrix, input matrix, and disturbance set, respectively. ■

To decrease the conservativeness of our safety filter, we want to find the conformant model $\mathbf{M} = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \mathrm{DT}\right)$ whose estimated disturbance set $\widehat{\mathcal{W}}_D$ has the smallest volume.

For simplicity, this is typically achieved in two steps [238, 241]: First, a standard system identification is performed to obtain $\widehat{A}_D$ and $\widehat{B}_D$ [234]. Second, an optimization problem is solved to minimize the volume of $\widehat{\mathcal{W}}_D$. Instead of this two-step approach, we propose to address both aspects simultaneously by solving

$$\underset{\mathbf{M}=\left(\widehat{A}_D,\widehat{B}_D,\widehat{\mathcal{W}}_D,\mathrm{DT}\right)}{\text{minimize}} \quad \text{volume of } \widehat{\mathcal{W}}_D \tag{5.4a}$$

$$\text{subject to} \quad \mathbf{M} \text{ is a conformant model.} \tag{5.4b}$$

Hence, we use state-space representations in contrast to existing set membership identification methods [239, 242], which exploit autoregressive exogenous structures.

The volume of a general zonotope $\mathcal{Z} = \langle c, G\rangle_Z \subset \mathbb{R}^{n_x}$ can be computed exactly [162] or estimated using sampling-based techniques [243]. However, both approaches are computationally too expensive for large-scale systems, so we must use a suitable heuristic to cast (5.4) as a convex optimization problem (COP). For instance, suitable choices for the cost in (5.4a) are the 1-norm of G, or the Frobenius norm of G [93]. Nevertheless, we can exactly solve (5.4) when restricting $\widehat{\mathcal{W}}_D = \left\langle c_{\widehat{\mathcal{W}}_D}, G_{\widehat{\mathcal{W}}_D}\right\rangle_Z$ to be a parallelotope with symmetric positive definite generator matrix $G_{\widehat{\mathcal{W}}_D}$, as shown in the following proposition.

Proposition 5.2 (Conformant Model with Parallelotopic Disturbance Set): Let $\{x(t_k), u(t_k), x(t_{k+1})\}_{k=1}^N$ be a finite set of training data. If $A_D^\star, B_D^\star, c^\star, G^\star$ is the solution of the COP

$$\underset{A_D,B_D,c,G}{\text{minimize}} \quad -\log(\det(G)) \tag{5.5a}$$

$$\text{subject to} \quad G = G^T \succ \mathbf{0} \tag{5.5b}$$

$$\left|Gx(t_{k+1}) - A_D x(t_k) - B_D u(t_k) - c\right| \leq \mathbf{1} \quad \text{for } k \in \mathbb{N}_{[1,N]}, \tag{5.5c}$$

$\mathbf{M}^\star = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \mathrm{DT}\right)$ is the solution of (5.4), where $\widehat{A}_D = (G^\star)^{-1}A_D^\star$, $\widehat{B}_D = (G^\star)^{-1}B_D^\star$, and $\widehat{\mathcal{W}}_D = \left\langle (G^\star)^{-1}c^\star, (G^\star)^{-1}\right\rangle_Z$ is restricted to be a parallelotope with symmetric positive definite generator matrix. ■

Proof. Model conformance constraint: Based on (2.2), the unique parameter vector $\lambda_k \in \mathbb{R}^{\mathtt{gen}(\widehat{\mathcal{W}}_D)}$ for any $\hat{w}_D(t_k) \in \widehat{\mathcal{W}}_D = \left\langle c_{\widehat{\mathcal{W}}_D}, G_{\widehat{\mathcal{W}}_D}\right\rangle_Z$ is given by $\lambda_k = G_{\widehat{\mathcal{W}}_D}^{-1}\left(\hat{w}_D(t_k) - c_{\widehat{\mathcal{W}}_D}\right)$ with $|\lambda_k| \leq \mathbf{1}$. By additionally using (5.3a), we obtain the model conformance constraint

$$\left|G_{\widehat{\mathcal{W}}_D}^{-1} x(t_{k+1}) - G_{\widehat{\mathcal{W}}_D}^{-1}\widehat{A}_D x(t_k) - G_{\widehat{\mathcal{W}}_D}^{-1}\widehat{B}_D u(t_k) - G_{\widehat{\mathcal{W}}_D}^{-1} c_{\widehat{\mathcal{W}}_D}\right| \leq \mathbf{1},$$

which is equivalent to (5.5c) when choosing $c_{\widehat{\mathcal{W}}_D} = G^{-1}c$ and $G_{\widehat{\mathcal{W}}_D} = G^{-1}$.

Cost function: Because $G_{\widehat{\mathcal{W}}_D} \in \mathbb{R}^{n_x \times n_x}$ is symmetric positive definite, the volume of $\widehat{\mathcal{W}}_D$ is proportional to $\det\left(G_{\widehat{\mathcal{W}}_D}\right)$ [162]. In addition, $\log(\det(M))$ equals $-\log\left(\det\left(M^{-1}\right)\right)$ for any symmetric positive definite matrix $M \in \mathbb{R}^{n \times n}$, the inverse of a symmetric positive definite matrix is also symmetric positive definite, and the determinant is logarithmically concave on the set of symmetric positive definite matrices [55, 163]. Thus, the convex cost function in (5.5a) selects the conformant model whose estimated disturbance set has the smallest volume. □

A matrix must be inverted when using Proposition 5.2 to obtain the optimal conformant model $\mathbf{M}^\star$. Numerical problems when computing the inverse of a matrix can be avoided by further restricting $\widehat{\mathcal{W}}_D$ to be a multidimensional interval, i.e., by restricting the corresponding generator matrix to be a diagonal matrix. In this case, (5.4) is a simple linear programming problem [59], as shown in the following proposition.

Proposition 5.3 (Conformant Model with Multidimensional Disturbance Interval): Let $\{x(t_k), u(t_k), x(t_{k+1})\}_{k=1}^N$ be a finite set of training data. If $A_D^\star, B_D^\star, c^\star, g^\star$ is the solution of the linear programming problem

$$\underset{A_D,B_D,c,g}{\text{minimize}} \quad \mathbf{1}^T g \tag{5.6a}$$

$$\text{subject to} \quad \left|x(t_{k+1}) - A_D x(t_k) - B_D u(t_k) - c\right| \leq g \quad \text{for } k \in \mathbb{N}_{[1,N]}, \tag{5.6b}$$

$\mathbf{M}^\star = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \mathrm{DT}\right)$ is the solution of (5.4), where $\widehat{A}_D = A_D^\star$, $\widehat{B}_D = B_D^\star$, and $\widehat{\mathcal{W}}_D = \langle c^\star, \mathtt{diag}(g^\star)\rangle_Z$ is restricted to be a multidimensional interval. ■

Proof. Model conformance constraint: For any $\hat{w}_D(t_k) \in \widehat{\mathcal{W}}_D = \left\langle c_{\widehat{\mathcal{W}}_D}, G_{\widehat{\mathcal{W}}_D}\right\rangle_Z$, there exists a $\lambda_k \in \mathbb{R}^{\mathtt{gen}(\widehat{\mathcal{W}}_D)}$ with $|\lambda_k| \leq \mathbf{1}$ such that $\hat{w}_D(t_k) - c_{\widehat{\mathcal{W}}_D} = G_{\widehat{\mathcal{W}}_D}\lambda_k$. These conditions can be equivalently reformulated as $\left|\hat{w}_D(t_k) - c_{\widehat{\mathcal{W}}_D}\right| \leq \mathtt{diag}\left(\left|G_{\widehat{\mathcal{W}}_D}\right|\right)$ because $G_{\widehat{\mathcal{W}}_D} \in \mathbb{R}^{n_x \times n_x}$ is a diagonal matrix and zonotopes are centrally symmetric sets. By additionally using (5.3a), the model conformance constraint in (5.6b) is obtained.

Cost function: The volume of $\left\langle c_{\widehat{\mathcal{W}}_D}, G_{\widehat{\mathcal{W}}_D}\right\rangle_Z$ equals the product of the elements of $\mathtt{diag}\left(\left|G_{\widehat{\mathcal{W}}_D}\right|\right) \in \mathbb{R}^{n_x}_{\geq 0}$, which is a nonconvex function. Nevertheless, because the model conformance constraint in (5.6b) is linear, it can be equivalently separated into a single constraint for each of the n_x dimensions. Thus, there is no coupling between any of the n_x elements of $\mathtt{diag}\left(\left|G_{\widehat{\mathcal{W}}_D}\right|\right)$. Therefore, minimizing the sum of any n_x convex functions whose single arguments are the elements of $\mathtt{diag}\left(\left|G_{\widehat{\mathcal{W}}_D}\right|\right)$ also minimizes the product of these elements, resulting in the smallest volume of $\widehat{\mathcal{W}}_D$. We choose these n_x convex functions as identity maps to obtain a simple COP, resulting in the linear cost function in (5.6a). □

By restricting $\widehat{\mathcal{W}}_D$ to be a parallelotope with a symmetric positive definite generator matrix or a multidimensional interval, we can exactly and efficiently solve the optimization problem in (5.4). However, using such restricted set representations might be too conservative for some applications. To overcome this potential problem, we propose another set membership identification approach that allows $\widehat{\mathcal{W}}_D$ to be a general zonotope and approximates the volume minimization of $\widehat{\mathcal{W}}_D$ by finding the minimum scaling factor $s_{\mathcal{X}}^\star \in \mathbb{R}_{\geq 0}$ such that $\widehat{\mathcal{W}}_D \subseteq s_{\mathcal{X}}^\star \mathcal{X}$, similar to (3.16). To cast this problem as a COP, we use the generator scaling framework [143], i.e., we fix the arbitrary orientations of the generators of $\widehat{\mathcal{W}}_D$ and optimize only their scaling factors, as shown in the following proposition.

Proposition 5.4 (Conformant Model with Zonotopic Disturbance Set): Let $\{x(t_k), u(t_k), x(t_{k+1})\}_{k=1}^N$ be a finite set of training data. In addition, let $A_D^\star, B_D^\star, c^\star, s_{\mathcal{X}}^\star, \lambda_1^\star$,

$\lambda_2^\star, \ldots, \lambda_N^\star$ be the solution of the COP

$$
\begin{aligned}
\underset{A_D, B_D, c, s_{\mathcal{X}} \lambda_1, \lambda_2, \ldots, \lambda_N}{\text{minimize}} \quad & J_{\mathbf{M}}\left(s_{\mathcal{X}}, \lambda_1, \lambda_2, \ldots, \lambda_N\right) && \text{(5.7a)}\\
\text{subject to} \quad & \lambda_{\max} = \max\left(\left|\begin{bmatrix}\lambda_1 & \lambda_2 & \ldots & \lambda_N\end{bmatrix}\right|\right) && \text{(5.7b)}\\
& 0 \leq s_{\mathcal{X}} && \text{(5.7c)}\\
& \langle c, G_{\text{fixed}} \mathtt{diag}\left(\lambda_{\max}\right)\rangle_Z \subseteq s_{\mathcal{X}} \mathcal{X} && \text{(5.7d)}\\
& x(t_{k+1}) - A_D x(t_k) - B_D u(t_k) = c + G_{\text{fixed}} \lambda_k \quad \text{for } k \in \mathbb{N}_{[1,N]}, && \text{(5.7e)}
\end{aligned}
$$

where $J_{\mathbf{M}}$ is a convex cost function, the function $\max(M)$ in (5.7b) returns a vector containing the maximum value of each row of $M \in \mathbb{R}^{\mathtt{gen}(\widehat{\mathcal{W}}_D) \times N}$, and $G_{\text{fixed}} \in \mathbb{R}^{n_x \times \mathtt{gen}(\widehat{\mathcal{W}}_D)}$ is a fixed generator matrix. Then, $\mathbf{M}^\star = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \text{DT}\right)$ is a conformant model, where $\widehat{A}_D = A_D^\star$, $\widehat{B}_D = B_D^\star$, and $\widehat{\mathcal{W}}_D = \langle c^\star, G_{\text{fixed}} \mathtt{diag}\left(\lambda_{\max}^\star\right)\rangle_Z$ with $\lambda_{\max}^\star = \max\left(\left|\begin{bmatrix}\lambda_1^\star & \lambda_2^\star & \ldots & \lambda_N^\star\end{bmatrix}\right|\right)$. Moreover, $\widehat{\mathcal{W}}_D \subseteq s_{\mathcal{X}}^\star \mathcal{X}$. ■

Proof. For any $\lambda_k \in \mathbb{R}^{\mathtt{gen}(\widehat{\mathcal{W}}_D)}$, there exists a $\underline{\lambda}_k \in \mathbb{R}^{\mathtt{gen}(\widehat{\mathcal{W}}_D)}$ with $|\underline{\lambda}_k| \leq \mathbf{1}$ such that $G_{\text{fixed}} \lambda_k = G_{\text{fixed}} \mathtt{diag}(|\lambda_k|) \underline{\lambda}_k$. In addition, $\langle \bar{c}, G_{\text{fixed}} \mathtt{diag}(|\lambda_k|)\rangle_Z \subseteq \left\langle \bar{c}, G_{\text{fixed}} \mathtt{diag}\left(\left|\bar{\lambda}_k\right|\right)\right\rangle_Z$ for any $\bar{c} \in \mathbb{R}^{n_x}$ and $\bar{\lambda}_k \in \mathbb{R}^{\mathtt{gen}(\widehat{\mathcal{W}}_D)}$ with $|\lambda_k| \leq \left|\bar{\lambda}_k\right|$. By using these relations, the fact that $|\lambda_k| \leq \lambda_{\max}$ for $k \in \mathbb{N}_{[1,N]}$ because of (5.7b), the model conformance constraint in (5.7e), and the system dynamics in (5.3a), it follows that the optimal $\mathbf{M}^\star$ is a conformant model. In addition, the constraints in (5.7c) and (5.7d) enforce $\widehat{\mathcal{W}}_D \subseteq s_{\mathcal{X}}^\star \mathcal{X}$. □

It is straightforward to show that the optimal models obtained by solving (5.6) and (5.7) are identical when choosing $J_{\mathbf{M}} = \mathbf{1}^T \max\left(\left|\begin{bmatrix}\lambda_1 & \lambda_2 & \ldots & \lambda_N\end{bmatrix}\right|\right)$ and $G_{\text{fixed}} = I$ in (5.7). Thus, the COP in (5.7) offers more flexibility at the expense of an increased computational cost.

In summary, we can efficiently compute an optimal linear model $\mathbf{M}^\star = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \text{DT}\right)$ that is conformant to the offline training data and has an estimated zonotopic disturbance set of small volume. To define a meaningful control problem, we assume that the tuple $\left(\widehat{A}_D, \widehat{B}_D\right)$ is stabilizable. If this assumption is violated, we could increase the number of measurements N, add more actuators, and follow more sophisticated experiment design approaches [234, 244]. In the following section, we use the optimal conformant model $\mathbf{M}^\star$ as an essential building block in our supervisory safety filter.

5.4 Robust Safety Filter

We want to avoid having big spikes when switching between a desired control input and a safe backup control input [245]. Thus, our safety filter aims to minimize modifying a desired control input to ensure the satisfaction of the safety constraints in (5.2). We achieve this goal by enforcing the state of the unknown system in (5.1) to stay within a DT, large safe set, which is obtained by slightly modifying the approaches in Section 3.5. Because the desired control inputs are only known during runtime, we solve an optimal control problem online, which takes a nonnegligible amount of time [216]. Similar to Subsection 4.4.1, we explicitly consider such computational delays instead of assuming that optimization problems can be solved instantaneously. In addition, we present our safety filter algorithm. Finally, we propose our online conformance update to restore formal safety guarantees as soon as we detect that $\hat{w}_D(t_k) \notin \widehat{\mathcal{W}}_D$ with $k \in \mathbb{N}$ for the optimal conformant model $\mathbf{M}^\star = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \mathrm{DT}\right)$.

5.4.1 Safe Sets

To provide the high-performance controller in Fig. 5.1 with a large region of operation, we use the large safe set approach in Subsection 3.5.2, which is based on a CT model. To apply this state feedback control method to the DT model $\mathbf{M}^\star = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \mathrm{DT}\right)$, we need to consider the following main modifications:

- The relation between the CT system at sampling times and the equivalent DT system is given in (2.26) and (2.27).
- The set-based state feedback controller in (3.4) is defined only at sampling times but not between them.
- The constraints corresponding to the time interval $[t_k, t_{k+1})$ are enforced only at the sampling time t_k.

By performing these slight modifications and solving (3.20), we compute optimal large safe sets $\mathcal{Z}_x^\star(t_0) \subseteq \mathcal{X}$ along with corresponding $\mathcal{Z}_u^{\star,\circ}(\cdot)$ in (3.21). Then, based on Proposition 3.12, these safety-preserving or safe backup controllers guarantee the satisfaction of the constraints in (5.2) for $\mathbf{M}^\star$ if $x(t_0) \in \mathcal{Z}_x^\star(t_0)$.

5.4.2 Computation Time Considerations

To compute the safe backup control input in (3.4) for the initial set $\mathcal{Z}_x^\star(t_0) \subseteq \mathcal{X}$, we must find the not necessarily unique initial parameter vector $\lambda_{x,0} \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z}_x^\star(t_0))}$ satisfying (3.2). Obtaining $\lambda_{x,0}$ can be achieved by solving the COP in (3.3), which causes a nonnegligible computational delay that invalidates the formal safety guarantees [216]. Alternatively, if the extreme points of $\mathcal{Z}_x^\star(t_0)$ are given, closed-form expressions of $\lambda_{x,0}$ exist [246]. However, obtaining the extreme points of a general zonotope is a computationally complex task [75]. To ensure the scalability of our approach, we present an efficient method for computing $\lambda_{x,0}$ without assuming that an optimization problem can be solved instantaneously at time step 0, as shown in the following lemma.

Lemma 5.5 (Parallelotope-in-Zonotope Parameter Vector): Let a parallelotope $\mathcal{P} = \langle c_1, G_1 \rangle_Z \subset \mathbb{R}^n$ and a zonotope $\mathcal{Z} = \langle c_2, G_2 \rangle_Z \subset \mathbb{R}^n$ be given. In addition, let a matrix $\Gamma \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z}) \times n}$ and a vector $\gamma \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z})}$ exist such that (2.15) is satisfied, where the arbitrary scaling factor is $\mathbf{1}$. Then, a valid parameter vector $\lambda_{\mathcal{Z}} \in \mathbb{R}^{\mathtt{gen}(\mathcal{Z})}$ of $\mathcal{Z}$ with $|\lambda_{\mathcal{Z}}| \leq \mathbf{1}$ for parameterizing any $s \in \mathcal{P}$ is

$$\lambda_{\mathcal{Z}} = -\gamma + \Gamma G_1^{-1}(s - c_1), \tag{5.8}$$

i.e., s can be expressed by $s = c_2 + G_2 \lambda_{\mathcal{Z}}$. ■

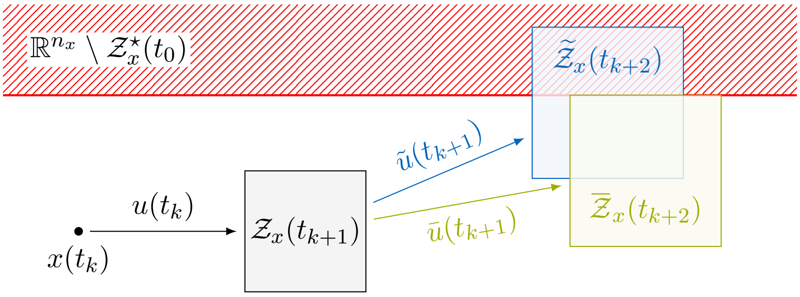

$\mathbb{R}^{n_x} \setminus \mathcal{Z}_x^\star(t_0)$
$\widetilde{\mathcal{Z}}_x(t_{k+2})$
$\tilde{u}(t_{k+1})$
$u(t_k)$
$\mathcal{Z}_x(t_{k+1})$
$\overline{\mathcal{Z}}_x(t_{k+2})$
$x(t_k)$
$\bar{u}(t_{k+1})$

Figure 5.2: Set-based safety filter. Because applying the desired input $\tilde{u}(t_{k+1})$ at time step $k+1$ might lead to leaving the optimal safe set $\mathcal{Z}_x^\star(t_0)$ at time step $k+2$, it is minimally modified to obtain the safe input $\bar{u}(t_{k+1})$.

Proof. Based on (2.2), any $s \in \mathcal{P}$ can be parameterized by the unique parameter vector $\lambda_\mathcal{P} = G_1^{-1}(s - c_1)$ of $\mathcal{P}$ with $|\lambda_\mathcal{P}| \leq \mathbf{1}$. If (2.15a) and (2.15b) are satisfied, it follows that

$$c_1 + G_1\lambda_\mathcal{P} = c_2 + G_2(-\gamma + \Gamma\lambda_\mathcal{P})$$

for any $\lambda_\mathcal{P} \in \mathbb{R}^n$. In addition, satisfaction of (2.15c) implies $|-\gamma + \Gamma\lambda_\mathcal{P}| \leq \mathbf{1}$ for all $\lambda_\mathcal{P}$ with $|\lambda_\mathcal{P}| \leq \mathbf{1}$. Thus, choosing

$$\begin{aligned} \lambda_\mathcal{Z} &= -\gamma + \Gamma\lambda_\mathcal{P} \\ &= -\gamma + \Gamma G_1^{-1}(s - c_1) \end{aligned}$$

results in a valid parameter vector of $\mathcal{Z}$ for any $s = c_1 + G_1\lambda_\mathcal{P}$. □

If we know that $x(t_0) \in \mathbb{R}^{n_x}$ lies within a small parallelotope that is contained in the zonotopic optimal safe set $\mathcal{Z}_x^\star(t_0)$, we can compute $\lambda_{x,0}$ at time step 0 based on Lemma 5.5 without solving an optimization problem. Before presenting our optimal control problem that explicitly considers all computation times for solving optimization problems online, subsequently, we give an overview of our set-based safety filter approach.

Because the desired control inputs are only known during runtime, we must solve an optimal control problem online. To explicitly consider such nonnegligible online computational delays, we verify the desired input not for the current but for the next time step, as illustrated in Fig. 5.2. In particular, at time step $k \in \mathbb{N}$, the state $x(t_k) \in \mathbb{R}^{n_x}$ is measured, and the input $u(t_k) \in \mathbb{R}^{n_u}$ that was previously verified as safe is applied until $k+1$. During this time, we want to verify safety when applying the desired input $\tilde{u}(t_{k+1}) \in \mathbb{R}^{n_u}$ at $k+1$. If safety might be violated, we minimally modify $\tilde{u}(t_{k+1})$ to obtain the safe input $\bar{u}(t_{k+1}) \in \mathcal{U}$ that ensures $x_{k+2} \in \mathcal{Z}_x^\star(t_0)$. By evaluating the system in (5.3a) in a set-based fashion, the reachable sets in Fig. 5.2 are

$$\begin{aligned} \mathcal{Z}_x(t_{k+1}) &= \left\{\widehat{A}_D x(t_k) + \widehat{B}_D u(t_k)\right\} \oplus \widehat{\mathcal{W}}_D \\ \widetilde{\mathcal{Z}}_x(t_{k+2}) &= \widehat{A}_D \mathcal{Z}_x(t_{k+1}) \oplus \left\{\widehat{B}_D \tilde{u}(t_{k+1})\right\} \oplus \widehat{\mathcal{W}}_D \\ \overline{\mathcal{Z}}_x(t_{k+2}) &= \widehat{A}_D \mathcal{Z}_x(t_{k+1}) \oplus \left\{\widehat{B}_D \bar{u}(t_{k+1})\right\} \oplus \widehat{\mathcal{W}}_D. \end{aligned}$$

We also want to mention that it might be infeasible to find a safe input $\bar{u}(t_{k+1})$ because the optimal safe set $\mathcal{Z}_x^\star(t_0) = \langle c_x^\star(t_0), G_x^\star(t_0)\rangle_Z$ is usually not an RCI set.

To verify or, if necessary, minimally modify the desired input $\tilde{u}(t_{k+1})$, we solve an optimal control problem that considers all online computation times starting at time step $k \in \mathbb{N}$. Let $\bar{u}^\star(t_{k+1}), \gamma^\star(t_{k+2}), \Gamma^\star(t_{k+2}), c_{\mathcal{P}}^\star(t_{k+2}), G_{\mathcal{P}}^\star(t_{k+2})$ be the solution of the COP

$$\underset{\substack{\bar{u}(t_{k+1}),\gamma(t_{k+2}),\Gamma(t_{k+2}),\\ c_{\mathcal{P}}(t_{k+2}),G_{\mathcal{P}}(t_{k+2})}}{\text{minimize}} \quad \left\|\tilde{u}(t_{k+1}) - \bar{u}(t_{k+1})\right\|_p \tag{5.9a}$$

$$\text{subject to} \quad \bar{u}(t_{k+1}) \in \mathcal{U} \tag{5.9b}$$

$$\overline{\mathcal{Z}}_x(t_{k+2}) = \left\{\widehat{A}_D^2 x(t_k) + \widehat{A}_D \widehat{B}_D u(t_k) + \widehat{B}_D \bar{u}(t_{k+1})\right\} \oplus \widehat{A}_D \widehat{\mathcal{W}}_D \oplus \widehat{\mathcal{W}}_D \tag{5.9c}$$

$$\langle c_{\mathcal{P}}(t_{k+2}), G_{\mathcal{P}}(t_{k+2})\rangle_Z = \texttt{reduce}\left(\overline{\mathcal{Z}}_x(t_{k+2}), 1\right) \tag{5.9d}$$

$$G_{\mathcal{P}}(t_{k+2}) = G_x^\star(t_0)\Gamma(t_{k+2}) \tag{5.9e}$$

$$c_x^\star(t_0) - c_{\mathcal{P}}(t_{k+2}) = G_x^\star(t_0)\gamma(t_{k+2}) \tag{5.9f}$$

$$\left|\begin{bmatrix}\Gamma(t_{k+2}) & \gamma(t_{k+2})\end{bmatrix}\right| \mathbf{1} \leq \mathbf{1}, \tag{5.9g}$$

where any p-norm with $p \geq 1$ can be chosen for the cost function in (5.9a). Then, the optimal safe input at time step $k+1$ is $\bar{u}^\star(t_{k+1}) \in \mathcal{U}$.

Instead of the standard constraint $\overline{\mathcal{Z}}_x(t_{k+2}) \subseteq \mathcal{Z}_x^\star(t_0)$, we use (5.9d) through (5.9g) based on Lemma 5.5 to ensure that the set-based safe backup control input in (3.4) with (3.21) can be computed without solving an optimization problem at time step $k+2$. In particular, based on (5.8), the required initial parameter vector is

$$\lambda_{x,0} = -\gamma^\star(t_{k+2}) + \Gamma^\star(t_{k+2})\big(G_{\mathcal{P}}^\star(t_{k+2})\big)^{-1}\big(x(t_{k+2}) - c_{\mathcal{P}}^\star(t_{k+2})\big). \tag{5.10}$$

In addition, $G_{\mathcal{P}}^\star(t_{k+2}) \in \mathbb{R}^{n_x \times n_x}$ only depends on the generator matrix of $\widehat{A}_D \widehat{\mathcal{W}}_D \oplus \widehat{\mathcal{W}}_D$ based on (5.9d). Therefore, $G_{\mathcal{P}}^\star(t_{k+2})$ and its inverse $\big(G_{\mathcal{P}}^\star(t_{k+2})\big)^{-1}$ are independent of the current time step k and, thus, are computed only once. Because the inverse of a diagonal matrix can be easily obtained, simple multidimensional interval over-approximations instead of general parallelotopic ones can also be used in (5.9d). As a result, we only need to perform a few simple matrix operations to compute the safe backup control input in (3.4) with (3.21) at time step $k+2$.

The COP in (5.9) is a robust MPC problem, where the length of the prediction horizon is only one, the terminal set is not necessarily RCI, and all online computation times are explicitly considered. Although we could easily incorporate a larger horizon that increases the region of operation and the computation time [130], we opt for the small horizon for simplicity. In addition, if solving (5.9) requires a longer time than the sampling period $\Delta t \in \mathbb{R}_{>0}$ to complete, we abort the optimization prematurely to maintain the validity of our formal safety guarantees, similar to Subsection 4.4.1.

In summary, the optimal control problem in (5.9) minimally modifies the desired input $\tilde{u}(t_{k+1})$ while ensuring that the safe backup control input in (3.4) can be computed at time step $k+2$ without solving an optimization problem. In the following subsection, we show how the COP in (5.9) is integrated into our safety filter algorithm.

5.4.3 Algorithm

We now present Alg. 5.1 that implements our supervisory safety filter. This algorithm proceeds in two steps at each sampling time: First, the safe input $u(t_k)$ applied to the unknown system in (5.1) at time step $k \in \mathbb{N}$ is computed. Second, the COP in (5.9) is solved to verify or, if necessary, minimally modify the desired input $\tilde{u}(t_{k+1})$. If (5.9) is infeasible, i.e., if $\bar{u}(t_{k+1})$ equals the empty set $\emptyset$, we use the safe backup control input at the subsequent time step $k+1$. In the following theorem, we show that Alg. 5.1 achieves the control goal of this chapter formulated in Section 5.2.

Algorithm 5.1 Robust safety filter

```
 1: $\bar{u}^\star(t_0) \leftarrow u(t_0)$
 2: for $k \leftarrow 0, 1, 2, \dots$ do
 3:     get $x(t_k)$ and $\tilde{u}(t_{k+1})$
 4:     if $\bar{u}^\star(t_k) \not\equiv \emptyset$ then                                  ▷ use solution of (5.9)
 5:         $u(t_k) \leftarrow \bar{u}^\star(t_k)$
 6:         $k_{x,0} \leftarrow 0$                                                ▷ reset initial time step
 7:     else                                                          ▷ use safe backup control input
 8:         if $k_{x,0} \equiv 0$ then
 9:             $\lambda_{x,0} \leftarrow -\gamma^\star(t_k) + \Gamma^\star(t_k)\big(G_{\mathcal{P}}^\star(t_k)\big)^{-1}\big(x(t_k) - c_{\mathcal{P}}^\star(t_k)\big)$        ▷ see (5.10)
10:             $k_{x,0} \leftarrow k$                                            ▷ update initial time step
11:         end if
12:         $u(t_k) \leftarrow Kx(t_k) + c_u^{\star,\circ}\left(t_{k-k_{x,0}}\right) + G_u^{\star,\circ}\left(t_{k-k_{x,0}}\right)\lambda_{x,0}$          ▷ see (3.4) and (3.21)
13:     end if
14:     apply $u(t_k)$ to the unknown system in (5.1)
15:     $\bar{u}^\star(t_{k+1}), \gamma^\star(t_{k+2}), \Gamma^\star(t_{k+2}), c_{\mathcal{P}}^\star(t_{k+2}), G_{\mathcal{P}}^\star(t_{k+2}) \leftarrow$ solve (5.9) for $x(t_k), u(t_k), \tilde{u}(t_{k+1})$
16: end for
```

Theorem 5.6 (Properties of Alg. 5.1): Let $\mathcal{S}_{Kx} \subseteq \mathcal{X}$ be a safe set, let $\mathcal{Z}_x^\star(t_0)$ be the optimal safe set obtained by solving (3.20), and let the corresponding optimal model $\mathbf{M}^\star = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \mathrm{DT}\right)$ be also conformant to all online obtained data. In addition, let $x(t_0) \in \mathcal{Z}_x^\star(t_0)$, $\left\{\widehat{A}_D x(t_0) + \widehat{B}_D u(t_0)\right\} \oplus \widehat{\mathcal{W}}_D \subseteq \mathcal{Z}_x^\star(t_0)$, $u(t_0) \in \mathcal{U}$, and the COP in (5.9) be feasible for $x(t_0)$, $u(t_0)$, $\tilde{u}(t_1)$. Then, the applied control inputs in Alg. 5.1 are minimal modifications of the desired inputs so that the safety constraints in (5.2) are satisfied for the unknown system in (5.1). ■

Proof. Because $\mathbf{M}^\star$ is also conformant to all online obtained data, the satisfaction of the safety constraints in (5.2) for the estimated system in (5.3a) implies constraint satisfaction for the unknown system in (5.1). Thus, it is sufficient to consider (5.3a).

We use our safe backup controller to guarantee safety for an infinite time horizon. Because the initial time was chosen to be zero during its construction in Subsections 3.5.2 and 5.4.1, we appropriately shift the counter $k \in \mathbb{N}$ of the correction input zonotope $\mathcal{Z}_u^{\star,\circ}(t_k)$ in line 12 of Alg. 5.1. Then, applying the resulting safe backup control inputs to the system ensures the satisfaction of the safety constraints in (5.2) based on Proposition 3.12.

If the COP in (5.9) is feasible, the control inputs in line 5 of Alg. 5.1 are minimal modifications of the desired inputs with respect to the cost function in (5.9a). In addition, the state and input constraints are satisfied for the next time step because of the incorporated reachability analysis in (5.9) and $\mathcal{Z}_x^\star(t_0) \subseteq \mathcal{X}$. Moreover, if the COP in (5.9) is infeasible, we use the safe backup controller until it is feasible again. □

In summary, Alg. 5.1 ensures the satisfaction of the safety constraints in (5.2) while considering all computation times for solving optimization problems. This statement is only valid if the optimal conformant model $\mathbf{M}^\star = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \mathrm{DT}\right)$ is valid at all times, which, however, is constructed offline in Section 5.3 based on a finite set of training data. Because the system in (5.1) is unknown, we have no guarantee that $\hat{w}_D(t_k) \in \widehat{\mathcal{W}}_D$ for all $k \in \mathbb{N}$. Thus, we perform conformance updates online to restore formal safety guarantees if a model invalidation is detected, as presented in the following subsection.

5.4.4 Online Conformance Updates

We update $\mathbf{M}^\star$, $\mathcal{Z}_x^\star(t_0)$, and $\mathcal{Z}_u^{\star,\circ}(\cdot)$ online as soon as $\hat{w}_D(t_k) \notin \widehat{\mathcal{W}}_D$ to restore formal safety guarantees, similar to [45]. Restoring a conformant model can be achieved by solving the COPs presented in Section 5.3 including not only the offline training data but also all online obtained data. Although the number of constraints scales only linearly with the amount of data, this approach quickly poses computational and memory problems as time proceeds. Therefore, to ensure scalability, an update is needed that is independent of the amount of online data, which implies independence of the elapsed time.

We address this issue by fixing $\widehat{A}_D$ and $\widehat{B}_D$ of our offline-constructed optimal conformant model $\mathbf{M}^\star = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \mathrm{DT}\right)$ and by minimally enlarging $\widehat{\mathcal{W}}_D$ to restore model conformance. To enable a fast update procedure, we restrict $\widehat{\mathcal{W}}_D$ to be a multidimensional interval $\left[\underline{\widehat{\mathcal{W}}_D}, \overline{\widehat{\mathcal{W}}_D}\right]$ with lower bound $\underline{\widehat{\mathcal{W}}_D}$ and upper bound $\overline{\widehat{\mathcal{W}}_D}$. If we detect that $\hat{w}_D^{(i)}(t_k) < \underline{\widehat{\mathcal{W}}_D^{(i)}}$ or $\overline{\widehat{\mathcal{W}}_D^{(i)}} < \hat{w}_D^{(i)}(t_k)$ for any $i \in \mathbb{R}^{n_x}$, we set $\underline{\widehat{\mathcal{W}}_D^{(i)}}$ or $\overline{\widehat{\mathcal{W}}_D^{(i)}}$ equal to $\hat{w}_D^{(i)}(t_k)$ to restore model conformance. After updating $\mathbf{M}^\star$, we also check if there still exists a small safe set $\mathcal{S}_{Kx} \subseteq \mathcal{X}$. In addition, we update $\mathcal{Z}_x^\star(t_0)$ by solving (3.20) and compute the resulting correction input zonotope sequence $\mathcal{Z}_u^{\star,\circ}(\cdot)$ in (3.21).

Because no conformant model is available during these online updates, the satisfaction of the safety constraints in (5.2) can no longer be formally guaranteed. Thus, quickly performing these updates and reducing the probability of constraint violation using the previous safe backup controller is the best we can do in this situation. Therefore, if $\hat{w}_D(t_k) \notin \widehat{\mathcal{W}}_D$, we set the control input $\bar{u}^\star(t_k)$ in Alg. 5.1 equal to $\emptyset$ as long as our online conformance update process is running.

5.5 Numerical Examples

In this section, we demonstrate the effectiveness of our safety filter approach using four numerical examples taken from the literature [130, 142, 143, 236]. To show the low conservativeness of our large safe sets, we also compute tight RCI under-approximations of the maximal robust control invariant (MRCI) set for the two low-dimensional examples by executing Alg. 2.2 with convergence tolerance $\epsilon = 10^{-5}$.

For all four numerical examples, we make the following choices: The sampling period is $\Delta t = 0.1\,\mathrm{s}$. In addition, we increment $k_{x,0} \in \mathbb{N}_{>0}$ based on Theorem 3.13 until the cost of the COP in (3.20) is unchanged for two consecutive iterations or 50 iterations are reached, which is done to ensure finite termination of the iterative procedure. This final $k_{x,0}$ is then used for all subsequent online conformance updates. The cost in (3.20a) is $J_{\mathcal{Z}_x(t_0)} = \mathbf{1}^T s_{x,0}$ so that (3.20) is a simple linear programming problem [59]. After solving (3.20) for the offline training data, we erase the i^{th} column of G_{fixed} if the i^{th} element of the optimal generator scaling vector $s^{\star}_{x,0}$ is smaller than 0.05. This erasure is done because these generators of G_{fixed} significantly increase the computation times when solving (3.20) online to perform conformance updates, although the shape of the optimal $\mathcal{Z}^{\star}_x(t_0)$ is typically only slightly changed as $\widehat{A}_D$ and $\widehat{B}_D$ are fixed. Moreover, we choose the 2-norm for the cost function in (5.9a), i.e., we choose the Euclidean norm.

As mentioned in Section 5.2, the training data is not required to be recorded from a single run of the system but can be obtained by performing multiple short experiments. This helpful property allows us to efficiently handle unstable systems. Because the chosen experiment design [234, 244] for training data generation is irrelevant to our approach, for simplicity, we generate the training data by sampling uniformly from $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}$.

5.5.1 Two-Dimensional System without Disturbances

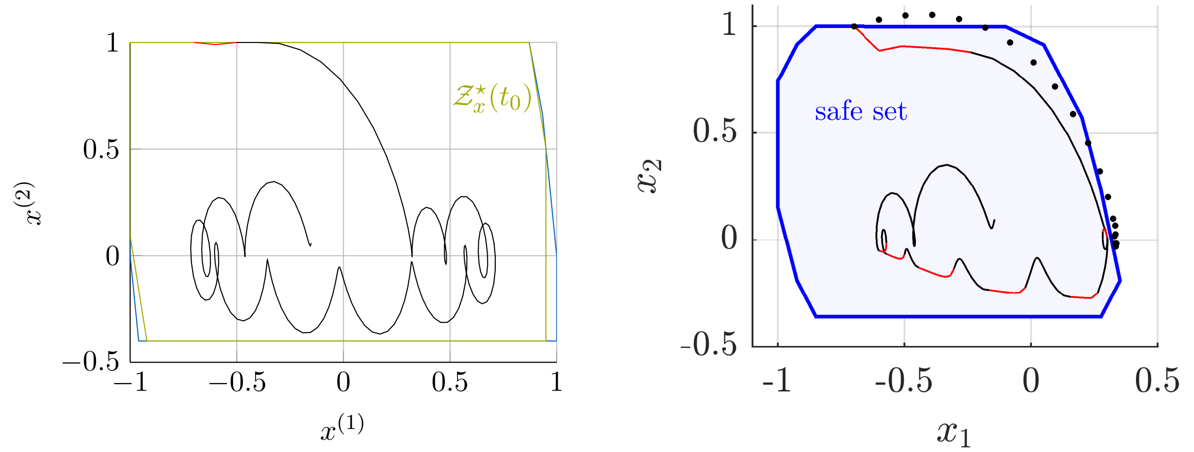
We consider the mass-spring-damper (MSD) example presented in [130], which is briefly summarized subsequently. The unknown system in (5.1) is described by

$$x(t_{k+1}) = \begin{bmatrix} 1.0 & 0.1 \\ -0.3 & 0.8 \end{bmatrix} x(t_k) + \begin{bmatrix} 0.0 \\ 0.1 \end{bmatrix} u(t_k) + w_D(t_k).$$

The disturbance set is $\mathcal{W}_D = \{\mathbf{0}\}$, and the multidimensional input and state constraint intervals are described by $\mathcal{U} = [-2.5, 2.5]$, $\underline{\mathcal{X}} = \begin{bmatrix} -1 & -0.4 \end{bmatrix}^T$, and $\overline{\mathcal{X}} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$, respectively. The stabilizing feedback matrix $K = \begin{bmatrix} -4.12 & -5.32 \end{bmatrix}$ is computed using linear-quadratic regulator (LQR)-based controller synthesis based on approximate system and input matrices that are assumed to be known. In addition, it is assumed that training data $\{x(t_i), u(t_i), x(t_{i+1})\}_{i=1}^{600}$ is generated by sampling uniformly from $\mathcal{X}$ and $\mathcal{U}$. The initial state is $x(t_0) = \begin{bmatrix} -0.7 & 1 \end{bmatrix}^T$ and the desired input is $\tilde{u}(t_k) = 2\sin(0.01\pi k) + 0.5\sin(0.12\pi k)$ for $k \in \mathbb{N}_{[0,200]}$.

By solving the linear programming problem in (5.6), we obtain the optimal conformant model $\mathbf{M}^{\star} = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \mathrm{DT}\right)$ based on the available training data. Because $\mathbf{M}^{\star}$ equals the unknown model $(A, B, \{\mathbf{0}\}, \mathrm{DT})$ up to floating-point precision, we never have to update $\mathbf{M}^{\star}$ online. To cover $\mathcal{X}$, we choose the columns of $G_{\text{fixed}} \in \mathbb{R}^{2\times 20}$ in (3.20) to be 20 uniformly distributed points around the top half unit circle.

In Fig. 5.3a, we present the simulation results when choosing $u(t_0) = -0.2$ for the initial input. As observed, our safety filter minimally modifies the desired input only in the first two time steps. Thus, our method intervenes significantly less than the safety filter approach



(a) Our approach. A tight RCI under-approximation of the MRCI set is visualized in blue, which shows the low conservativeness of our large safe set $\mathcal{Z}_x^\star(t_0)$.

(b) This figure is taken from [130]. The dotted black curve can be ignored.

Figure 5.3: Comparison of different safety filter approaches for the two-dimensional system. Red (black) color corresponds to states for which the desired input is (is not) minimally modified to guarantee safety at all times. Thus, our approach intervenes significantly less compared to [130].

in [130], whose performance is shown in Fig. 5.3b. To illustrate the low conservativeness of our optimal safe set $\mathcal{Z}_x^\star(t_0)$ in Fig. 5.3a, we also visualize a tight RCI under-approximation of the MRCI set.

To compare the set membership identification methods presented in Section 5.3, we subsequently assume that the unknown disturbance set $\mathcal{W}_D$ is not the origin but given by $[-0.1, 0.1]^2$, i.e., the volume of $\mathcal{W}_D$ is 0.04. In Fig. 5.4, we show the volumes of the estimated disturbance sets $\widehat{\mathcal{W}}_{D,(5.5)}$, $\widehat{\mathcal{W}}_{D,(5.6)}$, and $\widehat{\mathcal{W}}_{D,\mathrm{LLS}}$ that are obtained by solving (5.5), (5.6), and a linear least-squares system identification problem with subsequent parallelotopic volume minimization, respectively. As can be observed in Fig. 5.4, the volume of $\widehat{\mathcal{W}}_{D,(5.5)}$ is always smaller than the volume of $\widehat{\mathcal{W}}_{D,(5.6)}$. In addition, both volumes are monotonically increasing and converging to the volume of the unknown disturbance set $\mathcal{W}_D$ from below. In contrast to this monotonic increase, the volume of $\widehat{\mathcal{W}}_{D,\mathrm{LLS}}$ fluctuates and even exceeds $\mathcal{W}_D$. This observation shows the advantage when performing system identification and volume minimization in one step.

5.5.2 Unstable Three-Dimensional System

To demonstrate the usefulness of our online conformance updates proposed in Subsection 5.4.4, we consider the unstable system presented in [236], which is briefly summarized subsequently. The unknown system in (5.1) is described by

$$x(t_{k+1}) = \begin{bmatrix} -0.5 & 1.4 & 0.4 \\ -0.9 & 0.3 & -1.5 \\ 1.1 & 1.0 & -0.4 \end{bmatrix} x(t_k) + \begin{bmatrix} 0.1 & -0.3 \\ -0.1 & -0.7 \\ 0.7 & -1.0 \end{bmatrix} u(t_k) + w_D(t_k),$$

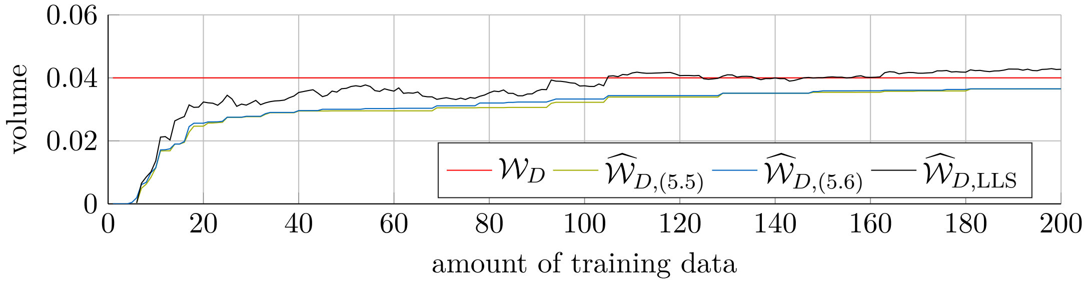


Figure 5.4: Comparison of different system identification approaches for the two-dimensional system. In addition to the unknown disturbance set $\mathcal{W}_D$, we visualize $\widehat{\mathcal{W}}_{D,\mathrm{LLS}}$, obtained by performing linear least-squares system identification with subsequent parallelotopic volume minimization. Moreover, $\widehat{\mathcal{W}}_{D,(5.5)}$ and $\widehat{\mathcal{W}}_{D,(5.6)}$ are obtained by solving (5.5) and (5.6), respectively.

and the state feedback matrix is

$$K = \begin{bmatrix} -2.45 & -1.29 & -2.40 \\ -0.61 & -0.03 & -2.18 \end{bmatrix}. \tag{5.11}$$

We assume that the unknown disturbance set is $\mathcal{W}_D = [-0.04, 0.04]^3$, and the known state and input constraint sets are $\mathcal{X} = [-1,1]^3$ and $\mathcal{U} = [-1,1]^2$, respectively. The initial state $x(t_0) \in \mathbb{R}^3$ and the initial input $u(t_0) \in \mathbb{R}^2$ are the origin. The desired input $\tilde{u}(t_k) \in \mathbb{R}^2$ and the disturbance $w_D(t_k) \in \mathbb{R}^3$ are uniformly sampled online from $\mathcal{U}$ and $\mathcal{W}_D$ for all $k \in \mathbb{N}_{[0,10^5]}$.

We generate training data $\{x(t_i), u(t_i), x(t_{i+1})\}_{i=1}^{100}$ by sampling uniformly from $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}_D$. By solving the linear programming problem in (5.6), we obtain the optimal conformant linear model $\mathbf{M}^\star = \left(\widehat{A}_D, \widehat{B}_D, \widehat{\mathcal{W}}_D, \mathrm{DT}\right)$ with

$$\widehat{A}_D = \begin{bmatrix} -0.5001 & 1.4013 & 0.3991 \\ -0.8998 & 0.3001 & -1.5004 \\ 1.0997 & 0.9997 & -0.4020 \end{bmatrix} \tag{5.12a}$$

$$\widehat{B}_D = \begin{bmatrix} 0.0994 & -0.2966 \\ -0.0997 & -0.6997 \\ 0.6983 & -0.9977 \end{bmatrix} \tag{5.12b}$$

$$\underline{\widehat{\mathcal{W}}_D} = \begin{bmatrix} -0.0392 \\ -0.0395 \\ -0.0361 \end{bmatrix}, \; \overline{\widehat{\mathcal{W}}_D} = \begin{bmatrix} 0.0390 \\ 0.0389 \\ 0.0387 \end{bmatrix}. \tag{5.12c}$$

Thus, the state feedback matrix in (5.11) stabilizes the estimated system $\left(\widehat{A}_D, \widehat{B}_D\right)$. Nevertheless, any stabilizing feedback matrix could be deployed, e.g., using LQR-based controller synthesis [155]. Moreover, because $\widehat{\mathcal{W}}_D \subset \mathcal{W}_D$, model invalidation will likely occur, requiring us to update $\widehat{\mathcal{W}}_D$ online. To cover $\mathcal{X}$, we choose the columns of $G_{\text{fixed}} \in \mathbb{R}^{3\times 70}$ in (3.20) to be 70 uniformly distributed points around the unit ball corresponding to the Euclidean norm.

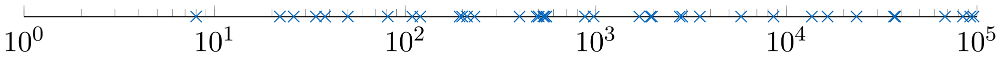
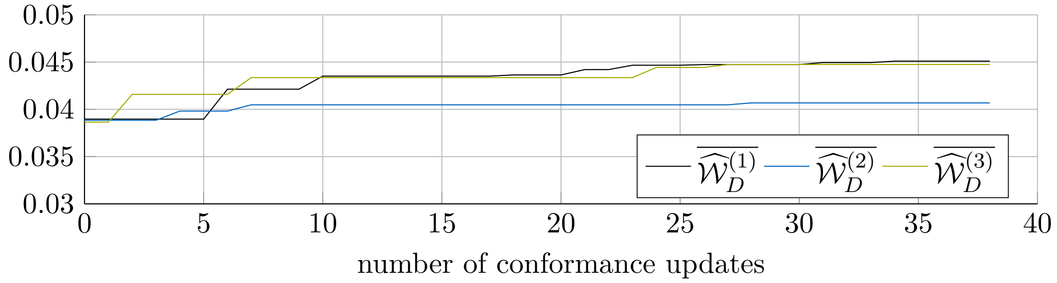
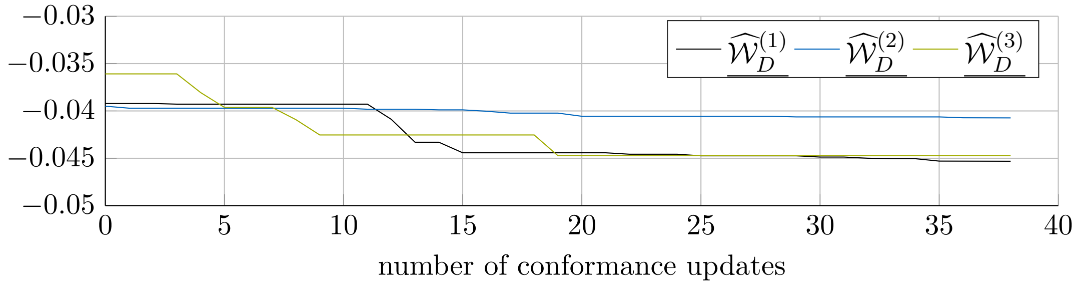


Figure 5.5: Online conformance updates of the three-dimensional system. The time steps $k \in \mathbb{N}_{[0,10^5]}$ are marked when conformance updates are performed, i.e., when $w_D(t_k) \notin \widehat{\mathcal{W}}_D$ is detected.



(a) Upper bounds of $\widehat{\mathcal{W}}_D$.



(b) Lower bounds of $\widehat{\mathcal{W}}_D$.

Figure 5.6: Evolution of estimated multidimensional disturbance interval $\widehat{\mathcal{W}}_D$ of the three-dimensional system, which is initialized in (5.12c).

In Fig. 5.5, we plot the 38 time steps $k \in \mathbb{N}$ when $w_D(t_k) \notin \widehat{\mathcal{W}}_D$ is detected. At these time steps, we update the model, the safe set, and the safe backup controller, as proposed in Subsection 5.4.4. Using a logarithmic scale makes it clear that most updates occur early on.

In Fig. 5.6, we visualize the evolution of the lower and upper bounds of the estimated disturbance set $\widehat{\mathcal{W}}_D$, which is initialized in (5.12c). As more uniformly sampled disturbances are gathered online, the bound changes in all three dimensions become smaller.

In Fig. 5.7, we show two-dimensional projections of the initial optimal safe set and a tight RCI under-approximation of the initial MRCI set based on the estimated disturbance bounds in (5.12c). In addition, we visualize the 38 updated optimal safe sets corresponding to updated conformant models. As can be seen in the $x^{(1)}$-$x^{(2)}$-plot in Fig. 5.7, the updated safe sets shrink in some generator directions but also grow in some others. Because computing a tight RCI under-approximation of the initial MRCI set takes more than 1 min, these computations are unsuitable for online updating. Nevertheless, our online conformance updates, which include updating $\widehat{\mathcal{W}}_D$, verifying the existence of a small safe set $\mathcal{S}_{Kx} \subseteq \mathcal{X}$, and updating $\mathcal{Z}_x^\star(t_0) \subseteq \mathcal{X}$ along with corresponding $\mathcal{Z}_u^{\star,\circ}(\cdot)$, take 57 ms on average with a standard deviation of 4 ms.

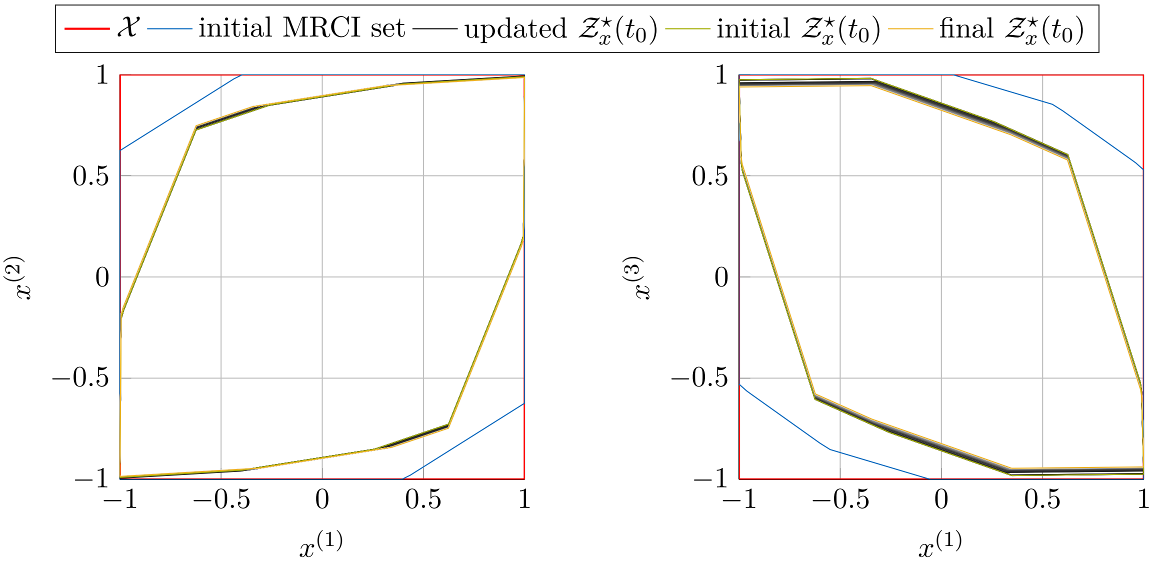


Figure 5.7: Evolution of large safe sets of the three-dimensional system. The updated optimal safe sets corresponding to updated conformant models are shown, where a lighter gray tone corresponds to a higher number of updates.

Because the sampling period is $\Delta t = 100\,\mathrm{ms}$, our online conformance updates are performed in real time.

5.5.3 Nonlinear Continuous-Time Six-Dimensional System

To demonstrate the generalizability of our DT safety filter approach, we consider the nonlinear, CT, longitudinal quadrotor model proposed in [143, 247], which is briefly summarized subsequently. The unknown system is described by the set of ordinary differential equations

$$\dot{x}^{(1)} = x^{(3)} \tag{5.13a}$$

$$\dot{x}^{(2)} = x^{(4)} \tag{5.13b}$$

$$\dot{x}^{(3)} = u^{(1)} n_1 \sin\left(x^{(5)}\right) \tag{5.13c}$$

$$\dot{x}^{(4)} = u^{(1)} n_1 \cos\left(x^{(5)}\right) - g \tag{5.13d}$$

$$\dot{x}^{(5)} = x^{(6)} \tag{5.13e}$$

$$\dot{x}^{(6)} = -d_0 x^{(5)} - d_1 x^{(6)} + n_0 u^{(2)}, \tag{5.13f}$$

where $x^{(1)}$ to $x^{(6)}$ represent the horizontal and vertical positions, horizontal and vertical velocities, roll, and roll velocity, respectively. The constant parameters are $g = 9.81$, $d_0 = 70$, $d_1 = 17$, $n_0 = 55$, $n_1 = 0.89/1.4$, and the multidimensional state and input constraint intervals are described by

$$\begin{aligned}
\underline{\mathcal{X}} &= \begin{bmatrix} -1.7 & 0.3 & -0.8 & -1 & -\pi/12 & -\pi/2 \end{bmatrix}^T \\
\overline{\mathcal{X}} &= \begin{bmatrix} 1.7 & 2.0 & 0.8 & 1 & \pi/12 & \pi/2 \end{bmatrix}^T \\
\underline{\mathcal{U}} &= \begin{bmatrix} g/n_1 - 1.5 & -\pi/12 \end{bmatrix}^T \\
\overline{\mathcal{U}} &= \begin{bmatrix} g/n_1 + 1.5 & \pi/12 \end{bmatrix}^T .
\end{aligned}$$

To satisfy our assumption that $\mathcal{X}$ and $\mathcal{U}$ contain the origin, we perform a simple coordinate transformation, i.e., we shift $x^{(2)}$ by -1.15 and $u^{(1)}$ by $-g/n_1$. We generate training data $\{x(t_i), u(t_i), x(t_{i+1})\}_{i=1}^{100}$ by using the MATLAB function ode45[1] to solve (5.13) and by sampling uniformly from $\mathcal{X}$ and $\mathcal{U}$. In addition, we compute the stabilizing feedback matrix $K \in \mathbb{R}^{2\times 6}$ using LQR-based controller synthesis [155], where the state and input weighting matrices are $Q = 10I$ and $R = I$. The fixed generator matrix $G_{\text{fixed}} \in \mathbb{R}^{6\times 48}$ in (3.20) is taken from [143]. The initial state is $x(t_0) = \begin{bmatrix} 0 & 1.15 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ and the initial input is $u(t_0) = \begin{bmatrix} g/n_1 & 0 \end{bmatrix}^T$. Moreover, the desired input $\tilde{u}(t_k) \in \mathbb{R}^2$ is uniformly sampled online from $\mathcal{U}$ for all $k \in \mathbb{N}_{[0,10^5]}$.

Because solving (3.20) initially for the offline training data takes 19 s, we slightly simplify (3.20) to enable real-time conformance updates. In particular, we restrict $\mathcal{Z}_x(t_0)$ and $\mathcal{Z}_u(\cdot)$ to be scaled versions of the optimal initial zonotopes, as shown in Fig. 5.8. As a result, our update takes only 145 ms on average with a standard deviation of 8 ms. Thus, our approach can update formal safety guarantees at sampling times for nonlinear, CT systems in real time.

To demonstrate the difficulty of this numerical example, we compare our results with two existing methods for computing large safe sets. Because the approach in [40] has an exponential computational complexity with respect to the state space dimension, we abort the corresponding computations prematurely after 24 h. We also use the method in [137], which requires the linear system to be presented in controller canonical form. Using the corresponding publicly available code[2], the transformation of our initial conformant model into this form involves the inverse of a matrix whose condition number is greater than 10^6, which leads to significant numerical errors.

5.5.4 Continuous-Time Twelve-Dimensional System

To demonstrate the scalability of our approach, we consider the twelve-dimensional, underactuated, CT quadrotor model presented in Subsection 4.5.2 [142, 219]. Because the state of the system is assumed to be measurable in this chapter, we ignore the output equations in Subsection 4.5.2. In addition, we assume the uncertain wind to be modeled by the unknown, bounded disturbance $\begin{bmatrix} w^{(4)} & w^{(5)} & w^{(6)} \end{bmatrix}^T \in [-0.05, 0.05]^3$ that affects only the three spatial velocities.

[1] https://mathworks.com/help/matlab/ref/ode45.html

[2] https://github.com/janis10/cis2m

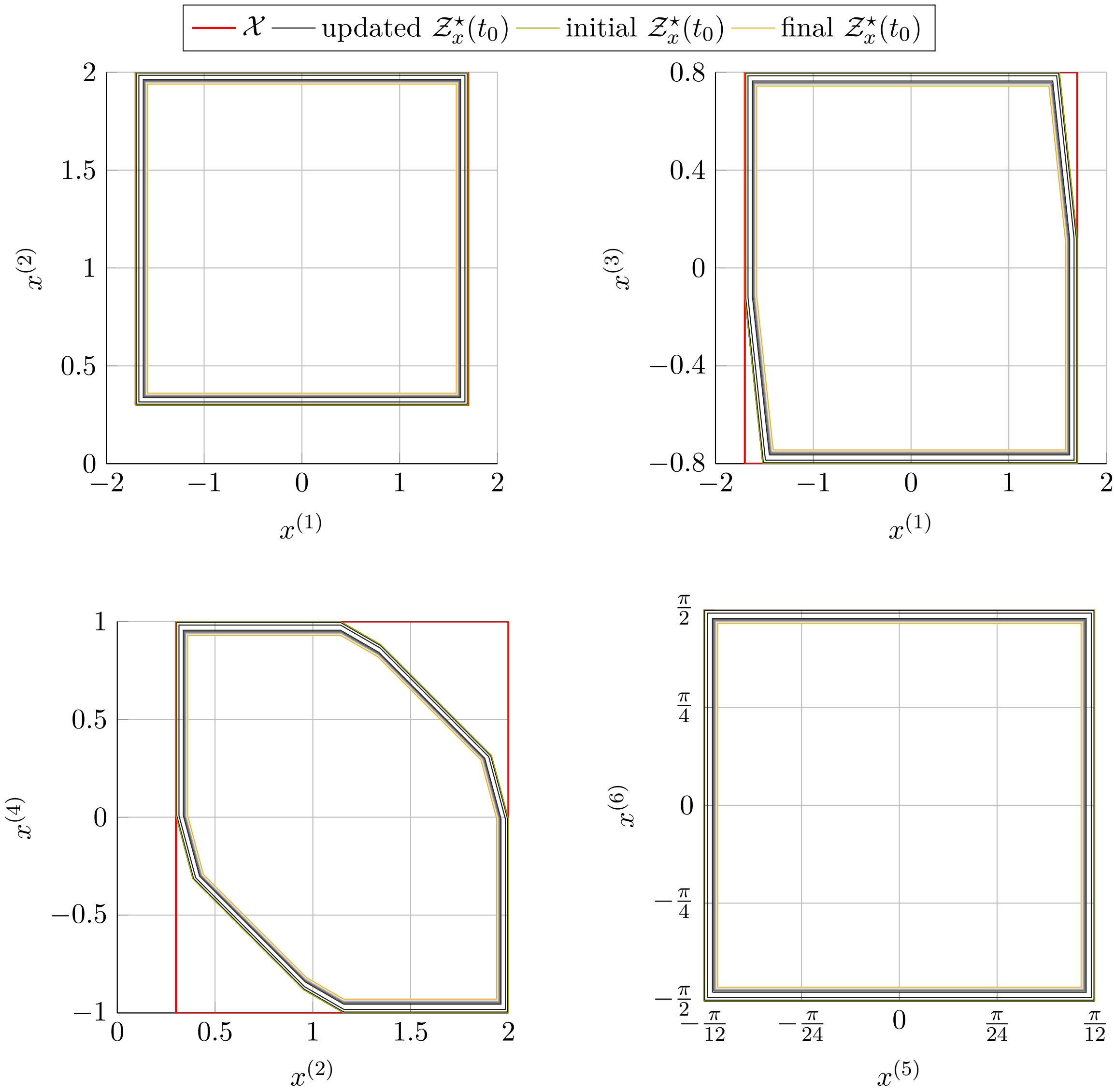


Figure 5.8: Evolution of large safe sets of the six-dimensional system. The updated optimal safe sets corresponding to updated conformant models are shown, where a lighter gray tone corresponds to a higher number of updates.

We generate training data $\{x(t_i), u(t_i), x(t_{i+1})\}_{i=1}^{1000}$ by using the MATLAB function ode45 to solve the system of linear differential equations and by sampling uniformly from the state and input constraint sets. In addition, we compute the stabilizing feedback matrix $K \in \mathbb{R}^{4\times 12}$ using LQR-based controller synthesis [155], where the state and input weighting matrices are $Q = 10I$ and $R = I$. The fixed generator matrix $G_{\text{fixed}} \in \mathbb{R}^{12\times 52}$ in (3.20) is obtained following the approach in [143], i.e., by examining the sparsity of the system matrix. The initial state $x(t_0) \in \mathbb{R}^{12}$ and the initial input $u(t_0) \in \mathbb{R}^4$ are the origin. Moreover, the desired input $\tilde{u}(t_k) \in \mathbb{R}^4$ is uniformly sampled online from the input constraint set for all $k \in \mathbb{N}_{[0,10^5]}$.

Two-dimensional projections of the initial solution of (3.20) are shown in Fig. 5.9. Because solving (3.20) initially for the offline training data takes 40 min, we slightly simplify (3.20) analogously to Subsection 5.5.3. As a result, our 20 updates take 1.01 s on average with a standard deviation of 85 ms. In summary, our approach quickly updates formal safety guarantees for medium-sized problems.

5.6 Summary

In this chapter, we have presented supervisory safety filters that enable formal safety guarantees for any controller. Only if the desired input of the corresponding high-performance controller might lead to leaving our large safe set in the future, it is modified in the least restrictive way.

Unlike most other work on robust controller synthesis, we make no assumptions about the availability of a system model along with its corresponding disturbance set. Thus, we perform offline set membership identification based on a finite set of available training data. Because a new measurement obtained online might invalidate the formal safety guarantees of our safety filter, fast online conformance updating is crucial. In contrast to existing work, our updates are performed in real time, even for medium-sized problems, as shown in Section 5.5. These real-time updates are enabled by designing our update procedure to be independent of the number of measurements and by using scalable reachability analysis as well as convex optimization algorithms. We have demonstrated our supervisory safety filter approach's effectiveness, generalizability, and scalability using four numerical examples taken from the literature, including a six-dimensional, nonlinear quadrotor system.

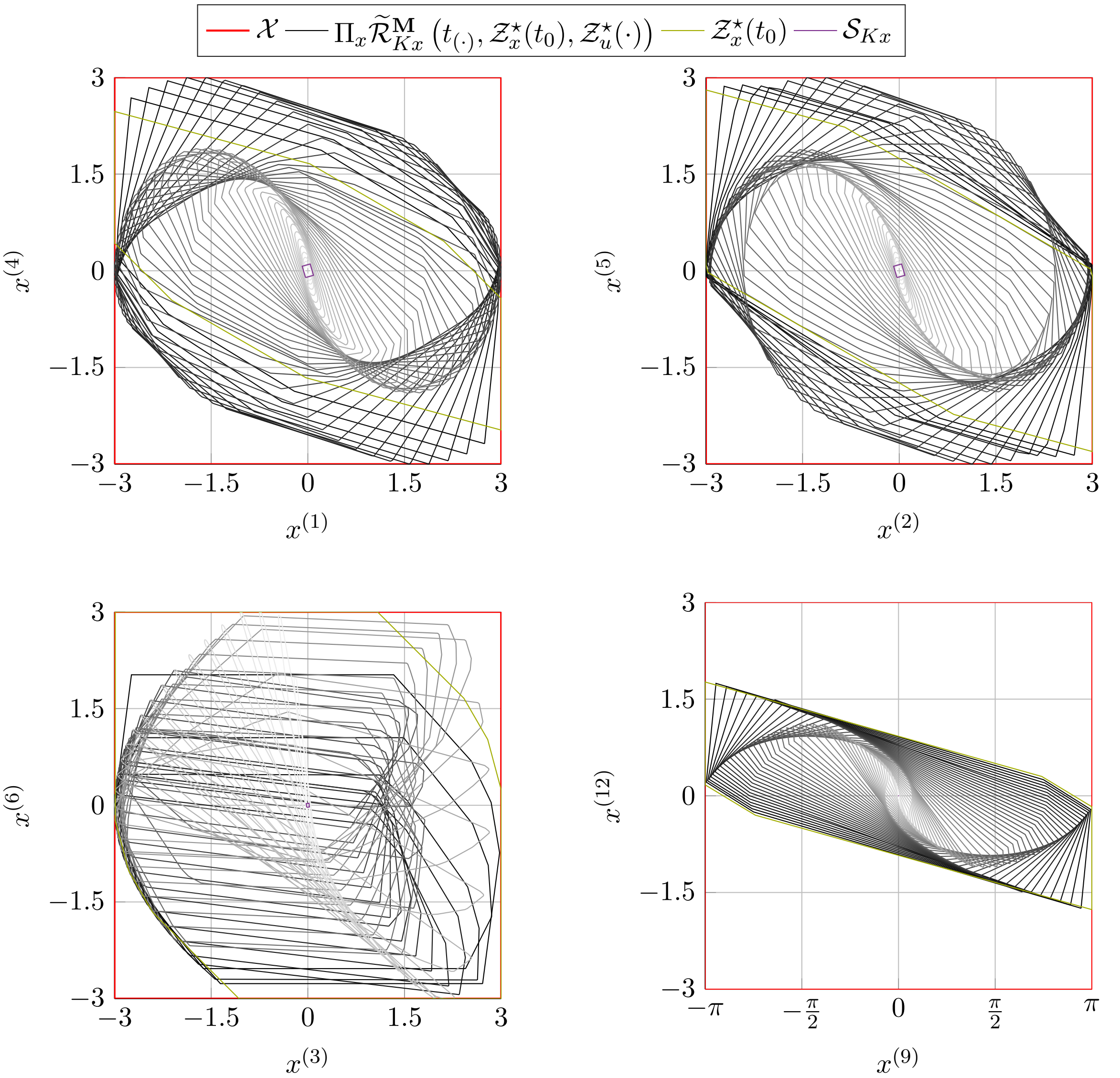


Figure 5.9: Initial safe sets of the twelve-dimensional system. In addition, reachable sets are shown, where a lighter gray tone corresponds to a larger prediction horizon.

6 Safety Verification of Autonomous Vehicles

In this chapter, which is based on [46], we verify the safety of autonomous vehicles online while considering the uniqueness of each traffic situation. After reviewing the relevant literature in Section 6.1, we formulate the safety verification goal in Section 6.2. In Section 6.3, we present the traditional set-based verification approach. To guarantee timely, safe motion plans, we propose an anytime approach that provides conservative safety verification results quickly and continually refines them until the available computation time is elapsed. Thus, our algorithm can be interrupted at any time while using the available computational resources optimally. After presenting this anytime approach in Section 6.4, we demonstrate its effectiveness using two traffic scenario benchmarks in Section 6.5. Finally, we summarize this chapter in Section 6.6.

6.1 Introduction and State of the Art

In the previous chapters, we have mainly focused on time-invariant systems affected by unknown but bounded disturbances. For this setting, we have proposed multiple robust control approaches that provide formal safety guarantees for an infinite time horizon. In this chapter, we extend this setting by considering not only the system that we want to control but also other safety-relevant agents that are located in the shared environment. Thus, we compute reachable sets not only for the controlled system but also for models of the other uncontrolled agents to provide formal safety guarantees, which claim that the intersections of reachable sets are empty. In addition, we perform safety verification as a special case of a supervisory safety filter, as shown in Fig. 6.1. Thus, we want to answer the question "Is the desired control input u_{desired} safe?" while assuming the existence of a safety-preserving or safe backup controller that provides a safe input if u_{desired} is unsafe [17, 248, 249]. Because of its high practical relevance and interdisciplinary challenge [126, 250], we consider autonomous driving in this chapter, i.e., we want to formally verify that the desired trajectory of the controlled autonomous vehicle is safe.

Predicting the movement of other traffic participants is crucial for motion planning [251–254], threat assessment [255, 256], and safety verification of autonomous vehicles [17, 18]. Several techniques have been developed based on their intended use. Rather simple safety metrics have been proposed to warn drivers based on predicting a single behavior of other traffic participants, such as the time to collision [257, 258] and combinations of several metrics [259, 260]. In addition, collision mitigation systems that typically require short prediction horizons often rely on predicting a single future behavior [261, 262]. Moreover, threat assessments mainly use stochastic predictions [263], either by performing Monte Carlo simulations [264, 265], which consider a finite number of future trajectories, or by predicting occupancy probability distributions [266–268], which account for infinitely many possible behaviors. However, none of these methods can formally verify the safety of the desired trajectory.

When using the term safety in the context of not necessarily cooperative autonomous vehicles, as considered in this chapter, we refer to legal safety [17, 269]. In this safety concept, the

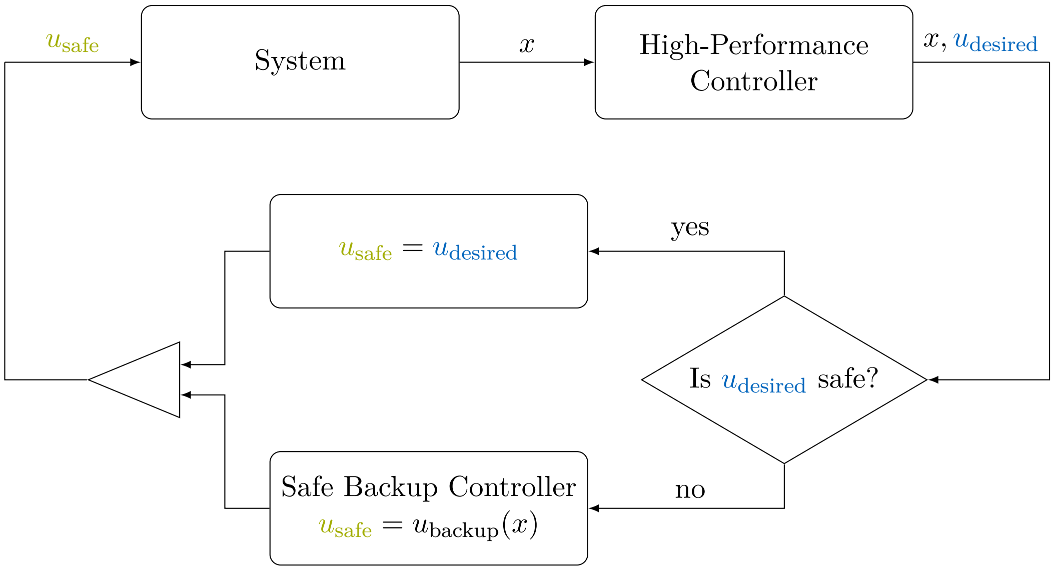

u_{safe}
System
x
High-Performance Controller
x, u_{desired}
$u_{\text{safe}} = u_{\text{desired}}$
yes
Is u_{desired} safe?
no
Safe Backup Controller
$u_{\text{safe}} = u_{\text{backup}}(x)$

Figure 6.1: Safety verification concept.

ego vehicle must always be in a state that is not causing a collision while assuming that all other traffic participants obey the traffic rules [270–272]. Based on the definition of legal safety and induction, safe sets can be defined and constructed [248, 249], e.g., by realizing a safe distance behind another traffic participant or a standstill [17, 273]. Then, if the state of the ego vehicle lies within such a safe set, safety for an infinite time horizon can be formally guaranteed. In addition to legal safety, there exist other closely related definitions of safety [274, 275]. Moreover, these safety concepts are used not only in autonomous vehicles but also in, e.g., robotics [24, 276–278].

In [269], the importance of set-based prediction of other traffic participants is highlighted. However, no prediction algorithm is provided for formally computing reachable or occupancy sets, i.e., sets of occupied X-Y-positions. In [17, 279–281], reachability analysis is used to obtain over-approximations of the occupancy sets of all surrounding safety-relevant traffic participants, i.e., all possible behaviors that satisfy the traffic rule assumption are captured. When the trajectory of the ego vehicle, i.e., the vehicle performing these predictions, does not intersect the over-approximative occupancy sets of other traffic participants at any time, it can be deduced that no collision occurs. In Fig. 6.2, we show an example of these predicted occupancy sets of all surrounding safety-relevant traffic participants and the ego vehicle for a finite prediction horizon $N \in \mathbb{N}_{>0}$.

On the one hand, considering all possible future behaviors of the other traffic participants increasingly restricts the solution space of the ego vehicle's trajectory, the larger N is chosen. For instance, when doubling N in the example in Fig. 6.2, the occupancy sets of the third traffic participant will intersect the desired trajectory of the ego vehicle, which wants to make a left turn. Thus, set-based safety verification approaches are usually used to formally verify trajectories of short time horizons.

On the other hand, there exist long-term trajectories of the ego vehicle that are initially unsafe for some parts when considering all possible future behaviors. Nevertheless, such trajectories

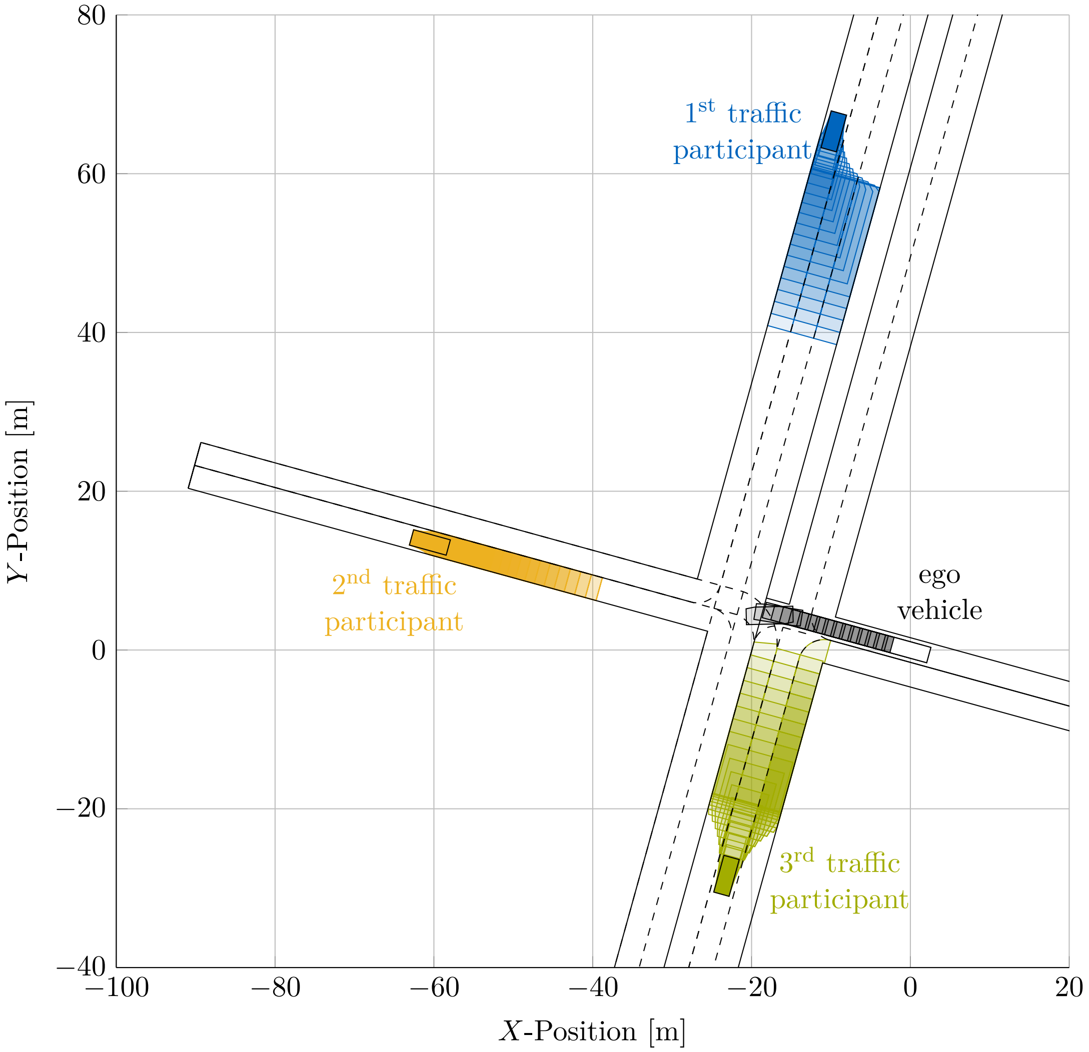


Figure 6.2: Initial occupancy sets for uncontrolled intersection, i.e., no traffic lights control the traffic flow. The sampling period is $\Delta t = 0.1\,\mathrm{s}$ and the prediction horizon is $N = 17$, i.e., the over-approximative occupancy sets of all other safety-relevant traffic participants are predicted for the next 1.7 s. Because the desired trajectory of the ego vehicle does not intersect them at any time, it can be safely executed.

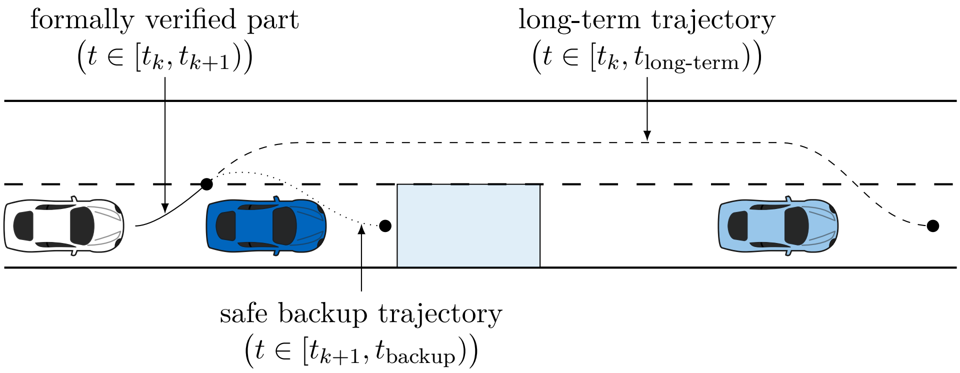


Figure 6.3: Comparison of long-term and safe backup trajectory planning of the white ego vehicle, which wants to overtake the other blue traffic participant. The predicted occupancy set of the other vehicle at $t_{\text{backup}} \in \mathbb{R}_{>0}$ and the most likely position of the other vehicle at $t_{\text{long-term}} \in \mathbb{R}_{>0}$ are shown by the lighter blue rectangle and vehicle, respectively.

often become safe because uncertainty about the other traffic participants' future behaviors is reduced as time proceeds. For instance, the third traffic participant in Fig. 6.2 might slow down significantly in preparation for making a right turn. Thus, an off-the-shelf motion planner can compute a long-term trajectory based on the most likely maneuvers of the other safety-relevant traffic participants, as shown in Fig. 6.3. Then, the introduced set-based safety verification method is applied only to the first part of the desired motion plan, i.e., this method can be seen as moving horizon verification or model predictive verification. If formally verified, this first part of the long-term trajectory can be safely executed. Conversely, if the verification fails, the previously verified safe backup or fail-safe maneuver of the safe backup controller is executed, ensuring the safety of the ego vehicle for an infinite time horizon.

A challenging aspect of set-based prediction and verification that has not yet received much attention is the high dependence of computation time on the varying number of surrounding traffic participants, which causes significant variations in computational demand. For instance, at a busy intersection in an urban area, this number can easily vary from only a few to more than 100 because of many surrounding pedestrians, cyclists, and dog walkers. However, the computing resources of the ego vehicle are limited. Using conventional prediction techniques in such congested traffic scenarios requires, e.g., disregarding some safety-relevant traffic participants from the prediction or performing fewer simulations in a Monte Carlo simulation. However, these measures inevitably reduce the safety of the ego vehicle.

It is clear from the presented literature review that it is an unresolved issue to provide an online safety verification approach that considers the uniqueness of each traffic situation. In this chapter, we address this issue by proposing an anytime approach that quickly provides conservative formal verification results and continually refines them until the available computation time is elapsed. Thus, our algorithm aims to guarantee timely, safe trajectories of the controlled autonomous vehicle and can be interrupted at any time while optimally using the available computational resources. Moreover, if the safety of the desired trajectory cannot be verified in time, the safety-preserving backup controller overwrites the desired control inputs. Before we present our anytime safety verification approach, we formulate the verification problem of this chapter in the following section.

6.2 Problem Formulation

Typically, an exact model $\mathbf{M}_{\text{exact}}$ of another traffic participant is not known by the ego vehicle unless, e.g., transmitted by vehicle-to-vehicle communication [170,171]. Thus, we use $n_{\mathbf{M}} \in \mathbb{N}_{>0}$ models of different complexities that are collected in the vector $\mathbf{M} = \begin{bmatrix} \mathbf{M}^{(1)} & \mathbf{M}^{(2)} & \dots & \mathbf{M}^{(n_{\mathbf{M}})} \end{bmatrix}$ and that are based on physical constraints and traffic rules [270–272,281]. If $\mathbf{M}^{(i)}$ with $i \in \mathbb{N}_{[1,n_{\mathbf{M}}]}$ is a conformant model [235], $\mathbf{M}^{(i)}$ is also known as an abstraction of $\mathbf{M}_{\text{exact}}$. In this chapter, we assume that all models are abstractions of the unknown $\mathbf{M}_{\text{exact}}$ and that the reachable set $\widetilde{\mathcal{R}}_{\text{over}}^{\mathbf{M}^{(i)}}(\cdot,\cdot) \subset \mathbb{R}^{n_{x_i}}$ can be efficiently computed. In addition, all safety-relevant traffic participants are assumed to be detected by the sensors of the ego vehicle.

The i^{th} model $\mathbf{M}^{(i)} = (f_i(x_i, \mathbf{0}, w_i), \mathcal{W}_i, \text{CT})$ with $i \in \mathbb{N}_{[1,n_{\mathbf{M}}]}$ represents the uncontrolled, CT system in (2.16) that evolves according to

$$\dot{x}_i(t) = f_i(x_i(t), \mathbf{0}, w_i(t)), \tag{6.1}$$

where $x_i(t) \in \mathbb{R}^{n_{x_i}}$ and $w_i(t) \in \mathbb{R}^{n_{w_i}}$ denote the i^{th} state and i^{th} disturbance at time $t \in \mathbb{R}_{\geq 0}$, respectively, whereas the control input is element of $\mathbb{R}^0$. The disturbances, such as the steering rate or the acceleration of the other traffic participant, are unknown but bounded by the i^{th} disturbance set $\mathcal{W}_i \subset \mathbb{R}^{n_{w_i}}$.

Subsequently, we compute reachable sets for the i^{th} model $\mathbf{M}^{(i)}$, the initial state set $\mathcal{Z}_{x_i}(t_0) \subset \mathbb{R}^{n_{x_i}}$, discrete sampling times $t_k = k\Delta t$, and time intervals $[t_k, t_{k+1})$ with $k \in \mathbb{N}$. Identical to the reachability analysis in Section 3.3, this is achieved by computing reachable sets for consecutive time steps of size Δt until the specified time is reached. Similar to (3.5) and (3.11), we introduce the following recursively defined set sequence:

$$\widetilde{\mathcal{R}}_{\mathbf{0}}^{\mathbf{M}^{(i)}}\left(t_0, \mathcal{Z}_{x_i}(t_0)\right) = \mathcal{Z}_{x_i}(t_0) \tag{6.2a}$$

$$\widetilde{\mathcal{R}}_{\mathbf{0}}^{\mathbf{M}^{(i)}}\left(t_k, \mathcal{Z}_{x_i}(t_0)\right) = \widetilde{\mathcal{R}}_{\text{over}}^{\mathbf{M}^{(i)}}\left(\Delta t, \widetilde{\mathcal{R}}_{\mathbf{0}}^{\mathbf{M}^{(i)}}\left(t_{k-1}, \mathcal{Z}_{x_i}(t_0)\right)\right). \tag{6.2b}$$

Similar to Theorems 3.2 and 3.7, we prove in the following proposition that the sets in (6.2) are over-approximating the reachable sets of $\mathbf{M}^{(i)}$ when using no control input.

Proposition 6.1 (Set Propagation using No Control): For all $x_i(t_0) \in \mathcal{Z}_{x_i}(t_0)$, applying no control input to $\mathbf{M}^{(i)} = (f_i(x_i, \mathbf{0}, w_i), \mathcal{W}_i, \text{CT})$ results in

$$x_i(t_k) \in \widetilde{\mathcal{R}}_{\mathbf{0}}^{\mathbf{M}^{(i)}}\left(t_k, \mathcal{Z}_{x_i}(t_0)\right),$$

where $k \in \mathbb{N}$. ■

The proof is omitted because Proposition 6.1 follows directly from the definition of over-approximative reachable sets and set propagation.

Up to now, we have focused on performing reachability analysis for discrete sampling times. Nevertheless, the safety of the ego vehicle must be verified not only at but also between sampling times. Thus, based on Proposition 6.1 and (2.22b), we compute reachable sets for an arbitrary time interval $[t_k, t_{k+1})$ according to

$$\widetilde{\mathcal{R}}_{\mathbf{0}}^{\mathbf{M}^{(i)}}\left([t_k, t_{k+1}), \mathcal{Z}_{x_i}(t_0)\right) = \widetilde{\mathcal{R}}_{\text{over}}^{\mathbf{M}^{(i)}}\left([0, \Delta t), \widetilde{\mathcal{R}}_{\mathbf{0}}^{\mathbf{M}^{(i)}}\left(t_k, \mathcal{Z}_{x_i}(t_0)\right)\right).$$

In summary, we can efficiently compute the set of states that are reachable for all $x_i(t_0) \in \mathcal{Z}_{x_i}(t_0)$ when applying no control input to $\mathbf{M}^{(i)} = (f_i(x_i, \mathbf{0}, w_i), \mathcal{W}_i, \text{CT})$.

Similar to the definitions of Π_x and Π_u in (2.19), we introduce the mapping $\Pi_{XY}^{\mathbf{M}^{(i)}} : 2^{\mathbb{R}^{n_{x_i}}} \rightarrow 2^{\mathbb{R}^2}$ to project a set of states of another traffic participant onto its set of occupied X-Y-positions. By projecting the reachable set of another traffic participant onto the two-dimensional set of occupied X-Y-positions, the occupancy set is obtained, which is introduced in the following definition.

Definition 6.2 (Occupancy Set of Other Traffic Participant): The occupancy set for the i^{th} model $\mathbf{M}^{(i)}$, the initial state set $\mathcal{Z}_{x_i}(t_k) \subset \mathbb{R}^{n_{x_i}}$ at t_k, and the prediction time interval $[t_{j-1}, t_j)$ with $j \in \mathbb{N}_{[1,N]}$ is $\Pi_{XY}^{\mathbf{M}^{(i)}}\left(\widetilde{\mathcal{R}}_{\mathbf{0}}^{\mathbf{M}^{(i)}}\left([t_{j-1}, t_j), \mathcal{Z}_{x_i}(t_k)\right)\right)$. We use $\mathcal{Q}_k^{(j)}\left(\mathbf{M}^{(i)}\right)$ to obtain a more concise notation for representing this occupancy set. ■

Because all models are abstractions, it follows that

$$\mathcal{Q}_k^{(j)}\left(\mathbf{M}_{\text{exact}}\right) \subseteq \bigcap_{i=1}^{n_{\mathbf{M}}} \mathcal{Q}_k^{(j)}\left(\mathbf{M}^{(i)}\right) \tag{6.3}$$

for any $j \in \mathbb{N}_{>0}$ and $k \in \mathbb{N}$ [17, Prop. 5.1]. This set relation allows us to over-approximate the exact occupancy set of another traffic participant by intersecting the occupancy sets of $n_{\mathbf{M}}$ different models. Thus, the over-approximation becomes tighter each time a new model is added, as illustrated in Fig. 6.4. We also want to mention that, in general, $\mathcal{Q}_k^{(j)}(\mathbf{M}^{(i)}) \nsubseteq \mathcal{Q}_k^{(j)}(\mathbf{M}^{(i+1)})$ and $\mathcal{Q}_k^{(j)}(\mathbf{M}^{(i+1)}) \nsubseteq \mathcal{Q}_k^{(j)}(\mathbf{M}^{(i)})$ for any $i \in \mathbb{N}_{[1,n_{\mathbf{M}}-1]}$, j, and k, as it is the case for $\mathbf{M}^{(2)}$ and $\mathbf{M}^{(3)}$ in Fig. 6.4.

Example 6.3 (Infinite-Acceleration-Based Model): A simple first model $\mathbf{M}^{(1)}$ can be constructed by modeling the other traffic participant as a point mass [282, 283]. The corresponding dynamics is

$$\dot{x}_1(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_1(t) + w_1(t),$$

where $x_1^{(1)}$ to $x_1^{(4)}$ represent the X-position, Y-position, X-velocity, and Y-velocity, respectively. Moreover, the first two elements $w_1^{(1)}$ and $w_1^{(2)}$ of the disturbance vector are zero, and the last two elements are constrained by $\sqrt{\left(w_1^{(3)}\right)^2 + \left(w_1^{(4)}\right)^2} \leq a_{\max}$, where $a_{\max} \in \mathbb{R}_{>0}$ denotes a maximum absolute acceleration. By allowing $a_{\max}$ to be infinite and assuming a maximum absolute velocity $v_{\max} \in \mathbb{R}_{>0}$, the exact set of occupied X-Y-positions during $[t_{j-1}, t_j)$ is

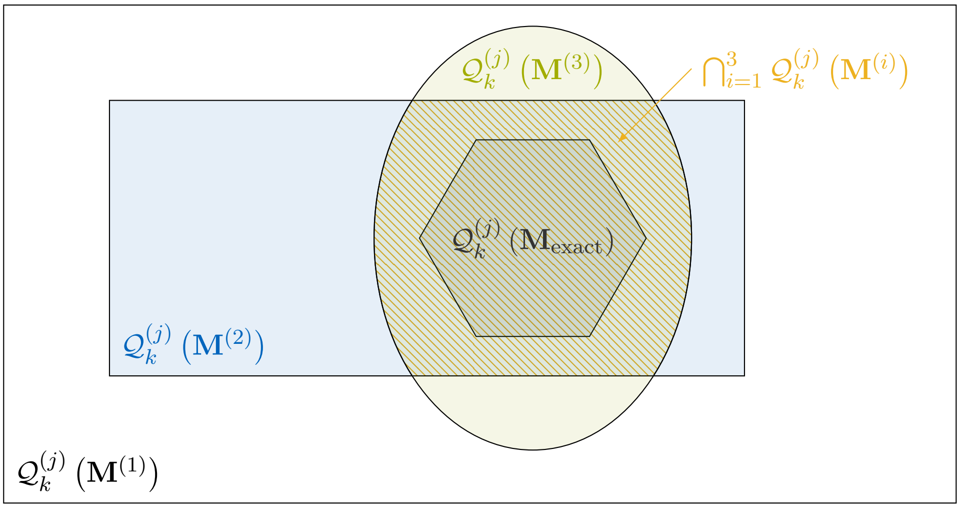

$\mathcal{Q}_k^{(j)}\left(\mathbf{M}^{(3)}\right)$
$\bigcap_{i=1}^{3} \mathcal{Q}_k^{(j)}\left(\mathbf{M}^{(i)}\right)$
$\mathcal{Q}_k^{(j)}\left(\mathbf{M}_{\text{exact}}\right)$
$\mathcal{Q}_k^{(j)}\left(\mathbf{M}^{(2)}\right)$
$\mathcal{Q}_k^{(j)}\left(\mathbf{M}^{(1)}\right)$

Figure 6.4: Occupancy set intersection of different models $\mathbf{M}^{(i)}$ with $i \in \mathbb{N}_{[1,3]}$ to tightly over-approximate the occupancy set of the unknown model $\mathbf{M}_{\text{exact}}$.

$\left\langle c, (t_j v_{\max})^2 I \right\rangle_E \subset \mathbb{R}^2$, where $c = \left[x_1^{(1)}(0) \quad x_1^{(2)}(0)\right]^T$ is the center of this circle. A simple zonotopic over-approximation is $\langle c, t_j v_{\max} I\rangle_Z$, i.e., given by a square with length $2t_j v_{\max}$ and center c. Finally, to obtain the occupancy set, the vehicle dimensions must be added by Minkowski addition based on the orientation of the considered traffic participant. ■

Analogous to Definition 6.2, the occupancy set of the ego vehicle based on the desired trajectory at t_k and the prediction time interval $[t_{j-1}, t_j)$ is denoted by $\mathcal{T}_k^{(j)} \subset \mathbb{R}^2$. We assume that the uncertainties due to an imperfect tracking controller and the dimensions of the ego vehicle are already included in the set $\mathcal{T}_k^{(j)}$ [17]. Then, the goal in this chapter is to quickly verify the safety of the desired trajectory of the ego vehicle, i.e., to quickly verify at sampling time t_k that $\mathcal{Q}_k^{(j)}\left(\mathbf{M}_{\text{exact}}\right) \cap \mathcal{T}_k^{(j)} \equiv \emptyset$ for all $j \in \mathbb{N}_{[1,N]}$.

6.3 Traditional Safety Verification

In this section, we provide an overview of the traditional safety verification approach that uses set-based predictions of other traffic participants [279, 281]. The corresponding method is presented in Alg. 6.1 and is executed in parallel for each surrounding safety-relevant traffic participant at t_k, where the function `any` returns Boolean "true" if any element of the Boolean input vector is "true"; otherwise, it returns "false". Essentially, this algorithm iterates over all N prediction time intervals and checks if the occupancy sets of the ego vehicle and the considered traffic participant intersect, indicating a potential future collision.

Alg. 6.1 has two outputs: The first output of Alg. 6.1 is Boolean "true" if there exists a possible collision between the ego vehicle and the other traffic participant; otherwise, it is "false". The second output is the vector of occupancy sets of the other traffic participant $\mathcal{O}_k = \left[\mathcal{O}_k^{(1)} \quad \mathcal{O}_k^{(2)} \quad \dots \quad \mathcal{O}_k^{(N)}\right]$. Analogous to $\mathcal{O}_k$, we denote the vector of occupancy sets of

Algorithm 6.1 Traditional safety verification

```
Input: $\mathcal{T}_k, \mathcal{Q}_k, \mathbf{M}, N$
Output: $a_c, \mathcal{O}_k$
1: updateModelParameters()
2: for all $j \in \mathbb{N}_{[1,N]}$ do
3:     $\mathcal{O}_k^{(j)} \leftarrow \bigcap_{i=1}^{n_\mathrm{M}} \mathcal{Q}_k^{(j)}\left(\mathbf{M}^{(i)}\right)$
4:     $c^{(j)} \leftarrow \mathcal{O}_k^{(j)} \cap \mathcal{T}_k^{(j)} \neq \emptyset$
5: end for
6: $a_c \leftarrow$ any($c$)
```

the ego vehicle by $\mathcal{T}_k = \begin{bmatrix}\mathcal{T}_k^{(1)} & \mathcal{T}_k^{(2)} & \dots & \mathcal{T}_k^{(N)}\end{bmatrix}$, which is the first input of Alg. 6.1. The second, third, and fourth inputs are the vector of occupancy set operators $\mathcal{Q}_k$, the vector of models $\mathbf{M}$, and the prediction horizon N, respectively. Subsequently, we describe the main steps of Alg. 6.1 in more detail.

As mentioned in Section 6.2, the parameters of the models collected in $\mathbf{M}$ are primarily based on traffic rules and physical constraints [270–272, 281]. For instance, some models check traffic rule compliance of other traffic participants, such as staying in their lane or exceeding a maximum velocity $v_{\max} \in \mathbb{R}_{>0}$, e.g., given by an exact or relaxed speed limit. If the ego vehicle detects a conformance violation, the corresponding parameter is adapted or removed to restore model conformance online, similar to Subsection 5.4.4. For instance, restoring model conformance is achieved by increasing the individual speed limit or removing the assumption that the other traffic participant will follow lanes in the future. Otherwise, the models would no longer be conformant to the real system. This conformance-preserving parameter updating procedure is handled by the function `updateModelParameters`, which is called in line 1 of Alg. 6.1. In line 3 of Alg. 6.1, the occupancy set $\mathcal{O}_k^{(j)}$ for the other traffic participant at t_k and prediction time interval $[t_{j-1}, t_j)$ is computed based on (6.3). Subsequently, collision checks are performed in line 4, i.e., it is checked whether there exists a nonempty intersection for any of the N time intervals. If no intersection is detected, the motion plan of the ego vehicle is formally verified as safe with respect to the considered traffic participant. Otherwise, the ego vehicle must repair the desired trajectory [284, 285] or perform a safe backup maneuver to ensure safety, as shown in Fig. 6.3.

6.4 Anytime Safety Verification

In this section, we present our anytime safety verification approach. In Subsection 6.4.1, we propose our algorithm that aims to quickly verify that the desired motion plan of the ego vehicle is collision-free. While previous works provide a formal concept, none of these algorithms are anytime capable, i.e., the algorithm can be interrupted at any time after completing a short start-up phase and the quality of the results improves until the available computation time is elapsed [286]. To design an efficient anytime safety verification algorithm, we

- reuse the vector of occupancy sets $\mathcal{O}_{k-1}$ obtained at t_{k-1} (Subsection 6.4.2);

- sort the vector of models $\mathbf{M}$ based on their computational complexity and perform collision checks immediately after a new occupancy set has been computed (Subsection 6.4.3); and
- refine the predicted occupancy sets $\mathcal{O}_k^{(j)}$ for as long as computation time allows (Subsection 6.4.4).

6.4.1 Algorithm

Our anytime safety verification procedure is presented in Alg. 6.2. It has the same inputs and outputs as Alg. 6.1 with the exception that we use the occupancy vector of the other traffic participant of the previous time step $\mathcal{O}_{k-1}$ for $k \in \mathbb{N}_{>0}$ as an additional input. Subsequently, we describe the main steps of Alg. 6.2 in more detail.

Algorithm 6.2 Anytime safety verification

```
Input: $\mathcal{T}_k, \mathcal{Q}_k, \mathbf{M}, N, \mathcal{O}_{k-1}$
Output: $a_c, \mathcal{O}_k$
 1: if updateModelParameters() then
 2:     for all $j \in \mathbb{N}_{[1,N]}$ do
 3:         $\mathcal{O}_k^{(j)} \leftarrow \mathbb{R}^2$
 4:     end for
 5: else
 6:     $\mathcal{O}_k \leftarrow$ lazyUpdate$(\mathcal{O}_{k-1})$                ▷ see Subsection 6.4.2
 7:     $\mathcal{O}_k^{(N)} \leftarrow \mathbb{R}^2$
 8: end if
 9: for all $j \in \mathbb{N}_{[1,N]}$ do                                   ▷ see Subsection 6.4.3
10:     $m^{(j)} \leftarrow 0$
11:     $c^{(j)} \leftarrow \mathcal{O}_k^{(j)} \cap \mathcal{T}_k^{(j)} \not\equiv \emptyset$
12:     while $c^{(j)} \wedge \left(m^{(j)} < n_{\mathbf{M}}\right)$ do
13:         $m^{(j)} \leftarrow m^{(j)} + 1$
14:         $\mathcal{O}_k^{(j)} \leftarrow \mathcal{O}_k^{(j)} \cap \mathcal{Q}_k^{(j)}\left(\mathbf{M}^{(m^{(j)})}\right)$
15:         $c^{(j)} \leftarrow \mathcal{O}_k^{(j)} \cap \mathcal{T}_k^{(j)} \not\equiv \emptyset$
16:     end while
17: end for
18: $a_c \leftarrow$ any$(c)$
19: for all $j \in \mathbb{N}_{[1,N]}$ do                                   ▷ see Subsection 6.4.4
20:     for all $i \in \mathbb{N}_{[m^{(j)}+1, n_{\mathbf{M}}]}$ do
21:         $\mathcal{O}_k^{(j)} \leftarrow \mathcal{O}_k^{(j)} \cap \mathcal{Q}_k^{(j)}\left(\mathbf{M}^{(i)}\right)$
22:     end for
23: end for
```

In line 1 of Alg. 6.2, we check if the updated model parameters at t_k have changed compared to those at t_{k-1} based on the newly available sensor measurements. If altered, the occupancy sets of the other traffic participant computed at t_{k-1} are based on models that are possibly no longer conformant to the real system, invalidating the formal safety guarantees. Thus, we slightly modify the function updateModelParameters used in Alg. 6.1 by introducing a return value that is Boolean "true" if the model parameters have changed or the current time t equals the initial time t_0; otherwise, it is "false". If the model parameters have changed, all N occupancy sets $\mathcal{O}_k^{(j)}$ with $j \in \mathbb{N}_{[1,N]}$ are initialized with $\mathbb{R}^2$ in line 3 of Alg. 6.2. Otherwise, we reuse the occupancy vector of the previous time step $\mathcal{O}_{k-1}$ to quickly obtain over-approximations of $\mathcal{O}_k^{(j)}$ with $j \in \mathbb{N}_{[1,N-1]}$, as described in Subsection 6.4.2 in more detail.

In lines 9 to 17, we aim to quickly verify that no collision occurs for all N prediction time intervals. The fast verification is achieved by ordering the vector of models $\mathbf{M}$ based on their computational complexity. In addition, collision checks are performed in line 15 immediately after a newly computed occupancy set has been intersected with the overall occupancy set in line 14, as described in Subsection 6.4.3 in more detail. As in Alg. 6.1, the Boolean collision vector $c \in \mathbb{B}^N$ stores the safety verification result for all N prediction time intervals. Furthermore, the scalar $m^{(j)} \in \mathbb{N}_{[1,n_{\mathbf{M}}]}$ with $j \in \mathbb{N}_{[1,N]}$ in Alg. 6.2 corresponds to the number of models required to verify safety for the prediction time interval $[t_{j-1}, t_j)$.

If more computation time is available after verifying the safety of the ego vehicle, i.e., if the safety verification takes less than Δt, the remaining models are also used to refine the occupancy sets $\mathcal{O}_k^{(j)}$ in lines 19 to 23, as described in Subsection 6.4.4 in more detail. This refinement is done to reduce the over-approximation of these occupancy sets for their potential future reuse. Finally, Alg. 6.2 returns the safety verification result $\texttt{any}(c)$ in addition to the vector of occupancy sets $\mathcal{O}_k$ at t_k.

6.4.2 Reuse of Occupancy Sets

We can quickly predict the occupancy sets of another traffic participant at t_k for $k \in \mathbb{N}_{>0}$ by reusing $\mathcal{O}_{k-1}$ obtained at t_{k-1} if the model parameters are unchanged. As a result, we only need to compute the occupancy set $\mathcal{O}_k^{(N)}$ corresponding to the last prediction time interval $[t_{N-1}, t_N)$ while using elements of $\mathcal{O}_{k-1}$ as over-approximations corresponding to all other intervals, as shown in the following proposition.

Proposition 6.4 (Reuse of Occupancy Sets): $\mathcal{Q}_k^{(j-1)}\left(\mathbf{M}^{(i)}\right) \subseteq \mathcal{Q}_{k-1}^{(j)}\left(\mathbf{M}^{(i)}\right)$ for $j \in \mathbb{N}_{[2,N]}$, $k \in \mathbb{N}_{>0}$, and any model $\mathbf{M}^{(i)}$ with $i \in \mathbb{N}_{[1,n_{\mathbf{M}}]}$. ■

Proof. The considered absolute time intervals are the same, i.e., $[t_{k+(j-1)-1}, t_{k+(j-1)})$ equals $[t_{(k-1)+j-1}, t_{(k-1)+j})$. Based on this equality, the set relation follows directly from Proposition 6.1 and Definition 6.2. □

As defined in Section 6.3, the vector of occupancy sets of another traffic participant at t_{k-1} is given by $\mathcal{O}_{k-1} = \begin{bmatrix}\mathcal{O}_{k-1}^{(1)} & \mathcal{O}_{k-1}^{(2)} & \mathcal{O}_{k-1}^{(3)} & \dots & \mathcal{O}_{k-1}^{(N-1)} & \mathcal{O}_{k-1}^{(N)}\end{bmatrix}$. To perform a circular shift of this vector, we introduce the lazy update function

$$\texttt{lazyUpdate}\left(\mathcal{O}_{k-1}\right) = \begin{bmatrix}\mathcal{O}_{k-1}^{(2)} & \mathcal{O}_{k-1}^{(3)} & \dots & \mathcal{O}_{k-1}^{(N-1)} & \mathcal{O}_{k-1}^{(N)} & \mathcal{O}_{k-1}^{(1)}\end{bmatrix},$$

which is called in line 6 of Alg. 6.2. By using $\texttt{lazyUpdate}\left(\mathcal{O}_{k-1}\right)$, we quickly obtain an over-approximative result for all prediction time intervals $[t_{j-1}, t_j)$ with $j \in \mathbb{N}_{[1,N-1]}$ at t_k based

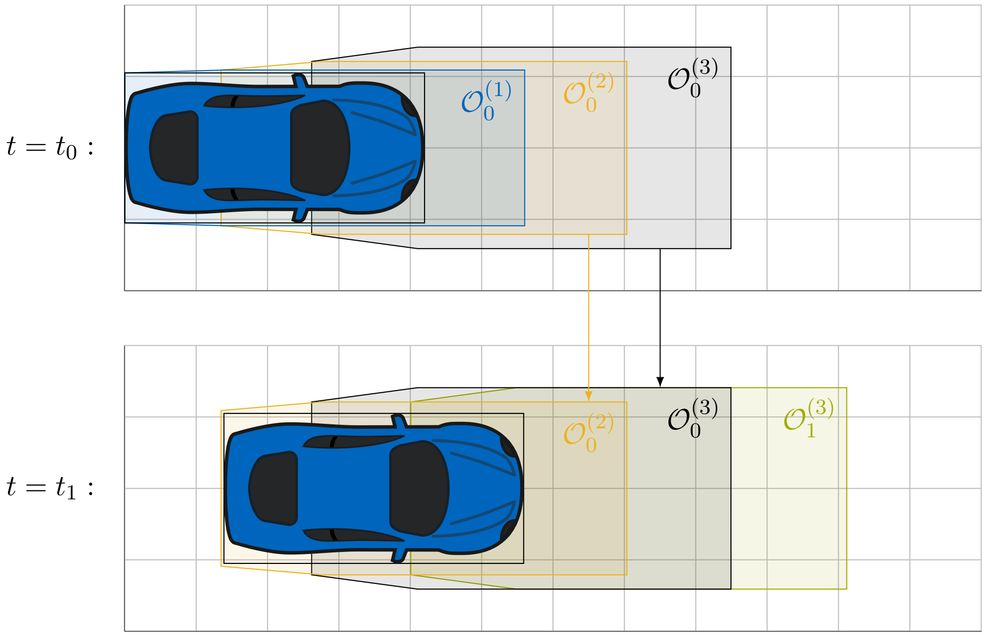


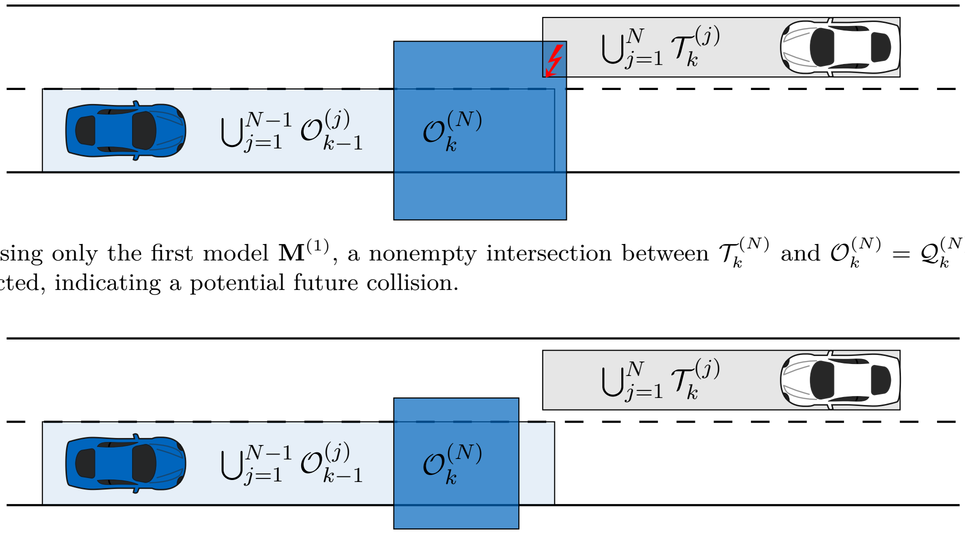
Figure 6.5: Reuse of occupancy sets. The occupancy sets $\mathcal{O}_0^{(2)}$ and $\mathcal{O}_0^{(3)}$ obtained at t_0 are reused at t_1 to over-approximate $\mathcal{O}_1^{(1)}$ and $\mathcal{O}_1^{(2)}$, respectively. Thus, only $\mathcal{O}_1^{(3)}$ must be computed at t_1 for obtaining the valid vector of occupancy sets $\mathcal{O}_1 = \begin{bmatrix} \mathcal{O}_0^{(2)} & \mathcal{O}_0^{(3)} & \mathcal{O}_1^{(3)} \end{bmatrix}$.

on Proposition 6.4. Thus, only $\mathcal{O}_k^{(N)}$ must be computed based on the newly available sensor measurements at t_k to obtain a valid vector of occupancy sets $\mathcal{O}_k$.

Example 6.5 (Reuse of Occupancy Sets): In the upper plot of Fig. 6.5, all occupancy sets at t_0 for $N = 3$ and the rightward moving vehicle are illustrated. Based on Proposition 6.4, we exploit the fact that $\mathcal{O}_1^{(1)} \subseteq \mathcal{O}_0^{(2)}$ and $\mathcal{O}_1^{(2)} \subseteq \mathcal{O}_0^{(3)}$ to quickly obtain an over-approximative result for the first two prediction time intervals at t_1, as shown in the lower plot of Fig. 6.5. Thus, only the last occupancy set $\mathcal{O}_1^{(3)}$ must be computed at t_1. ■

6.4.3 Fast Safety Verification

In lines 9 to 17 of Alg. 6.2, we aim at quickly verifying that the desired trajectory of the ego vehicle is safe. For all $j \in \mathbb{N}_{[1,N-1]}$, the occupancy set of the ego vehicle $\mathcal{T}_k^{(j)}$ is checked for a nonempty intersection with $\mathcal{O}_k^{(j)}$, which is possibly over-approximated by a reused set. If $\mathcal{O}_{k-1}^{(j+1)}$ can be reused and the trajectory of the ego vehicle is unchanged, i.e., $\mathcal{T}_k^{(j)} \subseteq \mathcal{T}_{k-1}^{(j+1)}$, the collision check in line 11 always returns Boolean "false". Thus, the check can be omitted in these cases. However, if a collision is detected in line 11 for a reused set and a changed motion plan, it is unclear whether this is an actual or spurious collision due to the reuse of previously computed occupancy sets. In this case, we verify safety for the first $N - 1$ prediction time intervals exactly as done for the last one $[t_{N-1}, t_N)$, which is described subsequently.


$\bigcup_{j=1}^{N} \mathcal{T}_k^{(j)}$
$\bigcup_{j=1}^{N-1} \mathcal{O}_{k-1}^{(j)}$
$\mathcal{O}_k^{(N)}$
$\bigcup_{j=1}^{N} \mathcal{T}_k^{(j)}$
$\bigcup_{j=1}^{N-1} \mathcal{O}_{k-1}^{(j)}$
$\mathcal{O}_k^{(N)}$

(a) By using only the first model $\mathbf{M}^{(1)}$, a nonempty intersection between $\mathcal{T}_k^{(N)}$ and $\mathcal{O}_k^{(N)} = \mathcal{Q}_k^{(N)}(\mathbf{M}^{(1)})$ is detected, indicating a potential future collision.

(b) By using $m^{(N)} \in \mathbb{N}_{[1,n_{\mathbf{M}}]}$ models, an empty intersection between $\mathcal{T}_k^{(N)}$ and $\mathcal{O}_k^{(N)} = \bigcap_{i=1}^{m^{(N)}} \mathcal{Q}_k^{(N)}\left(\mathbf{M}^{(i)}\right)$ is detected, verifying safety for the last prediction time interval $[t_{N-1}, t_N)$.

Figure 6.6: Fast safety verification. The first $N-1$ occupancy sets of the other blue traffic participant at t_k are over-approximated by the collision-free reused sets $\mathcal{O}_{k-1}^{(j)}$ with $j \in \mathbb{N}_{[1,N-1]}$ computed at t_{k-1}. Thus, only $\mathcal{O}_k^{(N)}$ must be computed at t_k.

To speed up the safety verification, we sort the vector of models $\mathbf{M}$ such that $\mathbf{M}^{(i)}$ has a lower computational complexity than $\mathbf{M}^{(i+1)}$ for all $i \in \mathbb{N}_{[1,n_{\mathbf{M}}-1]}$. As a suitable complexity measure, we use the number of floating point operations required to obtain the corresponding occupancy set. Then, we compute $\mathcal{Q}_k^{(N)}(\mathbf{M}^{(1)})$ corresponding to the simplest model $\mathbf{M}^{(1)}$ and intersect this set with the occupancy set $\mathcal{O}_k^{(N)}$ in line 14 of Alg. 6.2. Subsequently, a collision check is performed in line 15. If a collision is detected for $\mathbf{M}^{(1)}$, as illustrated in Fig. 6.6a, we compute $\mathcal{Q}_k^{(N)}(\mathbf{M}^{(2)})$ for the second model $\mathbf{M}^{(2)}$, intersect it with $\mathcal{O}_k^{(N)}$ to reduce the over-approximation based on (6.3), and perform a new collision check. This procedure is repeated until safety is eventually verified for $\mathcal{O}_k^{(N)} = \bigcap_{i=1}^{m^{(N)}} \mathcal{Q}_k^{(N)}\left(\mathbf{M}^{(i)}\right)$ with $m^{(N)} \in \mathbb{N}_{[1,n_{\mathbf{M}}]}$, as illustrated in Fig. 6.6b. Therefore, we formally verify the desired motion plan of the ego vehicle as safe using as few models as possible, starting with the simplest ones.

This fast safety verification method produces different results for the same input data depending on the available computation time. Nevertheless, our interruptible anytime Alg. 6.2 can formally verify the safety of the ego vehicle's trajectory much faster than the traditional Alg. 6.1, as shown in the subsequent numerical examples in Section 6.5. As for the traditional verification approach, we execute the safe backup maneuver if we cannot formally verify a desired trajectory in time, as shown in Fig. 6.3.

6.4.4 Occupancy Set Refinements

In lines 19 to 23 of Alg. 6.2, our anytime safety verification procedure continues computing the occupancy sets $\mathcal{Q}_k^{(j)}\left(\mathbf{M}^{(i)}\right)$ based on the more complex models $\mathbf{M}^{(i)}$ for $i \in \mathbb{N}_{[m^{(j)}+1,n_{\mathbf{M}}]}$ and

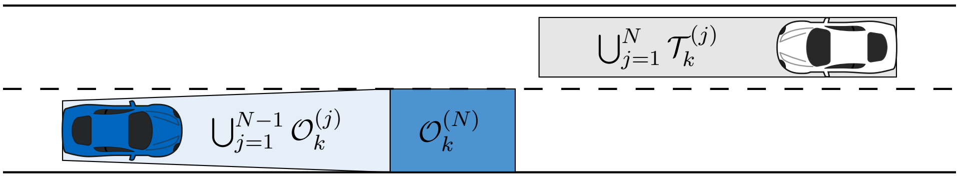

$\bigcup_{j=1}^{N} \mathcal{T}_k^{(j)}$
$\bigcup_{j=1}^{N-1} \mathcal{O}_k^{(j)}$
$\mathcal{O}_k^{(N)}$

Figure 6.7: Refined occupancy sets.

the newly available sensor data obtained at t_k, even though the safety verification result no longer changes. These computations are performed to reduce the over-approximation of the occupancy sets for their potential future reuse, i.e., at times $t_{k+\bar{k}}$ with $\bar{k} \in \mathbb{N}_{[1,N]}$. Thus, if more computation time is available, the other models are additionally used to refine the occupancy sets $\mathcal{O}_k^{(j)}$ for all $j \in \mathbb{N}_{[1,N]}$. Finally, after all occupancy sets are refined, as illustrated in Fig. 6.7, Alg. 6.2 returns the safety verification result and the vector of occupancy sets $\mathcal{O}_k$. Then, if enough computation time is available, these two outputs are identical to the ones of Alg. 6.1.

6.5 Numerical Examples

In this section, we compare the performance of both safety verification algorithms using two numerical examples from the literature [287]. Our anytime Alg. 6.2 has been integrated into the open-source MATLAB tool SPOT [280], which implements Alg. 6.1 and represents the two-dimensional occupancy sets by polytopes. Because polytopic collision detection using the MATLAB function polybool[1] is relatively slow, we recommend to speed up these computations in the future by incorporating, e.g., bounding volume hierarchies [288, 289], pre-computed collision checks [290], and existing collision detection libraries [291, 292].

To generate a long-term trajectory for the ego vehicle, as shown in Fig. 6.3, we use a standard sampling-based approach [293]. In addition, we use the following three model abstractions for all other traffic participants that are ordered by their computational complexity:

- an infinite-acceleration-based model $\mathbf{M}^{(1)}$, as introduced in Example 6.3;
- a finite-acceleration-based model $\mathbf{M}^{(2)}$ [279]; and
- a lane-following model $\mathbf{M}^{(3)}$ [279].

When choosing these models, $\mathcal{Q}_k^{(j)}(\mathbf{M}^{(2)}) \subseteq \mathcal{Q}_k^{(j)}(\mathbf{M}^{(1)})$ for any $j \in \mathbb{N}_{>0}$ and $k \in \mathbb{N}$, i.e., the second model $\mathbf{M}^{(2)}$ always produces tighter occupancy sets than $\mathbf{M}^{(1)}$. Thus, $\mathbf{M}^{(1)}$ is disregarded by SPOT to speed up the traditional safety verification computations. However, in general, $\mathcal{Q}_k^{(j)}(\mathbf{M}^{(2)}) \nsubseteq \mathcal{Q}_k^{(j)}(\mathbf{M}^{(3)})$ and $\mathcal{Q}_k^{(j)}(\mathbf{M}^{(3)}) \nsubseteq \mathcal{Q}_k^{(j)}(\mathbf{M}^{(2)})$. These typical set relations are also illustrated in Fig. 6.4.

To determine the computational speed-up potential, we terminate Alg. 6.2 as soon as the safety of the desired motion plan is verified. Because performance comparisons depend highly on the specific traffic scenario, we use scenarios provided by the motion planning benchmark suite CommonRoad [287]. This suite is a collection of composable benchmarks for motion

[1] https://mathworks.com/help/map/ref/polybool.html

planning of autonomous vehicles on roads that assigns each benchmark a unique identifier. This identifier specifies detailed information about, e.g., the ego vehicle, the cost function, the road network, and the other traffic participants. In addition to hand-crafted scenarios that provide challenging safety-critical situations, real-world recorded traffic data is also available. Because all benchmarks can be downloaded from the CommonRoad website[2], the following numerical experiments can be easily reproduced.

6.5.1 Uncontrolled Intersection

To compare the computed occupancy sets for two consecutive time steps, we use the CommonRoad benchmark PM1:MW1:DEU_Muc-3_1_T-1:2018b. Here, the ego vehicle is modeled as a point mass with maximum absolute acceleration of $11.5\,\frac{\mathrm{m}}{\mathrm{s}^2}$, the cost function is inspired by [293, Eq. 2 and Sec. 5.B], and the CommonRoad release 2018b is used. The considered traffic scenario comprises an uncontrolled intersection with three other traffic participants and specifies the ego vehicle to turn left. The initial configuration and the occupancy sets computed at t_0 by SPOT are shown in Fig. 6.2, where the sampling period is $\Delta t = 100\,\mathrm{ms}$ and the prediction horizon is $N = 17$. Thus, we predict the occupancy sets of all three safety-relevant traffic participants for the next 1.7 s.

The predicted occupancy sets computed by our interruptible anytime Alg. 6.2 at t_1 are shown in Fig. 6.8. In addition to reusing the occupancy sets obtained at t_0, it is sufficient to consider only the simplest model $\mathbf{M}^{(1)}$ for the first and second vehicles to verify safety at t_1. However, we must use all three model abstractions to formally verify the desired motion plan of the ego vehicle as safe with respect to the third traffic participant. By averaging the simulation results for t_1 over 10 runs, we obtain computational speed-ups of Alg. 6.2 compared to SPOT of 26.1, 23.8, and 4.9 for the first, second, and third traffic participant, respectively. This results in an overall speed-up of 9.6 while a single safety verification step takes 4.4 ms.

There are multiple reasons why our proposed anytime approach is not even faster. For instance, the slow polytopic collision check is performed each time after a new $\mathcal{Q}_k^{(N)}\left(\mathbf{M}^{(i)}\right)$ with $i \in \mathbb{N}_{[1,3]}$ is intersected with the overall $\mathcal{O}_k^{(N)}$ in line 14 of Alg. 6.2. In contrast to these frequent collision checks, SPOT only performs a single collision check for the final occupancy set $\mathcal{O}_k^{(N)}$. Thus, if computing the occupancy set for a more complex model has a lower complexity than performing the collision detection for a simpler model, which is true in this example for $\mathbf{M}^{(1)}$ and $\mathbf{M}^{(2)}$, it may be beneficial to skip this check to optimize, e.g., the expected overall computation time. Similarly important, some computations, e.g., obtaining the reachable lanes for $\mathbf{M}^{(3)}$, must be performed irrespective of whether the result is only used for the last prediction time interval $[t_{N-1}, t_N)$ or for all N intervals. However, the complexity of these computations will be significantly reduced in a future C++ implementation of SPOT.

6.5.2 Three-Lane Highway

The second vehicular traffic scenario we consider is given by the CommonRoad benchmark PM1:MW1:DEU_A9-2_1_T-1:2018b. It features a three-lane highway, where the ego vehicle is initially located in the middle lane and must perform a lane change to the right one, as shown

[2] commonroad.in.tum.de

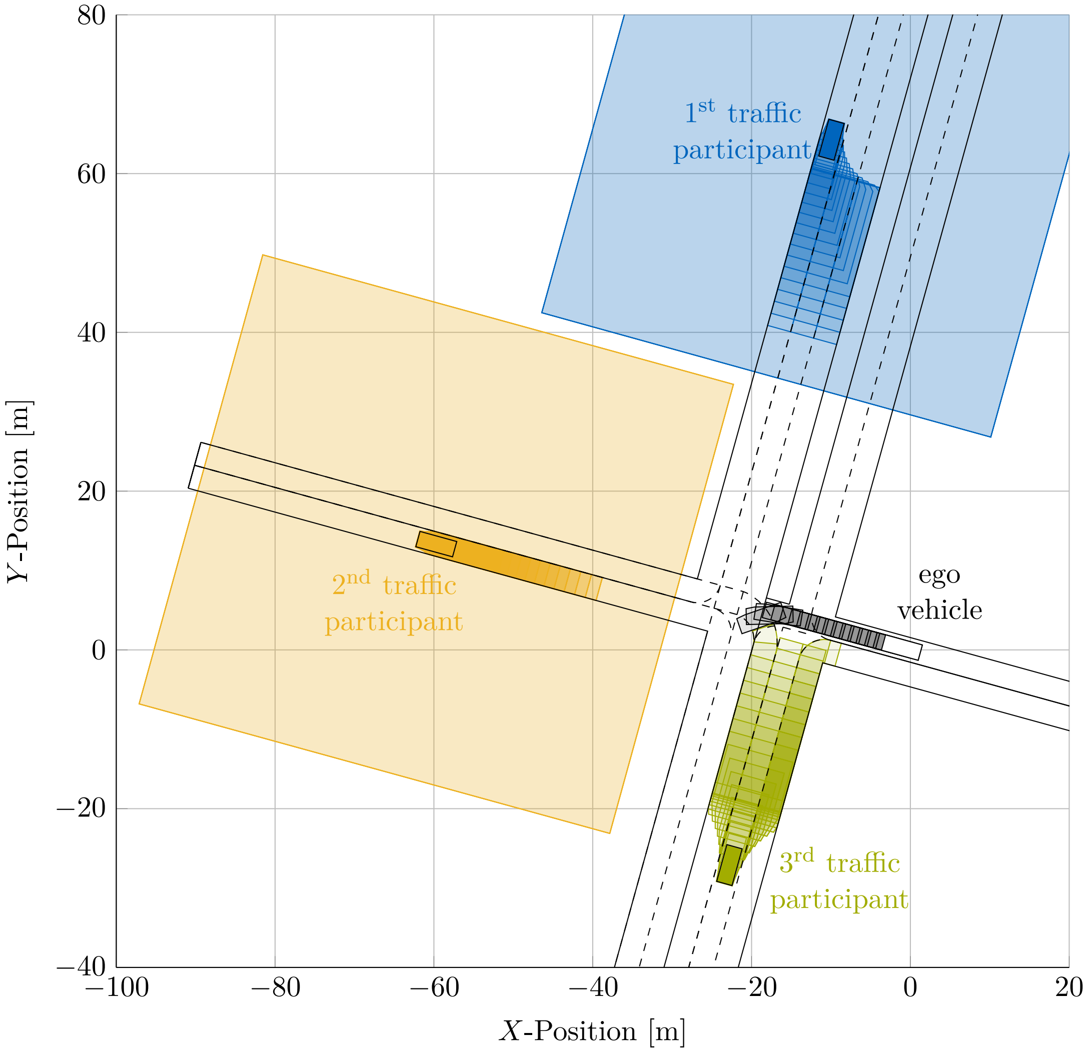


Figure 6.8: Predicted occupancy sets for uncontrolled intersection obtained by interrupting Alg. 6.2 as soon as the safety of the desired trajectory is formally verified. The traffic scenario is described by the CommonRoad benchmark PM1:MW1:DEU_Muc-3_1_T-1:2018b.

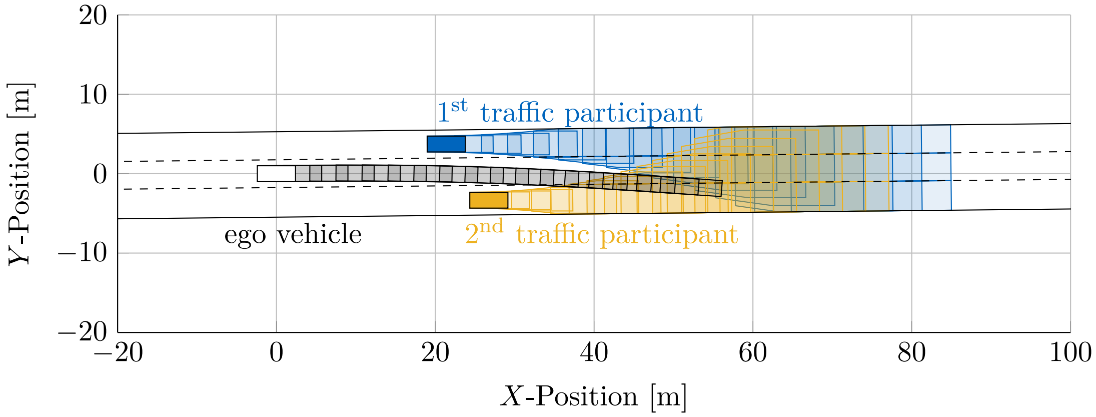


Figure 6.9: Initial occupancy sets for three-lane highway. The traffic scenario is described by the CommonRoad benchmark PM1:MW1:DEU_A9-2_1_T-1:2018b.

in Fig. 6.9. In addition, this scenario includes two other safety-relevant traffic participants. As for the previous CommonRoad benchmark in Subsection 6.5.1, the prediction horizon is $N = 17$ and the sampling period is $\Delta t = 100\,\mathrm{ms}$. Then, performing the whole lane change maneuver takes 13 time steps.

As mentioned before, SPOT implements Alg. 6.1 and disregards the simplest model $\mathbf{M}^{(1)}$. To provide a fair comparison of both safety verification algorithms, i.e., to use the same model abstractions, Alg. 6.2 also disregards $\mathbf{M}^{(1)}$ subsequently, i.e., it considers only $\mathbf{M}^{(2)}$ and $\mathbf{M}^{(3)}$. By averaging the simulation results for t_k with $k \in \mathbb{N}_{[1,13]}$ over 10 runs, we obtain computational speed-ups of Alg. 6.2 compared to SPOT of 51.0 and 57.6 for the first and second traffic participant, respectively. This results in an overall speed-up of 54.3 while a single safety verification step takes 0.5 ms. In contrast to the previous benchmark, it is unnecessary to consider the most complex model $\mathbf{M}^{(3)}$ for either of the two other traffic participants to guarantee safety, which is why a higher overall speed-up is obtained.

6.6 Summary

In this chapter, we have proposed an anytime safety verification approach, which attempts to quickly verify that the desired trajectory of the ego vehicle is collision-free. In contrast to existing methods, our approach is anytime capable, i.e., the algorithm can be interrupted at any time after completing a short start-up phase and the quality of the results improves until the available computation time is elapsed. In particular, we reuse the occupancy sets obtained at the previous time step, sort the models of other traffic participants based on their computational complexity, and refine the predicted occupancy sets for as long as computation time allows. Finally, we have demonstrated the effectiveness of our proposed anytime safety verification approach using an uncontrolled intersection and a three-lane highway traffic scenario. As shown, our anytime method achieves significant computational speed-ups for verifying the safety of the ego vehicle.

7 Conclusions and Future Work

In this chapter, we conclude this thesis. In Section 7.1, we summarize our main contributions. Finally, we suggest promising future research directions and further improvements of our proposed control approaches in Section 7.2.

7.1 Summary of Contributions

As seen by the failures of the Lewis spacecraft and the radiation therapy machine Therac-25 in Chapter 1, guaranteeing the safety of autonomous systems for an infinite time horizon is crucial when deploying these systems in safety-critical applications. To provide such safety guarantees, a wide variety of approaches that compute robust control invariant (RCI) sets already exist in the literature. However, these approaches typically suffer from exponential computational complexity with respect to the problem dimension or excessive conservativeness.

To overcome these limitations, we have presented scalable algorithms for computing nonconservative safe sets of sampled-data systems along with corresponding set-based, safety-preserving controllers. These controllers formally guarantee robust state and input constraint satisfaction at all times if the initial state of the dynamical system lies within the safe set, which is not necessarily RCI. Because safe sets are usually desired to have minimum or maximum volume, we have proposed multiple methods to synthesize such sets. Because our computations are based on scalable reachability analysis and convex optimization, the computational complexity of our safe set approaches is only polynomial with respect to the problem dimension. To evaluate the performance of these methods and validate the formal safety guarantees of their corresponding safety-preserving controllers, we have considered multiple numerical examples taken from the literature.

The efficient computation of nonconservative safe sets is beneficial not only for leveraging autonomous systems in safety-critical applications but also for enhancing other popular control methods, such as model predictive control (MPC). Thus, we have also proposed an efficient robust output feedback MPC approach that uses our safe sets as terminal sets. In particular, when iteratively solving an optimization problem on a moving horizon, the state at the end of this horizon is constrained to lie within our safe set. In addition, we have used a simple linear state observer to estimate the inaccessible state of the system based on noisy measurements obtained online. To demonstrate the effectiveness of our real-time robust output feedback MPC approach, we have used a nine-dimensional vehicle platooning system with a sampling period of 150 ms.

In addition to MPC, supervisory control is another important control area that benefits from our efficient safe set computations. Here, the goal is to guarantee the safety of the controlled system at all times while minimally modifying the desired input of an unverified high-performance controller, which is obtained, e.g., using machine learning techniques. We have achieved this goal by enforcing the state of the system to stay within our safe set at all times. Because our approach makes no assumptions about the availability of a model along with its corresponding disturbance bounds, a new measurement obtained online might invalidate the formal safety guarantees, which were based on a finite set of training data. In this case, we quickly update our safe set along with its corresponding safety-preserving controller online to restore model conformance. We have considered multiple numerical examples taken from the literature to demonstrate the usefulness and generalizability of our robust control approach.

The concept of safe sets along with corresponding safety-preserving controllers is also beneficial for formally verifying the safety of autonomous vehicles online. In particular, if the desired trajectory of the controlled autonomous vehicle intersects the predicted occupancy set of another traffic participant, the safety-preserving controller overwrites the desired control input to ensure safety. To guarantee timely, safe trajectories, we have proposed an anytime approach that provides conservative formal verification results quickly and continually refines them until the available computation time is elapsed. Thus, our algorithm can be interrupted at any time while using the available computational resources optimally. We have considered two traffic scenario benchmarks to demonstrate the effectiveness of our anytime safety verification method.

7.2 Future Research Directions

In this thesis, we have taken a step towards robustly controlling general, high-dimensional systems in real-world, safety-critical applications. Subsequently, we identify several promising directions for future work and further improvements of our proposed robust control approaches.

Computing safe sets of nonlinear or hybrid systems is a challenging task [152–154]. Thus, computing safe sets for these classes of dynamical systems could benefit from our efficient linear approaches [294, 295]. The first promising steps in this research direction have already been made [52]. To make our safe set approaches more appealing for practitioners, the involved algorithm parameters could also be tuned adaptively, as has been proposed for reachability analysis over the past few years [296, 297]. Although our algorithms are scalable, they might still be computationally too demanding to tackle high-dimensional, real-world problems because the modeling of large convex optimization problems (COPs) is too slow. For instance, it takes more than 2 h to model some COPs in Section 3.6 using the MATLAB toolbox YALMIP [66]. This issue could be addressed by exploiting the underlying problem structure and using our black-box approaches as building blocks in compositional controller synthesis [89, 146], which typically consider couplings between subsystems as disturbances. Instead of YALMIP, other convex optimization modeling frameworks can also be used, e.g., CVX [67, 68]. Moreover, optimizing the volume or an approximation thereof is often a good heuristic for obtaining useful safe sets. As these safe sets are usually embedded in other robust control approaches, it is useful to leverage other cost functions for improving the overall control performance [298].

Because safe sets play a crucial role in our robust output feedback MPC formulation, exploring the research directions suggested above also directly benefits our MPC approach. In addition, it is beneficial to extend our method to deal with event-triggered MPC. In particular, instead of solving an optimal MPC problem at every sampling time, we only solve it if the over-approximation error of a reachable set lies above a certain threshold [296]. Another interesting direction for future work is to consider more sophisticated controller and prediction structures [185], such as our set-based disturbance feedback controller in Subsection 3.3.2.

While these structures will improve the overall control performance, it must be ensured that the corresponding set-based computations still have a polynomial computational complexity with respect to the problem dimension and that they can be performed in real time.

There also exist great opportunities for further improving our supervisory safety filter approach. Currently, our method decouples the safe set computations and the set membership identification, which is performed based on a finite set of training data. Instead of this modular approach, it might be beneficial to combine both steps to improve the overall control performance [148], i.e., the construction of an explicit model could be avoided. In general, minimally invasive safety filters provide formal safety guarantees for any controller, e.g., obtained using machine learning techniques. Thus, it is interesting to further investigate the effects of this least restrictive filtering on the resulting control performance [299]. Another promising direction for future work is exploring more sophisticated online conformance update procedures instead of minimally enlarging the estimated disturbance set. Moreover, to improve the real-time capabilities of our safety filters, it is beneficial to construct low-dimensional model abstractions of high-dimensional systems [300], i.e., to use suitable model order reduction techniques.

Regarding our anytime safety verification approach, there is room for further improvement that calls for future work. Because the benefits of our approach are most apparent in complex traffic scenarios when computational resources are particularly scarce, the further use of urban traffic data might reveal the full potential of our anytime method. In addition, integrating and evaluating our approach in an actual vehicle is of great importance [17, 18]. There also exist many possibilities for further speeding up the computations, e.g., by using additional model abstractions of different complexities and by suitably merging several other traffic participants in the same proximity. Another interesting direction for future work is to use our set-based anytime method to quickly verify not only autonomous vehicles but also other safety-critical systems, such as robots, ships, and airplanes. Instead of verifying the desired trajectory of the controlled autonomous vehicle, it is interesting to use an extension of our safety filter for minimally modifying the desired motion plan. This extension could result in smoother control input trajectories and, thus, more comfortable autonomous driving experiences.

Bibliography

[1] A. T. Bahill and S. J. Henderson. Requirements development, verification, and validation exhibited in famous failures. *Systems Engineering*, 8(1):1–14, 2005. doi:10.1002/sys.20017. 1

[2] NASA Investigation Board. Lewis spacecraft mission failure - Final report, 1998. URL: https://llis.nasa.gov/llis_lib/pdf/1009461main1_0625-mr.pdf. 1

[3] S. Baase. *A Gift of Fire: Social, Legal, and Ethical Issues for Computing Technology*. Pearson, 4th edition, 2013. 1

[4] N. G. Leveson and C. S. Turner. An investigation of the Therac-25 accidents. *Computer*, 26(7):18–41, 1993. doi:10.1109/MC.1993.274940. 1

[5] G. J. Myers, T. Badgett, and C. Sandler. *The Art of Software Testing*. Wiley, 3rd edition, 2012. doi:10.1002/9781119202486. 1

[6] D. G. Altman and J. M. Bland. Statistics notes: Absence of evidence is not evidence of absence. *BMJ*, 311:485, 1995. doi:10.1136/bmj.311.7003.485. 1

[7] J. R. Burch, E. M. Clarke, K. L. McMillan, and D. L. Dill. Sequential circuit verification using symbolic model checking. In *ACM/IEEE Design Automation Conference*, pages 46–51, 1990. doi:10.1145/123186.123223. 1

[8] T. Ball, A. Podelski, and S. K. Rajamani. Boolean and Cartesian abstraction for model checking C programs. In *Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 268–283, 2001. doi:10.1007/3-540-45319-9_19. 1

[9] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Conference on Computer Aided Verification*, pages 585–591, 2011. doi:10.1007/978-3-642-22110-1_47. 1

[10] W. Bibel. *Automated Theorem Proving*. Vieweg+Teubner Verlag, 2nd edition, 1987. doi:10.1007/978-3-322-90102-6. 2

[11] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Springer, 2002. doi:10.1007/3-540-45949-9. 2

[12] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008. 2

[13] E. M. Clarke, O. Grumberg, D. Kroening, D. Peled, and H. Veith. *Model Checking: Second Edition*. MIT Press, 2nd edition, 2018. 2

[14] E. M. Clarke, W. Klieber, M. Nováček, and P. Zuliani. Model checking and the state explosion problem. In *LASER Summer School on Software Engineering*, pages 1–30, 2012. `doi:10.1007/978-3-642-35746-6_1`. 2

[15] R. E. Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, 1961. `doi:10.1515/9781400874668`. 2

[16] M. Althoff, G. Frehse, and A. Girard. Set propagation techniques for reachability analysis. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):369–395, 2021. `doi:10.1146/annurev-control-071420-081941`. 2, 15

[17] M. Althoff and J. M. Dolan. Online verification of automated road vehicles using reachability analysis. *IEEE Transactions on Robotics*, 30(4):903–918, 2014. `doi:10.1109/TRO.2014.2312453`. 3, 96, 115, 116, 120, 121, 133

[18] C. Pek, S. Manzinger, M. Koschi, and M. Althoff. Using online verification to prevent autonomous vehicles from causing accidents. *Nature Machine Intelligence*, 2(9):518–528, 2020. `doi:10.1038/s42256-020-0225-y`. 3, 96, 115, 133

[19] S. Kaynama, J. Maidens, M. Oishi, I. M. Mitchell, and G. A. Dumont. Computing the viability kernel using maximal reachable sets. In *Conference on Hybrid Systems: Computation and Control*, pages 55–64, 2012. `doi:10.1145/2185632.2185644`. 3

[20] T. Dang, T. Dreossi, E. Fanchon, O. Maler, C. Piazza, and A. Rocca. Set-based analysis for biological modeling. In P. Liò and P. Zuliani, editors, *Automated Reasoning for Systems Biology and Medicine*, pages 157–189. Springer, 2019. `doi:10.1007/978-3-030-17297-8_6`. 3

[21] Y. C. Chen and A. D. Dominguez-Garcia. A method to study the effect of renewable resource variability on power system dynamics. *IEEE Transactions on Power Systems*, 27(4):1978–1989, 2012. `doi:10.1109/TPWRS.2012.2194168`. 3

[22] Y. Li, P. Zhang, and P. B. Luh. Formal analysis of networked microgrids dynamics. *IEEE Transactions on Power Systems*, 33(3):3418–3427, 2018. `doi:10.1109/TPWRS.2017.2780804`. 3

[23] S. B. Liu, H. Roehm, C. Heinzemann, I. Lütkebohle, J. Oehlerking, and M. Althoff. Provably safe motion of mobile robots in human environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1351–1357, 2017. `doi:10.1109/IROS.2017.8202313`. 3

[24] M. Althoff, A. Giusti, S. B. Liu, and A. Pereira. Effortless creation of safe robots from modules through self-programming and self-verification. *Science Robotics*, 4(31):eaaw1924, 2019. `doi:10.1126/scirobotics.aaw1924`. 3, 116

[25] S. Bogomolov, M. Forets, G. Frehse, F. Viry, A. Podelski, and C. Schilling. Reach set approximation through decomposition with low-dimensional sets and high-dimensional matrices. In *Conference on Hybrid Systems: Computation and Control*, pages 41–50, 2018. `doi:10.1145/3178126.3178128`. 3, 14, 28

[26] S. Bak, H.-D. Tran, and T. T. Johnson. Numerical verification of affine systems with up to a billion dimensions. In *Conference on Hybrid Systems: Computation and Control*, pages 23–32, 2019. doi:10.1145/3302504.3311792. 3

[27] M. Althoff. Reachability analysis of large linear systems with uncertain inputs in the Krylov subspace. *IEEE Transactions on Automatic Control*, 65(2):477–492, 2020. doi:10.1109/TAC.2019.2906432. 3, 18

[28] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009. doi:10.1007/978-1-4419-0224-5. 3

[29] C. Belta, B. Yordanov, and E. Aydin Gol. *Formal Methods for Discrete-Time Dynamical Systems*. Studies in Systems, Decision and Control. Springer, 2017. doi:10.1007/978-3-319-50763-7. 3

[30] S. L. Smith, J. Tůmová, C. Belta, and D. Rus. Optimal path planning for surveillance with temporal-logic constraints. *The International Journal of Robotics Research*, 30(14):1695–1708, 2011. doi:10.1177/0278364911417911. 3

[31] B. Yordanov, J. Tůmová, I. Černá, J. Barnat, and C. Belta. Temporal logic control of discrete-time piecewise affine systems. *IEEE Transactions on Automatic Control*, 57(6):1491–1504, 2012. doi:10.1109/TAC.2011.2178328. 3

[32] G. Reissig, A. Weber, and M. Rungger. Feedback refinement relations for the synthesis of symbolic controllers. *IEEE Transactions on Automatic Control*, 62(4):1781–1796, 2017. doi:10.1109/TAC.2016.2593947. 3

[33] C. Belta and S. Sadraddini. Formal methods for control synthesis: An optimization perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):115–140, 2019. doi:10.1146/annurev-control-053018-023717. 3

[34] M. Zamani, A. Abate, and A. Girard. Symbolic models for stochastic switched systems: A discretization and a discretization-free approach. *Automatica*, 55:183–196, 2015. doi:10.1016/j.automatica.2015.03.004. 3

[35] F. Gruber, E. S. Kim, and M. Arcak. Sparsity-aware finite abstraction. In *IEEE Conference on Decision and Control*, pages 2366–2371, 2017. doi:10.1109/CDC.2017.8263995. 3, 53

[36] D. P. Bertsekas. Infinite time reachability of state-space regions by using feedback control. *IEEE Transactions on Automatic Control*, 17(5):604–613, 1972. doi:10.1109/TAC.1972.1100085. 3, 24, 27

[37] I. Kolmanovsky and E. G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4(4):317–367, 1998. doi:10.1155/S1024123X98000866. 3, 21, 22, 28, 39, 78, 81

[38] S. V. Raković, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3):406–410, 2005. doi:10.1109/TAC.2005.843854. 3, 21, 22, 23, 28, 49, 88

[39] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Birkhäuser, 2nd edition, 2015. doi:10.1007/978-3-319-17933-9. 3, 10, 21, 24, 25, 27, 28, 78, 81

[40] M. Rungger and P. Tabuada. Computing robust controlled invariant sets of linear systems. *IEEE Transactions on Automatic Control*, 62(7):3665–3670, 2017. doi:10.1109/TAC.2017.2672859. 3, 21, 24, 27, 65, 68, 111

[41] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill, 2nd edition, 2017. 4, 27, 41, 42, 75, 96

[42] F. Borrelli, A. Bemporad, and M. Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017. doi:10.1017/9781139061759. 4, 21, 23, 24, 27, 41, 42, 75, 96

[43] S. V. Raković and W. S. Levine, editors. *Handbook of Model Predictive Control*. Control Engineering. Birkhäuser, 2019. doi:10.1007/978-3-319-77489-3. 4, 27, 41, 42, 75, 96

[44] E. Garone, S. Di Cairano, and I. Kolmanovsky. Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, 75:306–328, 2017. doi:10.1016/j.automatica.2016.08.013. 4, 96

[45] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2019. doi:10.1109/TAC.2018.2876389. 4, 96, 105

[46] F. Gruber and M. Althoff. Anytime safety verification of autonomous vehicles. In *IEEE Conference on Intelligent Transportation Systems*, pages 1708–1714, 2018. doi:10.1109/ITSC.2018.8569950. 4, 96, 115

[47] F. Gruber and M. Althoff. Scalable robust model predictive control for linear sampled-data systems. In *IEEE Conference on Decision and Control*, pages 438–444, 2019. doi:10.1109/CDC40024.2019.9029873. 4, 27, 75

[48] N. Kochdumper, F. Gruber, B. Schürmann, V. Gaßmann, M. Klischat, and M. Althoff. AROC: A toolbox for automated reachset optimal controller synthesis. In *Conference on Hybrid Systems: Computation and Control*, pages 1–6, 2021. doi:10.1145/3447928.3456703. 4, 25

[49] F. Gruber and M. Althoff. Computing safe sets of linear sampled-data systems. *IEEE Control Systems Letters*, 5(2):385–390, 2021. doi:10.1109/LCSYS.2020.3002476. 4, 27, 95

[50] F. Gruber and M. Althoff. Scalable robust output feedback MPC of linear sampled-data systems. In *IEEE Conference on Decision and Control*, pages 2563–2570, 2021. doi:10.1109/CDC45484.2021.9683384. 5, 18, 27, 75

[51] F. Gruber and M. Althoff. Scalable robust safety filter with unknown disturbance bounds. *IEEE Transactions on Automatic Control*, 68(12):7756–7770, 2023. doi:10.1109/TAC.2023.3292329. 5, 27, 95

[52] L. Schäfer, F. Gruber, and M. Althoff. Scalable computation of robust control invariant sets of nonlinear systems. *IEEE Transactions on Automatic Control*, 69(2):755–770, 2024. doi:10.1109/TAC.2023.3275305. 5, 17, 28, 51, 132

[53] R. T. Rockafellar. Lagrange multipliers and optimality. *SIAM Review*, 35(2):183–238, 1993. doi:10.1137/1035044. 7

[54] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming.* SIAM, 1994. doi:10.1137/1.9781611970791. 7, 52

[55] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004. doi:10.1017/CBO9780511804441. 7, 8, 9, 15, 25, 48, 52, 84, 88, 98

[56] D. P. Bertsekas. *Convex Optimization Algorithms.* Athena Scientific, 2015. 7, 8

[57] S. Boyd, A. Agrawal, and S. Barratt. Embedded convex optimization for control. In *IEEE Conference on Decision and Control*, 2020. URL: https://stanford.edu/~boyd/papers/cdc_20.html. 7

[58] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, 1970. 7

[59] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984. doi:10.1007/BF02579150. 8, 13, 99, 106

[60] R. J. Vanderbei. *Linear Programming: Foundations and Extensions.* Springer, 4th edition, 2014. doi:10.1007/978-1-4614-7630-6. 8, 13

[61] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms.* John Wiley & Sons, 3rd edition, 2006. 8

[62] K. C. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3 - A MATLAB software package for semidefinite programming. *Optimization Methods and Software*, 11(1-4):545–581, 1999. doi:10.1080/10556789908805762. 9

[63] J. F. Sturm. Using SeDuMi 1.02, A MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625–653, 1999. doi:10.1080/10556789908805766. 9

[64] MOSEK Aps. The MOSEK optimization toolbox for MATLAB manual. Version 9.2, 2021. URL: https://docs.mosek.com/9.2/toolbox/index.html. 9, 25

[65] Gurobi Optimization. Gurobi Optimizer Reference Manual. Version 9.1. URL: https://www.gurobi.com/documentation/9.1/refman/index.html. 9

[66] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *IEEE Symposium on Computer Aided Control Systems Design*, pages 284–289, 2004. doi:10.1109/CACSD.2004.1393890. 9, 25, 68, 132

[67] M. C. Grant and S. P. Boyd. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, volume 371, pages 95–110. Springer, 2008. doi:10.1007/978-1-84800-155-8_7. 9, 132

[68] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming. Version 2.2, 2021. URL: `http://cvxr.com/cvx/`. 9, 132

[69] A. B. Kurzhanskiĭ and I. Vályi. *Ellipsoidal Calculus for Estimation and Control.* Systems & Control: Foundations & Applications. Birkhäuser, 1997. 10

[70] G. M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics.* Springer, 1995. `doi:10.1007/978-1-4613-8431-1`. 10

[71] B. Grünbaum. *Convex Polytopes*, volume 221 of *Graduate Texts in Mathematics.* Springer, 2nd edition, 2003. `doi:10.1007/978-1-4613-0019-9`. 10

[72] W. Kühn. Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, 61:47–67, 1998. `doi:10.1007/BF02684450`. 10, 15, 18

[73] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis.* SIAM, 2009. `doi:10.1137/1.9780898717716`. 12

[74] M. Althoff, O. Stursberg, and M. Buss. Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes. *Nonlinear Analysis: Hybrid Systems*, 4(2):233–249, 2010. `doi:10.1016/j.nahs.2009.03.009`. 13, 16, 18

[75] A. Kulmburg and M. Althoff. On the co-NP-completeness of the zonotope containment problem. *European Journal of Control*, 62:84–91, 2021. `doi:10.1016/j.ejcon.2021.06.028`. 13, 16, 101

[76] M. Althoff. *Reachability analysis and its application to the safety assessment of autonomous cars.* Doctoral dissertation, Technical University of Munich, 2010. URL: `https://mediatum.ub.tum.de/doc/1287517/`. 13, 16, 18, 19, 20

[77] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari. Multi-parametric toolbox 3.0. In *European Control Conference*, pages 502–510, 2013. `doi:10.23919/ECC.2013.6669862`. 13, 14, 25, 68, 75

[78] J. Bourgain and J. Lindenstrauss. Approximating the ball by a Minkowski sum of segments with equal length. *Discrete & Computational Geometry*, 9(2):131–144, 1993. `doi:10.1007/BF02189313`. 13

[79] S. Sadraddini and R. Tedrake. Linear encodings for polytope containment problems. In *IEEE Conference on Decision and Control*, pages 4367–4372, 2019. `doi:10.1109/CDC40024.2019.9029363`. 14, 17

[80] A. A. Kurzhanskiy and P. Varaiya. Ellipsoidal toolbox (ET). In *IEEE Conference on Decision and Control*, pages 1498–1503, 2006. `doi:10.1109/CDC.2006.377036`. 14

[81] M. Althoff. An introduction to CORA 2015. In *Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 120–151, 2015. `doi:10.29007/zbkv`. 14, 21, 25

[82] F. Le Gall. Powers of tensors and fast matrix multiplication. In *International Symposium on Symbolic and Algebraic Computation*, pages 296–303, 2014. `doi:10.1145/2608628.2608664`. 14

[83] H. R. Tiwary. On the hardness of computing intersection, union and Minkowski sum of polytopes. *Discrete & Computational Geometry*, 40(3):469–479, 2008. `doi:10.1007/s00454-008-9097-3`. 14, 25, 28

[84] A.-K. Kopetzki, B. Schürmann, and M. Althoff. Methods for order reduction of zonotopes. In *IEEE Conference on Decision and Control*, pages 5626–5633, 2017. `doi:10.1109/CDC.2017.8264508`. 15, 16, 41

[85] X. Yang and J. K. Scott. A comparison of zonotope order reduction techniques. *Automatica*, 95:378–384, 2018. `doi:10.1016/j.automatica.2018.06.006`. 15, 41

[86] L. J. Guibas, A. Nguyen, and L. Zhang. Zonotopes as bounding volumes. *ACM-SIAM Symposium on Discrete Algorithms*, pages 803–812, 2003. 16, 18

[87] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Workshop on Hybrid Systems: Computation and Control*, pages 291–305. Springer, 2005. `doi:10.1007/978-3-540-31954-2_19`. 16, 18

[88] B. Schürmann, R. Vignali, M. Prandini, and M. Althoff. Set-based control for disturbed piecewise affine systems with state and actuation constraints. *Nonlinear Analysis: Hybrid Systems*, 36:100826, 2020. `doi:10.1016/j.nahs.2019.100826`. 16

[89] K. Ghasemi, S. Sadraddini, and C. Belta. Compositional synthesis via a convex parameterization of assume-guarantee contracts. In *Conference on Hybrid Systems: Computation and Control*, pages 1–10, 2020. `doi:10.1145/3365365.3382212`. 17, 132

[90] C. Combastel. A state bounding observer based on zonotopes. In *European Control Conference*, pages 2589–2594, 2003. `doi:10.23919/ECC.2003.7085991`. 18

[91] T. Alamo, J. M. Bravo, and E. F. Camacho. Guaranteed state estimation by zonotopes. *Automatica*, 41(6):1035–1043, 2005. `doi:10.1016/j.automatica.2004.12.008`. 18, 78

[92] V. T. H. Le, C. Stoica, T. Alamo, E. F. Camacho, and D. Dumur. Zonotopic guaranteed state estimation for uncertain systems. *Automatica*, 49(11):3418–3424, 2013. 18

[93] C. Combastel. Zonotopes and Kalman observers: Gain optimality under distinct uncertainty paradigms and robust convergence. *Automatica*, 55:265–273, 2015. `doi:10.1016/j.automatica.2015.03.008`. 18, 98

[94] A. Girard and C. Le Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Workshop on Hybrid Systems: Computation and Control*, pages 215–228, 2008. `doi:10.1007/978-3-540-78929-1_16`. 18, 21

[95] J. M. Bravo, T. Alamo, and E. F. Camacho. Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica*, 42(10):1745–1751, 2006. `doi:10.1016/j.automatica.2006.05.003`. 18, 82, 85

[96] B. Schürmann, N. Kochdumper, and M. Althoff. Reachset model predictive control for disturbed nonlinear systems. In *IEEE Conference on Decision and Control*, pages 3463–3470, 2018. `doi:10.1109/CDC.2018.8619781`. 18, 76

[97] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In *Workshop on Hybrid Systems: Computation and Control*, pages 73–88, 2000. `doi:10.1007/3-540-46430-1_10`. 18, 77

[98] A. A. Kurzhanskiy and P. Varaiya. Ellipsoidal techniques for reachability analysis of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 52(1):26–38, 2007. `doi:10.1109/TAC.2006.887900`. 18, 77

[99] E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise-linear dynamical systems. In *Workshop on Hybrid Systems: Computation and Control*, pages 20–31, 2000. `doi:10.1007/3-540-46430-1_6`. 18

[100] A. Chutinan and B. H. Krogh. Computational techniques for hybrid system verification. *IEEE Transactions on Automatic Control*, 48(1):64–75, 2003. `doi:10.1109/TAC.2002.806655`. 18

[101] F. H. Clarke, Y. S. Ledyaev, E. D. Sontag, and A. I. Subbotin. Asymptotic controllability implies feedback stabilization. *IEEE Transactions on Automatic Control*, 42(10):1394–1407, 1997. `doi:10.1109/9.633828`. 18

[102] S. V. Raković, F. A. C. C. Fontes, and I. V. Kolmanovsky. Reachability and invariance for linear sampled-data systems. In *IFAC World Congress*, pages 3057–3062, 2017. `doi:10.1016/j.ifacol.2017.08.675`. 18, 22, 24, 28, 39, 41, 46, 49

[103] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000. 18, 22

[104] G. Lafferriere, G. J. Pappas, and S. Yovine. Symbolic reachability computation for families of linear vector fields. *Journal of Symbolic Computation*, 32(3):231–253, 2001. `doi:10.1006/JSCO.2001.0472`. 19

[105] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003. `doi:10.1137/S00361445024180`. 19

[106] J. Alman and V. V. Williams. A refined laser method and faster matrix multiplication. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 522–539, 2021. `doi:10.1137/1.9781611976465.32`. 20

[107] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok. C2E2: A verification tool for stateflow models. In *Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 68–82, 2015. `doi:10.1007/978-3-662-46681-0_5`. 21

[108] S. Schupp, E. Ábrahám, I. B. Makhlouf, and S. Kowalewski. HyPro: A C++ library of state set representations for hybrid systems reachability analysis. In *NASA Formal Methods Symposium*, pages 288–294, 2017. `doi:10.1007/978-3-319-57288-8_20`. 21

[109] S. Bak and P. S. Duggirala. HyLAA: A tool for computing simulation-equivalent reachability for linear systems. In *Conference on Hybrid Systems: Computation and Control*, pages 173–178, 2017. `doi:10.1145/3049797.3049808`. 21

[110] S. Bogomolov, M. Forets, G. Frehse, K. Potomkin, and C. Schilling. JuliaReach: A toolbox for set-based reachability. In *Conference on Hybrid Systems: Computation and Control*, pages 39–44, 2019. doi:10.1145/3302504.3311804. 21

[111] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid systems. In *Conference on Computer Aided Verification*, pages 379–395, 2011. doi:10.1007/978-3-642-22110-1_30. 21

[112] R. Ray, A. Gurung, B. Das, E. Bartocci, S. Bogomolov, and R. Grosu. XSpeed: Accelerating reachability analysis on multi-core processors. In *Haifa Verification Conference*, pages 3–18, 2015. doi:10.1007/978-3-319-26287-1_1. 21

[113] M. Althoff, E. Ábrahám, M. Forets, G. Frehse, D. Freire, C. Schilling, S. Schupp, and M. Wetzlinger. ARCH-COMP21 category report: Continuous and hybrid systems with linear continuous dynamics. In *Workshop on Applied Verification of Continuous and Hybrid Systems*, pages 1–31, 2021. doi:10.29007/lhbw. 21

[114] M. Althoff, C. Le Guernic, and B. H. Krogh. Reachable set computation for uncertain time-varying linear systems. In *Conference on Hybrid systems: computation and control*, pages 93–102, 2011. doi:10.1145/1967701.1967717. 21

[115] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *IEEE Conference on Decision and Control*, pages 4042–4048, 2008. doi:10.1109/CDC.2008.4738704. 21

[116] M. Nagumo. Über die Lage der Integralkurven gewöhnlicher Differentialgleichungen. *Physico-Mathematical Society of Japan*, 24:551–559, 1942. doi:10.11429/ppmsj1919.24.0_551. 21

[117] P.-O. Gutman and M. Cwikel. Admissible sets and feedback control for discrete-time linear dynamical systems with bounded controls and states. *IEEE Transactions on Automatic Control*, 31(4):373–376, 1986. doi:10.1109/TAC.1986.1104270. 21

[118] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999. doi:10.1016/S0005-1098(99)00113-2. 21

[119] J.-P. Aubin, A. M. Bayen, and P. Saint-Pierre. *Viability Theory: New Directions*. Springer, 2nd edition, 2011. doi:10.1007/978-3-642-16684-6. 21, 27

[120] C.-J. Ong and E. G. Gilbert. The minimal disturbance invariant set: Outer approximations via its partial sums. *Automatica*, 42(9):1563–1568, 2006. doi:10.1016/j.automatica.2006.04.019. 22, 28, 88

[121] S. V. Raković and K. I. Kouramas. The minimal robust positively invariant set for linear discrete time systems: Approximation methods and control applications. In *IEEE Conference on Decision and Control*, pages 4562–4567, 2006. doi:10.1109/CDC.2006.377500. 22, 28, 88

[122] P. Trodden. A one-step approach to computing a polytopic robust positively invariant set. *IEEE Transactions on Automatic Control*, 61(12):4100–4105, 2016. `doi:10.1109/TAC.2016.2541300`. 22, 28, 88

[123] S. Tarbouriech and C. Burgat. Positively invariant sets for constrained continuous-time systems with cone properties. *IEEE Transactions on Automatic Control*, 39(2):401–405, 1994. `doi:10.1109/9.272344`. 24

[124] S. V. Raković and K. I. Kouramas. Invariant approximations of the minimal robust positively invariant set via finite time Aumann integrals. In *IEEE Conference on Decision and Control*, pages 194–199, 2007. `doi:10.1109/CDC.2007.4434165`. 24

[125] M. Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, 2016. `doi:10.1038/533452a`. 25

[126] P. Koopman and M. Wagner. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1):90–96, 2017. `doi:10.1109/MITS.2016.2583491`. 27, 115

[127] S. C. Ho, R. D. Hibberd, and B. L. Davies. Robot assisted knee surgery. *IEEE Engineering in Medicine and Biology Magazine*, 14(3):292–300, 1995. `doi:10.1109/51.391774`. 27

[128] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. `doi:10.1016/S0005-1098(99)00214-9`. 27, 75

[129] I. M. Mitchell, J. Yeh, F. J. Laine, and C. J. Tomlin. Ensuring safety for sampled data systems: An efficient algorithm for filtering potentially unsafe input signals. In *IEEE Conference on Decision and Control*, pages 7431–7438, 2016. `doi:10.1109/CDC.2016.7799417`. 27, 47, 77, 96

[130] K. P. Wabersich and M. N. Zeilinger. Linear model predictive safety certification for learning-based control. In *IEEE Conference on Decision and Control*, pages 7130–7135, 2018. `doi:10.1109/CDC.2018.8619829`. 27, 96, 103, 105, 106, 107

[131] P. Cardaliaguet. A differential game with two players and one target. *SIAM Journal on Control and Optimization*, 34(4):1441–1460, 1996. `doi:10.1137/S036301299427223X`. 27

[132] C. Liu and I. M. Jaimoukha. The computation of full-complexity polytopic robust control invariant sets. In *IEEE Conference on Decision and Control*, pages 6233–6238, 2015. `doi:10.1109/CDC.2015.7403200`. 28, 52, 58, 63, 64, 65, 68

[133] F. Tahir and I. M. Jaimoukha. Low-complexity polytopic invariant sets for linear systems subject to norm-bounded uncertainty. *IEEE Transactions on Automatic Control*, 60(5):1416–1421, 2015. `doi:10.1109/TAC.2014.2352692`. 28

[134] A. Gupta and P. Falcone. Full-complexity characterization of control-invariant domains for systems with uncertain parameter dependence. *IEEE Control Systems Letters*, 3(1):19–24, 2019. `doi:10.1109/LCSYS.2018.2849714`. 28, 52, 58, 63, 64, 65, 68

[135] A. Gupta, H. Köroğlu, and P. Falcone. Computation of robust control invariant sets with predefined complexity for uncertain systems. *International Journal of Robust and Nonlinear Control*, 31(5):1674–1688, 2021. `doi:10.1002/rnc.5378`. 28

[136] A. Wintenberg and N. Ozay. Implicit invariant sets for high-dimensional switched affine systems. In *IEEE Conference on Decision and Control*, pages 3291–3297, 2020. `doi:10.1109/CDC42340.2020.9303986`. 28, 39

[137] T. Anevlavis, Z. Liu, N. Ozay, and P. Tabuada. Controlled invariant sets: Implicit closed-form representations and applications, 2021. `arXiv:2107.08566`. 28, 111

[138] S. V. Raković and M. Barić. Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks. *IEEE Transactions on Automatic Control*, 55(7):1599–1614, 2010. `doi:10.1109/TAC.2010.2042341`. 28, 43

[139] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, 1994. `doi:10.1137/1.9781611970777`. 28

[140] A. Poznyak, A. Polyakov, and V. Azhmyakov. *Attractive Ellipsoids in Robust Control*. Systems & Control: Foundations & Applications. Birkhäuser, 2014. `doi:10.1007/978-3-319-09210-2`. 28

[141] S. Yu, Y. Zhou, T. Qu, F. Xu, and Y. Ma. Control invariant sets of linear systems with bounded disturbances. *International Journal of Control, Automation and Systems*, 16(2):622–629, 2018. `doi:10.1007/s12555-016-0745-8`. 28

[142] S. Kaynama, I. M. Mitchell, M. Oishi, and G. A. Dumont. Scalable safety-preserving robust control synthesis for continuous-time linear systems. *IEEE Transactions on Automatic Control*, 60(11):3065–3070, 2015. `doi:10.1109/TAC.2015.2411872`. 28, 77, 91, 105, 111

[143] I. M. Mitchell, J. Budzis, and A. Bolyachevets. Invariant, viability and discriminating kernel under-approximation via zonotope scaling, 2019. `arXiv:1901.01006`. 28, 30, 44, 48, 53, 99, 105, 110, 111, 113

[144] D. Althoff, M. Althoff, and S. Scherer. Online safety verification of trajectories for unmanned flight with offline computed robust invariant sets. In *IEEE/RSJ Conference on Intelligent Robots and Systems*, pages 3470–3477, 2015. `doi:10.1109/IROS.2015.7353861`. 28

[145] S. V. Raković, E. C. Kerrigan, D. Q. Mayne, and K. I. Kouramas. Optimized robust control invariance for linear discrete-time systems: Theoretical foundations. *Automatica*, 43(5):831–841, 2007. `doi:10.1016/j.automatica.2006.11.006`. 28, 43

[146] S. Riverso, M. Farina, and G. Ferrari-Trecate. Plug-and-play decentralized model predictive control for linear systems. *IEEE Transactions on Automatic Control*, 58(10):2608–2614, 2013. `doi:10.1109/TAC.2013.2254641`. 28, 77, 132

[147] K. Ghasemi, S. Sadraddini, and C. Belta. Compositional synthesis of decentralized robust set-invariance controllers for large-scale linear systems. In *IEEE Conference on Decision and Control*, pages 2054–2059, 2019. doi:10.1109/CDC40024.2019.9028887. 28, 42, 43

[148] Y. Chen and N. Ozay. Data-driven computation of robust control invariant sets with concurrent model selection. *IEEE Transactions on Control Systems Technology*, 30(2):495–506, 2022. doi:10.1109/TCST.2021.3069759. 28, 96, 133

[149] M. Fiacchini and M. Alamir. Computing control invariant sets in high dimension is easy, 2018. arXiv:1810.10372. 28

[150] S. Munir, M. Hovd, and S. Olaru. Low complexity constrained control using higher degree Lyapunov functions. *Automatica*, 98:215–222, 2018. doi:10.1016/j.automatica.2018.09.030. 28

[151] B. Legat, S. V. Raković, and R. M. Jungers. Piecewise semi-ellipsoidal control invariant sets. *IEEE Control Systems Letters*, 5(3):755–760, 2021. doi:10.1109/LCSYS.2020.3005326. 28

[152] M. Fiacchini, T. Alamo, and E. F. Camacho. On the computation of convex robust control invariant sets for nonlinear systems. *Automatica*, 46(8):1334–1338, 2010. doi:10.1016/j.automatica.2010.05.007. 28, 132

[153] M. A. Ben Sassi and A. Girard. Controller synthesis for robust invariance of polynomial dynamical systems using linear programming. *Systems & Control Letters*, 61(4):506–512, 2012. doi:10.1016/j.sysconle.2012.01.004. 28, 132

[154] S. Yu, C. Maier, H. Chen, and F. Allgöwer. Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems. *Systems & Control Letters*, 62(2):194–200, 2013. doi:10.1016/j.sysconle.2012.11.004. 28, 132

[155] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. Wiley, 1972. 28, 30, 42, 53, 68, 75, 79, 91, 108, 111, 113

[156] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4):523–533, 2006. doi:10.1016/j.automatica.2005.08.023. 28, 30, 44

[157] J. Skaf and S. P. Boyd. Design of affine controllers via convex optimization. *IEEE Transactions on Automatic Control*, 55(11):2476–2487, 2010. doi:10.1109/TAC.2010.2046053. 28

[158] J. Anderson, J. C. Doyle, S. H. Low, and N. Matni. System level synthesis. *Annual Reviews in Control*, 47:364–393, 2019. doi:10.1016/j.arcontrol.2019.03.006. 28

[159] D. Youla, H. Jabr, and J. Bongiorno. Modern Wiener-Hopf design of optimal controllers–Part II: The multivariable case. *IEEE Transactions on Automatic Control*, 21(3):319–338, 1976. doi:10.1109/TAC.1976.1101223. 28

[160] D. Mayne. Robust and stochastic model predictive control: Are we going in the right direction? *Annual Reviews in Control*, 41:184–192, 2016. doi:10.1016/j.arcontrol.2016.04.006. 41, 42, 75, 96

[161] D. E. Knuth. *The Art of Computer Programming, Volumes 1-4A*. Addison Wesley, 3rd edition, 2011. 47

[162] E. Gover and N. Krikorian. Determinants and the volumes of parallelotopes and zonotopes. *Linear Algebra and its Applications*, 433(1):28–40, 2010. doi:10.1016/j.laa.2010.01.031. 47, 52, 98

[163] L. Vandenberghe, S. Boyd, and S.-P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, 1998. doi:10.1137/S0895479896303430. 48, 52, 98

[164] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996. doi:10.1137/1038003. 52

[165] V. Gaßmann and M. Althoff. Scalable zonotope-ellipsoid conversions using the Euclidean zonotope norm. In *American Control Conference*, pages 4715–4721, 2020. doi:10.23919/ACC45564.2020.9147938. 52

[166] F. Blanchini. Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions. *IEEE Transactions on Automatic Control*, 39(2):428–433, 1994. doi:10.1109/9.272351. 58

[167] A. Gupta. *Control of constrained dynamical systems with performance guarantees.* Doctoral dissertation, Chalmers University of Technology, 2021. doi:10.13140/RG.2.2.29369.29286. 63

[168] I. Ben Makhlouf and S. Kowalewski. Networked cooperative platoon of vehicles for testing methods and verification tools. In *Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 37–42, 2014. doi:10.29007/zvkb. 64, 65

[169] A. Alam, A. Gattami, and K. H. Johansson. An experimental study on the fuel reduction potential of heavy duty vehicle platooning. In *IEEE Conference on Intelligent Transportation Systems*, pages 306–311, 2010. doi:10.1109/ITSC.2010.5625054. 64

[170] D. Elliott, W. Keen, and L. Miao. Recent advances in connected and automated vehicles. *Journal of Traffic and Transportation Engineering*, 6(2):109–131, 2019. doi:10.1016/j.jtte.2018.09.005. 64, 119

[171] Z. Wang, Y. Bian, S. E. Shladover, G. Wu, S. E. Li, and M. J. Barth. A survey on cooperative longitudinal motion control of multiple connected and automated vehicles. *IEEE Intelligent Transportation Systems Magazine*, 12(1):4–24, 2020. doi:10.1109/MITS.2019.2953562. 64, 119

[172] C. Conte, N. R. Voellmy, M. N. Zeilinger, M. Morari, and C. N. Jones. Distributed synthesis and control of constrained linear systems. In *American Control Conference*, pages 6017–6022, 2012. doi:10.1109/ACC.2012.6314654. 68

[173] P. Nilsson and N. Ozay. Synthesis of separable controlled invariant sets for modular local control design. In *American Control Conference*, pages 5656–5663, 2016. doi:10.1109/ACC.2016.7526557. 68

[174] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003. doi:10.1016/S0967-0661(02)00186-7. 75

[175] M. G. Forbes, R. S. Patwardhan, H. Hamadah, and R. B. Gopaluni. Model predictive control in industry: Challenges and opportunities. *IFAC Symposium on Advanced Control of Chemical Processes*, 48(8):531–538, 2015. doi:10.1016/j.ifacol.2015.09.022. 75

[176] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky. The development of model predictive control in automotive industry: A survey. In *IEEE Conference on Control Applications*, pages 295–302, 2012. doi:10.1109/CCA.2012.6402735. 75

[177] P. Soru, G. De Nicolao, C. Toffanin, C. Dalla Man, C. Cobelli, and L. Magni. MPC based artificial pancreas: Strategies for individualization and meal compensation. *Annual Reviews in Control*, 36(1):118–128, 2012. doi:10.1016/j.arcontrol.2012.03.009. 75

[178] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena. Model predictive control for power converters and drives: Advances and trends. *IEEE Transactions on Industrial Electronics*, 64(2):935–947, 2017. doi:10.1109/TIE.2016.2625238. 75

[179] A. Afram and F. Janabi-Sharifi. Theory and applications of HVAC control systems - A review of model predictive control (MPC). *Building and Environment*, 72:343–355, 2014. doi:10.1016/j.buildenv.2013.11.016. 75

[180] J. A. Primbs. Portfolio optimization applications of stochastic receding horizon control. In *American Control Conference*, pages 1811–1816, 2007. doi:10.1109/ACC.2007.4282251. 75

[181] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014. doi:10.1016/j.automatica.2014.10.128. 75, 96

[182] R. Oberdieck, N. A. Diangelakis, and E. N. Pistikopoulos. Explicit model predictive control: A connected-graph approach. *Automatica*, 76:103–112, 2017. doi:10.1016/j.automatica.2016.10.005. 75

[183] M. Kvasnica, P. Bakaráč, and M. Klaučo. Complexity reduction in explicit MPC: A reachability approach. *Systems & Control Letters*, 124:19–26, 2019. doi:10.1016/j.sysconle.2018.12.002. 75

[184] A. Alessio and A. Bemporad. A survey on explicit model predictive control. In L. Magni, D. M. Raimondo, and F. Allgöwer, editors, *Nonlinear Model Predictive Control.*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 345–369. Springer, 2009. doi:10.1007/978-3-642-01094-1_29. 75

[185] S. V. Raković. Invention of prediction structures and categorization of robust MPC syntheses. In *IFAC Nonlinear Model Predictive Control Conference*, pages 245–273, 2012. doi:10.3182/20120823-5-NL-3013.00038. 75, 132

[186] G. C. Goodwin, H. Kong, G. Mirzaeva, and M. M. Seron. Robust model predictive control: Reflections and opportunities. *Journal of Control and Decision*, 1(2):115–148, 2014. doi:10.1080/23307706.2014.913837. 75

[187] M. B. Saltık, L. Özkan, J. H. A. Ludlage, S. Weiland, and P. M. J. Van den Hof. An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects. *Journal of Process Control*, 61:77–102, 2018. doi:10.1016/j.jprocont.2017.10.006. 75

[188] H. Witsenhausen. A minimax control problem for sampled linear systems. *IEEE Transactions on Automatic Control*, 13(1):5–21, 1968. doi:10.1109/TAC.1968.1098788. 76

[189] J. H. Lee and Z. Yu. Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, 33(5):763–781, 1997. doi:10.1016/S0005-1098(96)00255-5. 76

[190] P. O. M. Scokaert and D. Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998. doi:10.1109/9.704989. 76

[191] M. Lazar, D. Muñoz de la Peña, W. P. M. H. Heemels, and T. Alamo. On input-to-state stability of min-max nonlinear model predictive control. *Systems & Control Letters*, 57(1):39–48, 2008. doi:10.1016/j.sysconle.2007.06.013. 76

[192] L. Chisci, J. A. Rossiter, and G. Zappa. Systems with persistent disturbances: Predictive control with restricted constraints. *Automatica*, 37(7):1019–1028, 2001. doi:10.1016/S0005-1098(01)00051-6. 76, 87

[193] D. Q. Mayne, M. M. Seron, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005. doi:10.1016/j.automatica.2004.08.019. 76, 77

[194] A. Richards and J. How. Robust stable model predictive control with constraint tightening. In *American Control Conference*, pages 1557–1562, 2006. doi:10.1109/ACC.2006.1656440. 76, 87

[195] S. V. Raković, B. Kouvaritakis, M. Cannon, C. Panos, and R. Findeisen. Parameterized tube model predictive control. *IEEE Transactions on Automatic Control*, 57(11):2746–2761, 2012. doi:10.1109/TAC.2012.2191174. 76

[196] S. V. Raković, W. S. Levine, and B. Açikmese. Elastic tube model predictive control. In *American Control Conference*, pages 3594–3599, 2016. doi:10.1109/ACC.2016.7525471. 76

[197] M. N. Zeilinger, D. M. Raimondo, A. Domahidi, M. Morari, and C. N. Jones. On real-time robust model predictive control. *Automatica*, 50(3):683–694, 2014. `doi:10.1016/j.automatica.2013.11.019`. 76

[198] R. Gonzalez, M. Fiacchini, T. Alamo, J. L. Guzman, and F. Rodríguez. Online robust tube-based MPC for time-varying systems: A practical approach. *International Journal of Control*, 84(6):1157–1170, 2011. `doi:10.1080/00207179.2011.594093`. 76

[199] D. Q. Mayne, S. V. Raković, R. Findeisen, and F. Allgöwer. Robust output feedback model predictive control of constrained linear systems. *Automatica*, 42(7):1217–1222, 2006. `doi:10.1016/j.automatica.2006.03.005`. 76

[200] D. Q. Mayne, S. V. Raković, R. Findeisen, and F. Allgöwer. Robust output feedback model predictive control of constrained linear systems: Time varying case. *Automatica*, 45(9):2082–2087, 2009. `doi:10.1016/j.automatica.2009.05.009`. 76, 78, 87

[201] V. T. H. Le, C. Stoica, D. Dumur, T. Alamo, and E. F. Camacho. Robust tube-based constrained predictive control via zonotopic set-membership estimation. In *IEEE Conference on Decision and Control and European Control Conference*, pages 4580–4585, 2011. `doi:10.1109/CDC.2011.6161131`. 76

[202] M. Kögel and R. Findeisen. Robust output feedback MPC for uncertain linear systems with reduced conservatism. In *IFAC World Congress*, pages 10685–10690, 2017. `doi:10.1016/j.ifacol.2017.08.2186`. 76, 78

[203] A. Bemporad and A. Garulli. Output-feedback predictive control of constrained linear systems via set-membership state estimation. *International Journal of Control*, 73(8):655–665, 2000. `doi:10.1080/002071700403420`. 76, 82

[204] L. Chisci and G. Zappa. Feasibility in predictive control of constrained linear systems: The output feedback case. *International Journal of Robust and Nonlinear Control*, 12(5):465–487, 2002. `doi:10.1002/rnc.658`. 76

[205] D. A. Copp and J. P. Hespanha. Nonlinear output-feedback model predictive control with moving horizon estimation. In *IEEE Conference on Decision and Control*, pages 3511–3517, 2014. `doi:10.1109/CDC.2014.7039934`. 76

[206] F. D. Brunner, M. A. Müller, and F. Allgöwer. Enhancing output-feedback MPC with set-valued moving horizon estimation. *IEEE Transactions on Automatic Control*, 63(9):2976–2986, 2018. `doi:10.1109/TAC.2018.2791899`. 76

[207] Z. Dong and D. Angeli. Homothetic tube-based robust economic MPC with integrated moving horizon estimation. *IEEE Transactions on Automatic Control*, 66(1):64–75, 2021. `doi:10.1109/TAC.2020.2973606`. 76

[208] M. Farina and R. Scattolini. Tube-based robust sampled-data MPC for linear continuous-time systems. *Automatica*, 48(7):1473–1476, 2012. `doi:10.1016/j.automatica.2012.03.026`. 77

[209] F. Blanchini, D. Casagrande, G. Giordano, and U. Viaro. Robust constrained model predictive control of fast electromechanical systems. *Journal of the Franklin Institute*, 353(9):2087–2103, 2016. `doi:10.1016/j.jfranklin.2016.03.009`. 77

[210] F. A. C. C. Fontes, S. V. Raković, and I. V. Kolmanovsky. Rigid tube model predictive control for linear sampled-data systems. In *IFAC World Congress*, pages 9840–9845, 2017. `doi:10.1016/j.ifacol.2017.08.903`. 77

[211] I. Alvarado, D. Limon, D. Muñoz de la Peña, T. Alamo, and E. F. Camacho. Enhanced ISS nominal MPC based on constraint tightening for constrained linear systems. In *UKACC International Conference on Control*, pages 1–6, 2010. `doi:10.1049/ic.2010.0258`. 77

[212] P. D. Christofides, R. Scattolini, D. Muñoz de la Peña, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers and Chemical Engineering*, 51:21–41, 2013. `doi:10.1016/j.compchemeng.2012.05.011`. 77

[213] J. M. Maestre and R. R. Negenborn. *Distributed Model Predictive Control Made Easy*. Springer, 2014. `doi:10.1007/978-94-007-7006-5`. 77

[214] M. Althoff and J. J. Rath. Comparison of guaranteed state estimators for linear time-invariant systems. *Automatica*, 130:109662, 2021. `doi:10.1016/j.automatica.2021.109662`. 78

[215] C. Hu, C. Liu, and I. M. Jaimoukha. Computation of invariant tubes for robust output feedback model predictive control. *IFAC World Congress*, 53(2):7063–7069, 2020. `doi:10.1016/j.ifacol.2020.12.455`. 78

[216] V. M. Zavala and L. T. Biegler. The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica*, 45(1):86–93, 2009. `doi:10.1016/j.automatica.2008.06.011`. 82, 100, 101

[217] M. Althoff. On computing the Minkowski difference of zonotopes, 2015. `arXiv:1512.02794`. 87

[218] W. Tang, Z. Wang, Y. Wang, T. Raissi, and Y. Shen. Interval estimation methods for discrete-time linear time-invariant systems. *IEEE Transactions on Automatic Control*, 64(11):4717–4724, 2019. `doi:10.1109/TAC.2019.2902673`. 88

[219] S. Kaynama and C. J. Tomlin. Benchmark: Flight envelope protection in autonomous quadrotors. In *Workshop on Applied Verification for Continuous and Hybrid Systems*, 2014. 91, 111

[220] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu. Safe reinforcement learning via shielding. In *AAAI Conference on Artificial Intelligence*, pages 2669–2678, 2018. 96

[221] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):269–296, 2020. `doi:10.1146/annurev-control-090419-075625`. 96

[222] M. Yousefi, K. van Heusden, N. West, I. M. Mitchell, J. M. Ansermino, and G. A. Dumont. A formalized safety system for closed-loop anesthesia with pharmacokinetic and pharmacodynamic constraints. *Control Engineering Practice*, 84:23–31, 2019. doi:10.1016/j.conengprac.2018.11.009. 96

[223] A. Colombo and D. Del Vecchio. Least restrictive supervisors for intersection collision avoidance: A scheduling approach. *IEEE Transactions on Automatic Control*, 60(6):1515–1527, 2015. doi:10.1109/TAC.2014.2381453. 96

[224] L. Sha. Using simplicity to control complexity. *IEEE Software*, 18(4):20–28, 2001. doi:10.1109/MS.2001.936213. 96

[225] B. Könighofer, M. Alshiekh, R. Bloem, L. Humphrey, R. Könighofer, U. Topcu, and C. Wang. Shield synthesis. *Formal Methods in System Design*, 51(2):332–361, 2017. doi:10.1007/s10703-017-0276-9. 96

[226] N. Aréchiga and B. H. Krogh. Using verified control envelopes for safe controller design. In *American Control Conference*, pages 2918–2923, 2014. doi:10.1109/ACC.2014.6859307. 96

[227] S. Bak, K. Manamcheri, S. Mitra, and M. Caccamo. Sandboxing controllers for cyber-physical systems. In *IEEE/ACM Conference on Cyber-Physical Systems*, pages 3–12, 2011. doi:10.1109/ICCPS.2011.25. 96

[228] S. Bak, T. T. Johnson, M. Caccamo, and L. Sha. Real-time reachability for verified simplex design. In *IEEE Real-Time Systems Symposium*, pages 138–148, 2014. doi:10.1109/RTSS.2014.21. 96

[229] J. Wolff and M. Buss. Invariance control design for nonlinear control affine systems under hard state constraints. In *IFAC Symposium on Nonlinear Control Systems*, pages 555–560, 2004. doi:10.1016/S1474-6670(17)31282-X. 96

[230] M. Kimmel and S. Hirche. Invariance control for safe human-robot interaction in dynamic environments. *IEEE Transactions on Robotics*, 33(6):1327–1342, 2017. doi:10.1109/TRO.2017.2750697. 96

[231] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017. doi:10.1109/TAC.2016.2638961. 96

[232] B. T. Lopez, J.-J. E. Slotine, and J. P. How. Robust adaptive control barrier functions: An adaptive and data-driven approach to safety. *IEEE Control Systems Letters*, 5(3):1031–1036, 2021. doi:10.1109/LCSYS.2020.3005923. 96

[233] M. Milanese, J. Norton, H. Piet-Lahanier, and É. Walter, editors. *Bounding Approaches to System Identification*. Springer, 1996. doi:10.1007/978-1-4757-9545-5. 96

[234] L. Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010. doi:10.1016/j.arcontrol.2009.12.001. 96, 98, 100, 106

[235] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff. Model conformance for cyber-physical systems: A survey. *ACM Transactions on Cyber-Physical Systems*, 3(3):1–26, 2019. doi:10.1145/3306157. 96, 97, 119

[236] J. Berberich, A. Koch, C. W. Scherer, and F. Allgöwer. Robust data-driven state-feedback design. In *American Control Conference*, pages 1532–1538, 2020. doi:10.23919/ACC45564.2020.9147320. 96, 97, 105, 107

[237] S. K. Mulagaleti, A. Bemporad, and M. Zanon. Data-driven synthesis of robust invariant sets and controllers. *IEEE Control Systems Letters*, 6:1676–1681, 2022. doi:10.1109/LCSYS.2021.3130829. 96

[238] S. Sadraddini and C. Belta. Formal guarantees in data-driven model identification and control synthesis. In *Conference on Hybrid Systems: Computation and Control*, pages 147–156, 2018. doi:10.1145/3178126.3178145. 96, 98

[239] E. Terzi, L. Fagiano, M. Farina, and R. Scattolini. Learning-based predictive control for linear systems: A unitary approach. *Automatica*, 108:108473, 2019. doi:10.1016/j.automatica.2019.06.025. 96, 98

[240] H. J. van Waarde, C. De Persis, M. K. Camlibel, and P. Tesi. Willems' fundamental lemma for state-space systems and its extension to multiple datasets. *IEEE Control Systems Letters*, 4(3):602–607, 2020. doi:10.1109/LCSYS.2020.2986991. 97

[241] S. B. Liu and M. Althoff. Reachset conformance of forward dynamic models for the formal analysis of robots. In *IEEE/RSJ Conference on Intelligent Robots and Systems*, pages 370–376, 2018. doi:10.1109/IROS.2018.8593975. 98

[242] E. Walter and H. Piet-Lahanier. Recursive robust minimax estimation for models linear in their parameters. In *IFAC Identification and System Parameter Estimation*, pages 215–220, 1992. doi:10.1016/S1474-6670(17)50636-9. 98

[243] A. Chalkis, I. Z. Emiris, and V. Fisikopoulos. Practical volume estimation of zonotopes by a new annealing schedule for cooling convex bodies. In *International Congress on Mathematical Software*, pages 212–221, 2020. doi:10.1007/978-3-030-52200-1_21. 98

[244] M. Gevers. Identification for control: From the early achievements to the revival of experiment design. *European Journal of Control*, 11(4-5):335–352, 2005. doi:10.3166/ejc.11.335-352. 100, 106

[245] B. Schürmann, M. Klischat, N. Kochdumper, and M. Althoff. Formal Safety Net Control Using Backward Reachability Analysis. *IEEE Transactions on Automatic Control*, 67(11):5698–5713, 2022. doi:10.1109/TAC.2021.3124188. 100

[246] B. Schürmann, A. El-Guindy, and M. Althoff. Closed-form expressions of convex combinations. In *American Control Conference*, pages 2795–2801, 2016. doi:10.1109/ACC.2016.7525342. 101

[247] P. Bouffard. On-board model predictive control of a quadrotor helicopter: Design, implementation, and experiments. Technical Report UCB/EECS-2012-241, EECS Department, University of California, Berkeley, 2012. 110

[248] C. Pek and M. Althoff. Efficient computation of invariably safe states for motion planning of self-driving vehicles. In *IEEE/RSJ Conference on Intelligent Robots and Systems*, pages 3523–3530, 2018. doi:10.1109/IROS.2018.8593597. 115, 116

[249] C. Pek. *Provably safe motion planning for autonomous vehicles through online verification.* Doctoral dissertation, Technical University of Munich, 2020. URL: https://mediatum.ub.tum.de/doc/1534013/. 115, 116

[250] N. Kalra and S. M. Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016. doi:10.1016/j.tra.2016.09.010. 115

[251] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, 2016. doi:10.1109/TIV.2016.2578706. 115

[252] D. González, J. Pérez, V. Milanés, and F. Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2016. doi:10.1109/TITS.2015.2498841. 115

[253] W. Schwarting, J. Alonso-Mora, and D. Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):187–210, 2018. doi:10.1146/annurev-control-060117-105157. 115

[254] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser. A review of motion planning for highway autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):1826–1848, 2020. doi:10.1109/TITS.2019.2913998. 115

[255] S. Lefèvre, D. Vasquez, and C. Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH Journal*, 1(1):1–14, 2014. 115

[256] J. Dahl, G. Rodrigues de Campos, C. Olsson, and J. Fredriksson. Collision avoidance: A literature review on threat-assessment techniques. *IEEE Transactions on Intelligent Vehicles*, 4(1):101–113, 2019. doi:10.1109/TIV.2018.2886682. 115

[257] J. C. Hayward. Near-miss determination through use of a scale of danger. *Highway Research Record*, 384:24–35, 1972. 115

[258] K. Vogel. A comparison of headway and time to collision as safety indicators. *Accident Analysis & Prevention*, 35(3):427–433, 2003. doi:10.1016/S0001-4575(02)00022-2. 115

[259] A. Tamke, T. Dang, and G. Breuel. A flexible method for criticality assessment in driver assistance systems. In *IEEE Intelligent Vehicles Symposium*, pages 697–702, 2011. doi:10.1109/IVS.2011.5940482. 115

[260] S. Noh and W.-Y. Han. Collision avoidance in on-road environment for autonomous driving. In *International Conference on Control, Automation and Systems*, pages 884–889, 2014. doi:10.1109/ICCAS.2014.6987906. 115

[261] M. Brännström, E. Coelingh, and J. Sjöberg. Model-based threat assessment for avoiding arbitrary vehicle collisions. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):658–669, 2010. doi:10.1109/TITS.2010.2048314. 115

[262] J.-H. Kim and D.-S. Kum. Threat prediction algorithm based on local path candidates and surrounding vehicle trajectory predictions for automated driving vehicles. In *IEEE Intelligent Vehicles Symposium*, pages 1220–1225, 2015. doi:10.1109/IVS.2015.7225849. 115

[263] M. Althoff and A. Mergel. Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1237–1247, 2011. doi:10.1109/TITS.2011.2157342. 115

[264] A. Broadhurst, S. Baker, and T. Kanade. Monte Carlo road safety reasoning. In *IEEE Intelligent Vehicles Symposium*, pages 319–324, 2005. doi:10.1109/IVS.2005.1505122. 115

[265] A. Eidehall and L. Petersson. Statistical threat assessment for general road scenes using Monte Carlo sampling. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):137–147, 2008. doi:10.1109/TITS.2007.909241. 115

[266] M. Althoff, O. Stursberg, and M. Buss. Model-based probabilistic collision detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):299–310, 2009. doi:10.1109/TITS.2009.2018966. 115

[267] T. Gindele, S. Brechtel, and R. Dillmann. Learning driver behavior models from traffic observations for decision making and planning. *IEEE Intelligent Transportation Systems Magazine*, 7(1):69–79, 2015. doi:10.1109/MITS.2014.2357038. 115

[268] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, and J. Wang. Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models. *IEEE Transactions on Industrial Electronics*, 65(7):5999–6008, 2018. doi:10.1109/TIE.2017.2782236. 115

[269] B. Vanholme, D. Gruyer, B. Lusetti, S. Glaser, and S. Mammar. Highly automated driving on highways based on legal safety. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):333–347, 2013. doi:10.1109/TITS.2012.2225104. 115, 116

[270] United Nations Economic Commission for Europe. Convention on road traffic, 1968. URL: https://unece.org/fileadmin/DAM/trans/conventn/Conv_road_traffic_EN.pdf. 116, 119, 122

[271] A. Rizaldi, J. Keinholz, M. Huber, J. Feldle, F. Immler, M. Althoff, E. Hilgendorf, and T. Nipkow. Formalising and monitoring traffic rules for autonomous vehicles in Isabelle/HOL. In *Conference on Integrated Formal Methods*, pages 50–66, 2017. doi:10.1007/978-3-319-66845-1_4. 116, 119, 122

[272] S. Maierhofer, A.-K. Rettinger, E. C. Mayer, and M. Althoff. Formalization of interstate traffic rules in temporal logic. In *IEEE Intelligent Vehicles Symposium*, pages 752–759, 2020. `doi:10.1109/IV47402.2020.9304549`. 116, 119, 122

[273] S. Magdici and M. Althoff. Fail-safe motion planning of autonomous vehicles. In *IEEE Conference on Intelligent Transportation Systems*, pages 452–458, 2016. `doi:10.1109/ITSC.2016.7795594`. 116

[274] S. Shalev-Shwartz, S. Shammah, and A. Shashua. On a formal model of safe and scalable self-driving cars, 2017. `arXiv:1708.06374`. 116

[275] S. Vaskov, H. Larson, S. Kousik, M. Johnson-Roberson, and R. Vasudevan. Not-at-fault driving in traffic: A reachability-based approach. In *IEEE Intelligent Transportation Systems Conference*, pages 2785–2790, 2019. `doi:10.1109/ITSC.2019.8917052`. 116

[276] T. Fraichard and H. Asama. Inevitable collision states - a step towards safer robots? *Advanced Robotics*, 18(10):1001–1024, 2004. `doi:10.1163/1568553042674662`. 116

[277] S. Bouraine, T. Fraichard, and H. Salhi. Provably safe navigation for mobile robots with limited field-of-views in dynamic environments. *Autonomous Robots*, 32(3):267–283, 2012. `doi:10.1007/s10514-011-9258-8`. 116

[278] H. Täubig, U. Frese, C. Hertzberg, C. Lüth, S. Mohr, E. Vorobev, and D. Walter. Guaranteeing functional safety: Design for provability and computer-aided verification. *Autonomous Robots*, 32(3):303–331, 2012. `doi:10.1007/s10514-011-9271-y`. 116

[279] M. Althoff and S. Magdici. Set-based prediction of traffic participants on arbitrary road networks. *IEEE Transactions on Intelligent Vehicles*, 1(2):187–202, 2016. `doi:10.1109/TIV.2016.2622920`. 116, 121, 127

[280] M. Koschi and M. Althoff. SPOT: A tool for set-based prediction of traffic participants. In *IEEE Intelligent Vehicles Symposium*, pages 1686–1693, 2017. `doi:10.1109/IVS.2017.7995951`. 116, 127

[281] M. Koschi and M. Althoff. Set-based prediction of traffic participants considering occlusions and traffic rules. *IEEE Transactions on Intelligent Vehicles*, 6(2):249–265, 2021. `doi:10.1109/TIV.2020.3017385`. 116, 119, 121, 122

[282] D. N. Godbole, V. Hagenmeyer, R. Sengupta, and D. Swaroop. Design of emergency maneuvers for automated highway system: Obstacle avoidance problem. In *IEEE Conference on Decision and Control*, pages 4774–4779, 1997. `doi:10.1109/CDC.1997.649770`. 120

[283] R. Rajamani. *Vehicle Dynamics and Control*. Mechanical Engineering Series. Springer, 2nd edition, 2012. `doi:10.1007/978-1-4614-1433-9`. 120

[284] J. Ziegler, P. Bender, T. Dang, and C. Stiller. Trajectory planning for BERTHA - a local, continuous method. In *IEEE Intelligent Vehicles Symposium*, pages 450–457, 2014. `doi:10.1109/IVS.2014.6856581`. 122

[285] Y. Lin, S. Maierhofer, and M. Althoff. Sampling-based trajectory repairing for autonomous vehicles. In *IEEE Intelligent Transportation Systems Conference*, pages 572–579, 2021. `doi:10.1109/ITSC48978.2021.9565060`. 122

[286] S. Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73–83, 1996. `doi:10.1609/aimag.v17i3.1232`. 122

[287] M. Althoff, M. Koschi, and S. Manzinger. CommonRoad: Composable benchmarks for motion planning on roads. In *IEEE Intelligent Vehicles Symposium*, pages 719–726, 2017. `doi:10.1109/IVS.2017.7995802`. 127

[288] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):21–36, 1998. `doi:10.1109/2945.675649`. 127

[289] C. Ericson. *Real-Time Collision Detection.* CRC Press, 2005. `doi:10.1201/b14581`. 127

[290] A. Rizaldi, S. Söntges, and M. Althoff. On time-memory trade-off for collision detection. In *IEEE Intelligent Vehicles Symposium*, pages 1173–1180, 2015. `doi:10.1109/IVS.2015.7225842`. 127

[291] J. Pan, S. Chitta, and D. Manocha. FCL: A general purpose library for collision and proximity queries. In *IEEE International Conference on Robotics and Automation*, pages 3859–3866, 2012. `doi:10.1109/ICRA.2012.6225337`. 127

[292] C. Pek, V. Rusinov, S. Manzinger, M. C. Uste, and M. Althoff. CommonRoad Drivability Checker: Simplifying the development and validation of motion planning algorithms. In *IEEE Intelligent Vehicles Symposium*, pages 1013–1020, 2020. `doi:10.1109/IV47402.2020.9304544`. 127

[293] M. Werling, J. Ziegler, S. Kammel, and S. Thrun. Optimal trajectory generation for dynamic street scenarios in a Frenét frame. In *IEEE International Conference on Robotics and Automation*, pages 987–993, 2010. `doi:10.1109/ROBOT.2010.5509799`. 127, 128

[294] B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. *National Academy of Sciences*, 17(5):315–318, 1931. `doi:10.1073/pnas.17.5.315`. 132

[295] A. Mauroy, Y. Susuki, and I. Mezić. *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications.* Springer, 2020. `doi:10.1007/978-3-030-35713-9`. 132

[296] M. Wetzlinger, N. Kochdumper, and M. Althoff. Adaptive parameter tuning for reachability analysis of linear systems. In *IEEE Conference on Decision and Control*, pages 5145–5152, 2020. `doi:10.1109/CDC42340.2020.9304431`. 132

[297] M. Wetzlinger, A. Kulmburg, and M. Althoff. Adaptive parameter tuning for reachability analysis of nonlinear systems. In *Conference on Hybrid Systems: Computation and Control*, pages 1–11, 2021. `doi:10.1145/3447928.3456643`. 132

[298] V. F. Sokolov. Model evaluation for robust tracking under unknown upper bounds on perturbations and measurement noise. *IEEE Transactions on Automatic Control*, 59(2):483–488, 2014. `doi:10.1109/TAC.2013.2273295`. 132

[299] S. Gros, M. Zanon, and A. Bemporad. Safe reinforcement learning via projection on a safe set: How to achieve optimality? *IFAC World Congress*, 53(2):8076–8081, 2020. `doi:10.1016/j.ifacol.2020.12.2276`. 133

[300] M. Althoff and J. M. Dolan. Reachability computation of low-order models for the safety verification of high-order road vehicle models. In *American Control Conference*, pages 3559–3566, 2012. `doi:10.1109/ACC.2012.6314777`. 133

Thanks for reading ☺