Exploiting Structure in Safety Control

by

Zexiang Liu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)
in the University of Michigan
2024

Doctoral Committee:

Associate Professor Necmiye Ozay, Chair
Professor Jessy Grizzle
Professor Stéphane Lafortune
Associate Professor Peter Seiler
Professor Paulo Tabuada

Zexiang Liu
zexiang@umich.edu
ORCID iD: 0000-0001-8020-1619

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor, Prof. Necmiye Ozay. Without her unwavering support and encouragement, this dissertation would have remained a distant goal. Over the past five years, Necmiye has been an invaluable source of knowledge, guiding me in research and enhancing my communication skills. Her extensive expertise in control theory, commitment to high research standards, and adept problem-solving skills have consistently impressed me, making her my role model in research. While my journey with Necmiye draws to a close, I am confident that the lessons from this experience will continue to benefit me throughout my life.

I would also like to extend my deepest gratitude to my esteemed committee members: Prof. Jessy Grizzle, Prof. Stéphane Lafortune, Prof. Peter Seiler, and Prof. Paulo Tabuada. Their invaluable feedback and guidance on my dissertation, along with the support they have offered over the years, have been instrumental to my academic journey. Prof. Grizzle, who served as my academic advisor before I joined the PhD program and was the instructor of my first graduate-level course, helped me develop the mathematical skills crucial for my research. Prof. Lafortune and Prof. Seiler taught me discrete-event systems and robust control theory, expanding my knowledge of control and equipping me with the tools necessary for completing this dissertation. The content of Chapters 4 and 5 stems from collaborative research projects with Prof. Tabuada and my accomplished colleague, Dr. Tzanis Anevlavis. Prof. Tabuada's support and guidance were indispensable in bringing these projects to fruition.

I must also thank the other professors and peers with whom I had the privilege of collaborating during my doctoral studies, including Dr. Antoine Aspeel, Michael Arwashan, Dr. Tzanis Anevlavis, Haldun Balim, Dr. Glen Chou, Hao Chen, Dr. Xiaofan Cui, Dr. Zhe Du, Dr. Dan Li, Dr. Liren Yang, Prof. Yulong Gao, Prof. Dimitra Panagou, Dr. Kwesi Rutledge, Prof. Eduardo Sontag, Mohamad Louai Shehab, Dr. Yunus Emre Sahin, Jack Weitze, Prof. Sze Zheng Yong, Xiong Zeng. The opportunity to collaborate with and learn from you has been a profound pleasure.

Many thanks to my other colleagues at UM, including Sunho Jang, Ruya Karagulle, Dr. Kang Liu, Daphna Raz, and Andrew Wintenberg for all the fun events and research discussions we had together. You have colored my life in Ann Arbor beyond my initial expectation. Special thanks to my friends Shiyuan Wang and Mingjian Fu. Shiyuan, an exceptional roommate, and Mingjian, my

lifelong best friend, have provided steadfast support during challenging moments in my life. I also want to thank Dr. Judy Dyer for her valuable advices and feedback on my dissertation writing. Finally, I want to thank my parents and my other family members for their enduring support and caring.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

LIST OF ACRONYMS

RCIS robust controlled invariant set

CIS controlled invariant set

MPC model predictive control

LTL linear temporal logic

DoA domain of attraction

BRS backward reachable set

RPIS robust positively invariant set

HJR Hamilton-Jacobi reachability

LQR linear quadratic regulator

CBF control barrier functions

ABSTRACT

For safety-critical systems such as autonomous vehicles, power systems, and robotics, it is important to guarantee the systems operate under given safety constraints. Numerous safety control methods have been proposed for this purpose, but many of them are developed for a wide range of systems and do not take full advantage of the structures inherent in dynamics, controllers, and disturbances. This dissertation focuses on enhancing scalability and reducing conservativeness in safety control by leveraging these structures.

The first part of the dissertation focuses on developing scalable safety controller synthesis algorithms. We begin with analyzing the convergence properties of the inside-out algorithm, a well-established method for computing inner approximations of the maximal robust controlled invariant set (RCIS). Under mild conditions, we show that the inside-out algorithm converges exponentially to the maximal RCIS for linear systems, filling an important gap in the literature. Following the analysis of the inside-out algorithm, we develop efficient methods for computing implicit RCISs for discrete-time controllable systems. By augmenting the original system with a periodic structure, our implicit RCISs are constructed in closed form, making the proposed methods more scalable than competing approaches. Leveraging the convergence analysis for the inside-out algorithm, we further prove that the proposed implicit RCIS converges exponentially to a well-defined maximal set with a tuning parameter. Finally, we investigate the safety control problem for input-delayed systems, which are very common in the real world and possess a special structure in the system dynamics. By exploiting this structure, we show that the maximal RCIS for systems with input delay is embedded in the maximal RCIS of an auxiliary system, whose dimension is independent of the delay time. Leveraging this property, we propose an efficient method for computing the maximal RCIS for input-delayed systems, which scales well with the delay time.

In the second part of the dissertation, we focus on reducing the conservativeness in safety control, by leveraging structure in disturbance. One such structure is preview on disturbance. To assess the value of preview information in safety control, we introduce a metric called safety regret that quantifies the variation of the maximal RCIS as the preview horizon changes. For discrete-time linear systems, we prove the exponential convergence of the safety regret with the preview horizon and offer numerical algorithms that estimate the convergence rate. Our analysis can provide valuable insights when it comes to selecting sensors or perception algorithms with different pre-

diction horizons. It is worth noting that synthesizing safety controllers for systems with preview is in general a challenging task. In this dissertation, we present efficient methods for computing the maximal RCIS for three classes of systems with preview, for which we can again exploit special structures in system dynamics to improve scalability. Finally, we introduce a novel safety control framework called opportunistic safety control, enabling safe operation beyond the maximal RCIS. This framework identifies worst-case disturbance models for each state and constructs control inputs robust to these models. Such disturbance model and control inputs can be computed from the maximal RCIS of an auxiliary system. We show in both simulation and drone experiments that our approach outperforms the existing safety control framework, especially when the system operates beyond the maximal RCIS with unexpected disturbance.

CHAPTER 1

Introduction

As more and more autonomous functionality is introduced in human-cyber-physical systems, such as passenger vehicles and aircraft, guaranteeing their safe and correct operation becomes a major concern. From a control perspective, given a set of unsafe states of the system, we need to synthesize controllers such that the system's closed-loop trajectory is guaranteed to avoid unsafe states indefinitely, referred to as the *safety control problem* in this dissertation. The core mathematical tool to deal with safety control problems is the so-called robust controlled invariant set (RCIS) [20, 56, 103], characterizing sets of states where a controller can maintain their invariance. In the literature, some other tools to tackle safety control problems include reference governors [41], control barrier functions (CBF) [5, 4], safety filters [117, 110, 45], Hamilton-Jacobi reachability (HJR) [19, 47, 115], and viability theory [16], but these are essentially alternative representations or alias of RCISs. Because of the crucial role RCIS fulfills in safety control, it is the focus of this dissertation. In particular, we focus on two roadblocks that prevent many safety control approaches from being applied to real-world systems, namely scalability and conservativeness.

In terms of scalability, it is well known that the computation of RCIS suffers from the curse of dimensionality [119, 47, 25, 45]. The state-of-the-art safety control methods for continuous-time systems, such as HJR-based methods, can handle systems with up to only ten states at their limit [47, 115]. The situation is even worse for discrete-time systems, where the state-of-the art methods can barely handle systems with more than five states [73, 91]. To tackle the scalability bottleneck, multiple works have explored the idea of computing implicit RCIS, a set in a higher-dimensional space whose projection yields an RCIS. By avoiding expensive operations such as projection, implicit RCIS based methods have been shown to successfully scale to systems of dimension 13 in continuous time [45], and systems of dimension 19 in discrete time [119]. The main idea of constructing implicit RCIS is to implicitly compute the backward reachable sets (BRS) of a given RCIS (which should be easy to compute but commonly conservative). In theory, there is no guarantee that the BRSs of a conservative RCIS would converge to the maximal RCIS. Thus, compared with methods based on explicit RCIS [20, 19], the main concern of using implicit RCIS is that the resulting controller may behave overly conservatively. In this dissertation, we resolve this

concern by providing a novel convergence analysis of BRSs of RCISs (Chapter 3). We rigorously prove that under mild conditions, the BRSs of an RCIS converge to the maximal RCIS exponentially fast for discrete-time linear systems, thus providing a theoretical foundation for implicit RCIS based methods, and explaining why implicit RCIS works well in practice.

Another drawback of implicit RCIS based methods is the need of a pre-computed RCIS [119, 45], which adds one extra layer of complexity to these methods. In this dissertation, we propose a novel approach of constructing implicit RCIS (Chapter 4) for controllable systems, which does not require a pre-computed RCIS. Compared with the existing methods, our approach constructs implicit RCIS in closed form, and provides weak completeness and performance guarantees, thanks to our BRS convergence analysis in Chapter 3. In particular, for a nominal system, our approach is guaranteed to converge to the maximal controlled invariant set (CIS) exponentially fast with a tuning parameter. Similar results also hold for systems with disturbance. The key idea behind our approach is to close the loop of a controllable system with a parameterized controller that generates eventually periodic control inputs. This structure in controller along with a integrator-like structure inherent in any controllable system allows us to compute the set of all the initial states and controller parameters that satisfy the safety constraints in closed form. This set of safe initial states and controller parameters forms the implicit RCIS, whose projection onto the coordinates of the initial states yields an RCIS of the original system. The closed-form construction of implicit RCIS makes our approach more scalable to high-dimensional systems. For instance, our approach can construct CIS for nominal systems of dimension 200 within 3 seconds, and construct RCIS for uncertain systems of dimension 20 within 5 seconds (see Section 4.6.3 for details). We further generalize this approach by enabling disturbance feedback in controller parameterization (Chapter 5), which significantly reduces the conservativeness of the original approach.

At a high level, the enhancement of scalability in our methods for constructing implicit RCIS is a result of leveraging the structure in controllable systems and the structure imposed on the parameterized controller. In concept, the more structures a dynamical system possesses, the higher the likelihood of improving the scalability of safety control synthesis. This intuition is proven to be true in the next part of this dissertation, where we consider discrete-time systems with input delays.

Input delay is ubiquitous in real-world systems. For instance, in vehicle control systems, input delay occurs due to actuator dynamics, communication delays in CAN bus, or delays in the upstream perception module. The dimension of systems with input delay grows linearly with the delay time, making the computation of its maximal RCIS rapidly intractable as the delay time increases. In Chapter 6, by exploiting the structure in input-delayed systems, we show that the maximal RCIS of an input-delayed system is embedded in the maximal RCIS of an auxiliary dynamics without delay. This observation enables a method of computing the maximal RCISs for input-delayed systems that scale well with the delay time. In particular, as the delay time increases, safety constraints become

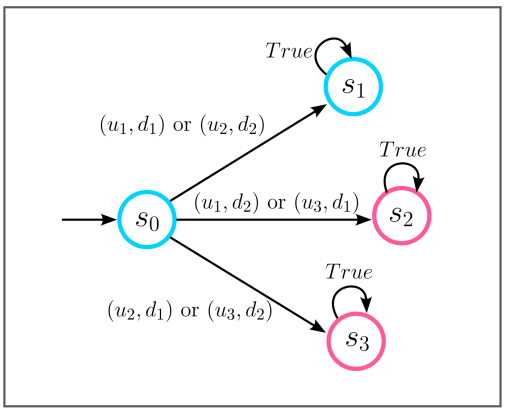


Figure 1.1: A transition system with four discrete states $\{s_0, s_1, s_2, s_3\}$, three controlled actions $\{u_1, u_2, u_3\}$ (control inputs), and two uncontrolled actions $\{d_1, d_2\}$ (disturbance). The state transition is determined by both controlled and uncontrolled actions. Let the safe set be $S = \{s_0, s_1\}$. The goal is to design a safe control policy such that the system state stays within S indefinitely robust to any uncontrolled actions. Suppose that at run time a control policy needs to determine the next controlled action without any knowledge of the next uncontrolled action. Then this goal can be achieved only if the system is initialized at s_1.

more challenging to satisfy, but the computational cost of the maximal RCIS remains relatively stable due to its connection to the auxiliary system without delay.

In addition to finding a scalable method of computing RCIS, Chapter 6 also provides an interesting observation: As the delay time grows, the maximal RCIS of a input-delayed system would continue to shrink until becoming empty. However, this negative impact of input delay can be compensated by adding preview on future disturbances. This motivates our study in the second part of this dissertation, namely how to systemically reduce conservativeness in safety control. Later we will show that preview on disturbance can not only compensate input delay, but also play a crucial role in improving the safety for generic systems. Prior to this, we need to explain the origins of the conservativeness in safety control.

Roughly speaking, the conservativeness of a safety controller can be evaluated by the size of the underlying RCIS it yields, since RCIS determines the subset of the state space where the safety controller is valid. Therefore, in principle, the most permissive safety controller is the one that yields the maximal RCIS. In the literature, most efforts (if not all) are put in how to find a tighter approximation of the maximal RCIS. The first part of this dissertation falls into this category, proposing approaches with a balance of conservativeness and scalability. However, in the second half of this dissertation, we want to answer a different question, namely how to enlarge the maximal RCIS of a given system. In other words, we are interested in reducing the conservativeness inherent in the system itself, instead of the conservativeness introduced by a specific safety control approach.

The key to reduce conservativeness is to exploit structure in disturbance. The main ideas behind

this can be explained via the toy example in Fig. 1.1: Given the safe set $S = \{s_0, s_1\}$, the maximal RCIS of this transition system is $\{s_1\}$, and thus existing safety control frameworks are only valid if the system starts at s_1. However, suppose that the system always starts at s_0. How can we ensure the safety of the system in this case? Apparently this is not possible unless we have additional knowledge of the disturbance. A simple solution is to assume that the control policy knows the next uncontrolled action in advance. Then, at s_0, a control policy can enforce the invariance of S by choosing u_1 for d_1, and u_2 for d_2. In this dissertation, we call this knowledge of future disturbance the *preview information*.

The idea of incorporating preview information into controller design has been explored extensively in the past [21], with many successful applications to real-world systems, such as autonomous vehicles [77, 120, 78], power systems [111] and humanoid robots [54]. Compared with purely state or output feedback control mechanisms, preview-based control allows feedforward control on available information of the incoming uncertainties to the system, and thus can substantially improve the control performance [21]. In the literature, there are several types of uncertainty considered "previewable". The first type is disturbances or uncontrolled inputs to the dynamical systems [78, 122]. Examples include road curvature for vehicles [120], or wind velocity for wind turbines [111], which can be previewed by dedicated perception systems [107, 71]. The second type of uncertainty is the reference signal in a reference tracking task [114, 28]. Unlike disturbances, its future values can be naturally obtained by the tracking controller at run time since the reference signal is commonly generated by other algorithmic components, such as a path planner. The third type of previewable uncertainty, recently studied in the context of online optimal control, is the unknown parameters in the optimization problem solved by the controller at run time [70, 109]. In many cases, one type of uncertainty (with the corresponding preview) can be converted easily to other types. For instance, unknown reference signals are modeled as disturbances in [122] and as unknown parameters in the cost function in [70]. In this dissertation, preview information always refers to the knowledge of future disturbance to the dynamics, that is, the first type above.

While prior research has explored preview information in control, fewer studies have investigated its application in safety control. In Chapter 7 of this dissertation, we study the safety control problem for continuous-state systems with preview, which, as explained in Chapter 2, is the dual of the problem for input-delayed systems. This duality relation makes the role of the preview horizon (with which the system dimension grows linearly) the opposite of the role of delay time. Intuitively, as we increase the preview horizon, the safety constraints become easier to satisfy, which explains why preview can compensate for the negative impact of input delay as discussed in Chapter 6. However, unlike the input-delayed case, the maximal RCIS computation still suffers from the curse of dimensionality. Choosing the preview horizon therefore becomes crucial since it determines the balance between safety and the computational cost. Conceptually, one can compute

the maximal RCIS at different preview horizons. The change in the size of the maximal RCIS reflects how much safety the system gains by using more preview information. However, this idea is computationally intractable. Instead, in Chapter 7, we introduce *safety regret* as a measure of the incremental value obtained from preview information in safety control. This metric is defined by the Hausdorff distance from a well-defined maximal safe set of the system with a finite preview horizon to that of a system with infinite preview. For discrete-time linear systems, under mild conditions, we prove that this safety regret converges to zero exponentially with the preview horizon. We also offer numerical algorithms for estimating the convergence rate of safety regret, enabling efficient evaluation of system safety at various preview horizons. In addition, although finding the maximal RCIS for systems with preview is generally challenging, there are specific classes of systems for which we can develop scalable approaches to efficiently compute their maximal RCIS. Three such classes of systems are presented in Chapter 8, where we leverage structure in the dynamics and/or preview for efficient computation.

While preview information can mitigate conservativeness in safety control and enable safe operation beyond the maximal RCIS, it is important to note that not all real-world systems have access to such preview information. In the final part of this dissertation, we explore strategies for ensuring safe operation beyond the maximal RCIS without relying on preview information. Let us gain some inspiration from the toy example in Fig. 1.1: Without preview on disturbance, there is no safe control policy that can ensure the system to stay within the safe set S when the system starts at s_0. However, note that if we apply either u_1 or u_2 at state s_0, depending on the uncontrolled action, there is still a chance that the system would transit to s_1 and be safe. However, if we apply u_3 at s_0, the system definitely leaves the safe set S in one step. Therefore, u_1 and u_2 are safer actions than u_3 at state s_0. Then, it is natural to come up with the idea that when it is impossible to be robust to all disturbance, one should synthesize a controller that always picks the most safe input at run time. In Chapter 9, we formalize this idea and propose a novel safety control framework called *opportunistic safety*. Unlike existing safety control frameworks, opportunistic safety controller can work both inside and outside the maximal RCIS, by identifying the worst-case disturbance model that can be handled at each state and constructing the control inputs robust to that worst-case disturbance model. We show that such disturbance models and control inputs can be jointly computed from the maximal RCIS for an auxiliary system, which can be approximated using existing tools. In simulation and in a drone experiment, this novel safety control framework shows great resilience to unexpected disturbances, and outperforms existing safety control approaches when operating outside the maximal RCIS.

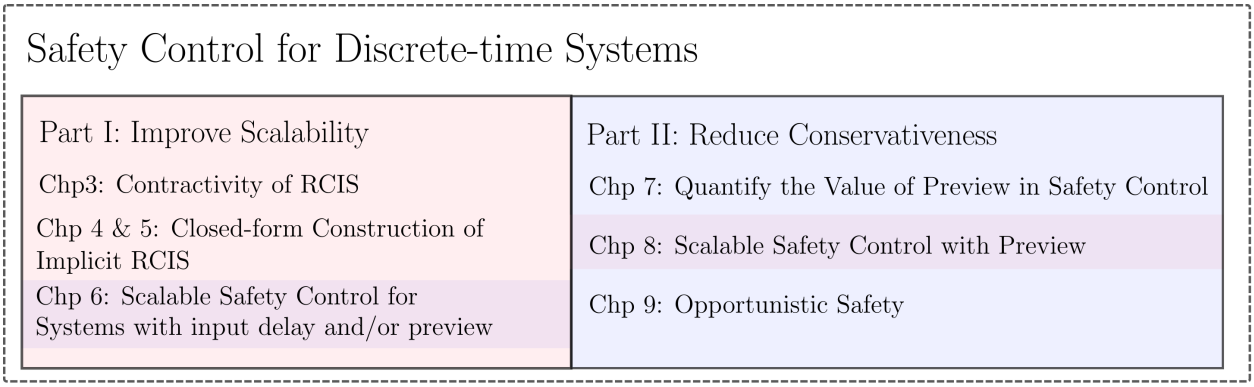


Figure 1.2: Dissertation Overview (Chapters 6 and 8 are highlighted since they are at the intersection of Parts I and II.)

1.1 Dissertation Overview and Summary of Contributions

This dissertation is structured in two parts. In the first part, we introduce safety control methods that can be applied to high-dimensional systems by exploiting structure in dynamics. In the second part, we investigate methods for reducing the conservativeness of safety control by leveraging structure in disturbance. Following is a brief summary of the main contributions from each chapter.

Part I: Scalable Safety Controller Synthesis: Design and Analysis

- In Chapter 3, we investigate the convergence properties of the k-step BRS for an RCIS. For linear systems with both control and disturbance inputs, we identify a condition for the exponential convergence of the k-step backward reachable set to the maximal RCIS. This condition is readily verifiable through linear programming when all sets are represented as polytopes. To our knowledge, this is the first such result in the literature for systems with additive disturbances. Results in this chapter are published in [80].

- In Chapter 4, we propose a method that computes implicit RCISs for linear systems with polytopic constraints in closed form, by lifting the original system to a higher-dimensional system with an eventually periodic structure. Due to the closed-form expression of the implicit RCISs, the proposed method is guaranteed to terminate in finite time, and is more scalable to the system dimension. Several case studies are provided to show the efficiency of our method and usages of the implicit RCIS. Furthermore, we present an extended version of this method in Chapter 5, which mitigates the conservativeness inherent in the original approach by allowing for disturbance feedback within the parameterized controllers. Results in these two chapters are reported in [7, 75].

- In Chapter 6, we address the safety control problem for input-delayed systems. We prove that the maximal RCIS of input-delayed systems is embedded in the maximal RCIS of a low-dimensional system without input delay. Leveraging this structural property, we propose an efficient method for computing the maximal RCIS of linear systems with input delays. We further extend this method to incorporate additional preview information on disturbance while preserving computational efficiency. Compared with the baseline approach of constructing a higher dimensional system by appending the state space with the delayed inputs and previewed disturbances, our method demonstrates superior scalability as the delay time increases. Results in this chapter are published in [83].

Part II: Safety Control Beyond the Maximal RCIS

- In Chapter 7, we explore the utility of preview information in safety control. We uncover novel structural properties of the maximal RCIS for systems with preview, enabling the approximation of the maximal RCIS from both inside and outside. To assess the value of preview information, we introduce the concept of *safety regret* as the Hausdorff distance between a properly defined maximal safe set of a system with finite preview and that of a system with infinite preview. We prove that this metric converges to zero exponentially with the preview horizon. Additionally, we develop algorithms to numerically evaluate the safety regret for various preview horizons. The theory and algorithms are illustrated using real-world examples. We also extend our analysis of safety regret to demonstrate how the preview horizon impacts the feasible domain of preview-based model predictive control (MPC) under different recursive feasibility constraints. Results in this chapter are reported in [81].

- In Chapter 8, we introduce efficient methods for computing the maximal RCIS for three specific classes of systems with preview. The first class comprises systems in Brunovsky canonical form with hyperbox safe sets. We show that the maximal RCIS can be obtained in closed form for systems in this class with preview. The second class encompasses switched linear systems with a preview on the switching signal. We introduce a novel modeling mechanism called *preview automaton* to model constraints on preview horizon, and then present an efficient algorithm for computing the maximal RCIS for linear switched systems equipped with preview automata. The last class involves systems with uncertain preview, posing a special output-feedback safety control problem. While the general solution to output-feedback safety control problem involves the maximal RCIS of a set dynamics, by exploiting a nilpotent structure inherent in systems with preview, we show that this set dynamics can be reduced to a finite-dimensional system, enabling efficient controller synthesis. Part of the results in this chapter is published in [78, 77].

- In Chapter 9, we propose a novel safety control framework that can work both inside and outside the maximal RCIS, by identifying the worst-case disturbance that can be handled at each state and constructing the control inputs robust to that worst-case disturbance model. We show that such disturbance models and control inputs can be jointly computed by considering an invariance problem for an auxiliary system. Finally, we demonstrate the efficacy of our method both in simulation and in a drone experiment. Results in this chapter are reported in [76].

Authorship Acknowledgment. Chapters 4 and 5 include the works [7, 75] published with Dr. Tzanis Anevlavis and Prof. Paulo Tabuada at UCLA. Chapter 6 presents the work in [83] published with Dr. Liren Yang at HUST. Chapter 9 presents the work in [76] published with Hao Chen at University of Michigan and Prof. Yulong Gao at Imperial College London.

Acknowledgment of other contributions. Below are contributions made during my PhD study which are not included within the scope of this dissertation:

- We propose an approach that computes inner-approximated BRSs for discrete-time nonlinear systems based on their Koopman approximations learned from data [18].
- We prove that continuous-time systems with multiple ω-limit sets cannot be immersed in a linear system, which sheds lights on the applicability of Koopman operator theory in learning linear representations for nonlinear systems [82].
- We provide a sample complexity analysis of ordinary least squares method for identifying over-parameterized ARX models, and show that the identified over-parameterized model converges to the minimal ground-truth model without any regularization, which we refer to as the *self regularization* property [33].
- Given a collection of normal and abnormal agents, we propose a inverse reinforcement learning based framework that can detect the abnormal agents from the normal ones by comparing reward functions learned for each individual agent with a common reward function [68].
- We propose a novel safety supervisor that smoothly overrides user inputs by blending the user inputs with some optimal safe inputs based on the value of an RCIS induced CBF. Our approach can effectively prevent systems from the chattering behaviors observed in conventional minimally invasive safety supervisor when operating near the boundary of the RCIS [14].

- In the context of driving scenarios, we propose a guardian architecture consisting of an driver intention estimator and a safety supervisor. The safety supervisor is designed to adjust its response based on real-time intention estimation, effectively reducing unnecessary caution while maintaining safety guarantees [105].

1.2 Related Works

RCIS and its backward reachable sets. Robust controlled invariant set (RCIS) has a wide range of applications in control and verification for safety-critical systems, such as autonomous vehicles [57, 35, 91, 29], robotic systems [110, 4], and power systems [51]. In many lines of work regarding RCIS, the backward reachable sets of RCISs need to be computed implicitly or explicitly. For instance, the computation of the maximal RCIS is known to be a difficult problem [20, 32, 103]. A common means to efficiently inner-approximate the maximal RCIS is by computing the backward reachable sets of a small but easy-to-compute RCIS [32, 103, 36, 119, 8], which is referred to as the *inside-out* algorithm in this dissertation. In the context of the constrained MPC, the terminal set of the MPC is typically chosen as an RCIS to guarantee constraint satisfaction and recursive feasibility [23, 74, 50]. The domain of attraction (DoA) of the MPC is then exactly equal to the T-step backward reachable set of the terminal set, where T is the prediction horizon [74]. More recently, inspired by MPC, a control framework referred to as safety filter is proposed for discrete-time systems [117] and continuous-time systems [110, 45] to equip a given nominal controller with safety guarantees in a minimally invasive way. In discrete time, the connection between the DoA of a safety filter and the backward reachable sets of RCISs is similar to that in MPC. In continuous time, a finite-step backward reachable set of some given RCIS is constructed implicitly, to improve the scalability and reduce the conservatism of the safety filter [110, 45].

Implicit RCISs. In Chapter 4, we propose a method that computes implicit RCISs in closed form. The concept of implicit RCISs is initially explored in [36] for nominal systems and in [100, 98, 119] for systems with disturbances, where sufficient conditions on set recurrence are checked by linear programs. Since no iterative set operation is involved, these methods scale well with the system dimension compared with the standard method in [20] for computing the maximal RCISs for discrete-time systems. Ideas similar to the implicit RCISs in Chapter 4, in the sense that finite input sequences are used, have also been explored in the context of MPC [88]. The goal there is to establish the asymptotic stability of a linear system, whereas in our case we exploit finite input sequences that describe an eventually periodic input signal, which leads to a closed-form expression for an implicit representation of controlled invariant sets. Other popular approaches first close the loop with a linear state-feedback control law, and then compute an invariant set of the closed-loop

system. Under this umbrella, an idea close to ours is reference governor, where the control loop is closed by a constant controller and the maximal admissible output set of the closed loop forms an implicit RCIS [59]. Another idea close to ours is found in [65], where recurrent sets are computed in the context of MPC without disturbances. These two approaches can be understood as special cases of our method in Chapter 4.

Safety control for systems with input delays. There are several results in the literature related to the invariance problem for time-delay systems. Lyapunov-based methods have been recently explored by [95, 52]. In particular, safety barrier functionals are proposed in [95] for general continuous-time nonlinear autonomous time-delay systems, and the "Artstein" model reduction method [12] is used in [52] for computing control barrier functions for continuous-time linear systems with input delay. For methods based on discrete-time linear systems, the work in [94] tackles time-varying input delay using polytope approximations, and then computes the maximal output admissible set of the closed-loop system stabilized by a linear feedback law. The work in [86, 85], on the other hand, compute invariant sets for discrete-time autonomous time-delay systems with different levels of conservativeness. In this dissertation, we are interested in finding the maximal RCISs for discrete-time systems with input delays. This problem cannot be handled by the above methods, since they are mainly designed for systems without disturbance and lack any guarantees on the maximality of the resulting controlled invariant set.

Safety control for systems with preview. Preview-based control has a rich literature [21], but the majority of the literature focuses on incorporating preview information into optimal control formulation [108, 114, 55, 120]. A prime example is MPC [40, 63, 122], where preview information is naturally incorporated into the state propagation constraints. In this case, the improvement due to the preview is measured by the amount of cost reduction after increasing the preview horizon. A recent work [122] proves in theory that the cost reduction in both the linear quadratic control and MPC formulations decays exponentially as the preview horizon increases. Similar results on the cost reduction with respect to the preview horizon can also be found in [114, 70, 109] under different optimal control setups. However, these results are not applicable to safety control problems, since the class of cost functions considered by [122, 114, 70, 109] cannot characterize state-input constraints. Notably, the impact of preview horizon is a relatively well-studied problem in linear temporal logic (LTL) synthesis [62, 49, 58], where preview is called lookahead. The work in [62] provides some extreme case analysis by checking the universal satisfiability of the LTL formula, which is similar to what we do in Chapter 7. The work in [58] provides upper and lower bounds on the preview horizon necessary for the existence of a controller that realizes a LTL specification, which sheds light on the impact of different preview horizons. Unfortunately, these analyses are for

finite-state transition systems only, which cannot be easily generalized for systems with real-valued states like those studied in Chapters 7 and 8.

Safety control beyond the maximal RCIS. The opportunistic safety control framework introduced in Chapter 9 draws inspiration from the line of work on best-effort synthesis in the field of reactive synthesis, as exemplified by [6, 37]. In these works, the goal is to synthesize controllers for finite-transition systems with respect to LTL specifications. The central idea is that if there is no winning strategy robust to all disturbances, one should select a strategy that is at least as good as the other strategies in terms of the disturbances it can robustly handle. Similar concepts are explored in abstraction-based control [106] and finite-horizon constrained optimal control [39]. To the best of our knowledge, the framework presented in Chapter 9 is the first to apply this idea to the safety control problem for continuous-state systems.

In the literature, there are other approaches aimed at extending safety controllers beyond the maximal RCIS. For instance, Li et al. in [69] propose a method to potentially expand the domain of a safety supervisor beyond the maximal RCIS by addressing a time-optimal control problem when the safety supervisor is undefined. As shown by the numerical examples in Section 9.3, our method demonstrates a significant performance improvement over the method in [69] when the system is outside the maximal RCIS and the disturbance is not entirely adversarial. Another line of work, such as [96, 116, 66] explores the possibility of relaxing the state-input constraints to find a larger maximal RCIS. These works typically assume that part of the constraints is not safety-critical and thus can be relaxed as needed [96], or a priority among different constraints is offered by the user [116, 66]. It is important to note that our approach does not rely on these assumptions, making direct comparison with these methods less relevant. In cases where the system of interest is stochastic, controllers based on the maximal probabilistic RCIS, as in [38, 2], can minimize the risk of constraint violation in the infinite horizon, even if the system is outside the maximal RCIS. This might be appropriate when the statistics of the disturbances are known. On the other hand, our method operates without requiring knowledge of the disturbance statistics and is more computationally tractable because RCISs are easier to compute than probabilistic RCISs.

CHAPTER 2

Preliminaries

To establish a precise definition for a control synthesis method, we typically specify three essential components: the system model, the specification (or control target), and the synthesis algorithm, as shown in Figure 2.1. In this chapter, we introduce the mathematical preliminaries of safety control from those three aspects.

Chapter Overview. We first define the dynamical systems considered in this dissertation in Section 2.2, and then introduce the safety specification in Section 2.3 and the definition of controlled invariance in Section 2.4. Finally, we present two basic algorithms of computing the maximal RCIS in Section 2.5.

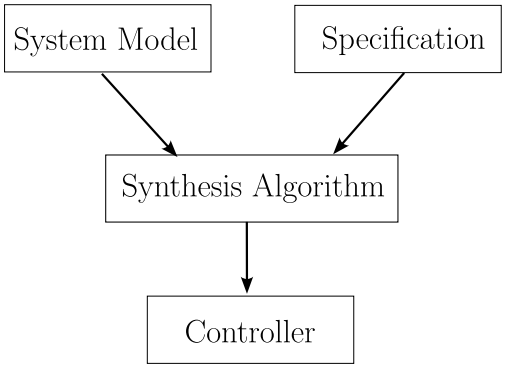


Figure 2.1: General diagram of controller synthesis methods

2.1 Mathematical Notations

Following is a list of mathematical notations used throughout the entire dissertation:

- The symbols $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{Z}^+$ denote the set of real numbers, the set of non-negative integers, and the set of positive integers respectively.

- For a function $f : X \to Y$ and a subset $X_0 \subseteq X$, the image of X_0 under f is denoted by $f(X_0)$.
- For a matrix $A \in \mathbb{R}^{n\times n}$ and a set $X \subseteq \mathbb{R}^n$, the set $\{Ax \mid x \in X\}$ is denoted by AX.
- For a scalar α and a set $X \subseteq \mathbb{R}^n$, the set $\{ax \mid x \in X\}$ is denoted by αX.
- The projection of a set $X \subseteq \mathbb{R}^{n+m}$ onto the first n coordinates is denoted by

$$\pi_{[1,n]}(X) = \{x_1 \in \mathbb{R}^n \mid (x_1, x_2) \in X\}.$$

- The convex hull of a subset X of $\mathbb{R}^n$ is denoted by $\text{conv}(X)$.
- The Minkowski sum of two sets X and Y is denoted by $X + Y = \{x + y \mid x \in X, y \in Y\}$. When $Y = \{y\}$ is a singleton, the Minkowski sum $X + Y$ is written as $X + y$ for short. Also, for a set X and a point y, $X - y = X + (-y)$.
- The Minkowski difference of two sets X and Y is denoted by $X - Y = \{x \in \mathbb{R}^n | x + Y \in X\}$.
- For a collection of sets $\{S_i\}_{i=1}^n$, the cartesian product of S_i for i from 1 to n is denoted by $S_1 \times S_2 \times ... \times S_n$. For the case $S_1 = S_2 = ... = S_n = S$, $S_1 \times ... \times S_n$ is denoted by S^n for short.
- For vectors $x_1 \in \mathbb{R}^n$ and $x_2 \in \mathbb{R}^m$, $(x_1, x_2) \in \mathbb{R}^{n+m}$ denotes the concatenation of the vectors.
- For a sequence $(x(t))_{t\geq 0}$, we denote its finite segment $(x(t))_{t=a}^b$ for some $a, b \in \mathbb{N}$ by $x(a : b)$.
- For a set of indexed variables u_1, ..., u_k in $\mathbb{R}^m$, their concatenation $(u_1, \cdots, u_k) \in \mathbb{R}^{mk}$ is denoted by $u_{1:k}$.

2.2 System Descriptions

We consider discrete-time dynamical systems in form of

$$x(t+1) = f(x(t), u(t), d(t)), \tag{2.1}$$

with states $x \in X$, inputs $u \in U$ and disturbance $d \in D$. The state space X and the input set U can be continuous (such as convex sets in $\mathbb{R}^n$), discrete (such as subsets of $\mathbb{N}$), or hybrid (that is, some coordinates are continuous and some are discrete). The disturbance set D contains all the possible disturbance inputs and is assumed to be bounded when the state space X is continuous.

We restrict the scope of this dissertation to *state-feedback* controller synthesis. A *static* state-feedback controller for a system in (2.1) is a function $\mathbf{u} : \mathbb{R}^n \to U$ that maps each x to a control

input $\mathbf{u}(x)$ in the input set U. In contrast, a *dynamic* state-feedback controller is a discrete-time system in the form of

$$x_c(t+1) = g(x_c(t), x(t)) \tag{2.2}$$

$$u(t) = h(x_c(t)), \tag{2.3}$$

where $x_c \in X_c$, $x \in X$, and $u \in U$ are the internal states, input, and output of the controller. The usages of dynamic controllers in safety control are explored in Chapters 4 and 5. In the rest of this dissertation, a controller refers to a static state-feedback controller if no further clarification is made. The following are several classes of discrete-time systems considered in this dissertation.

Linear systems. A system in (2.1) is *linear* if there exist matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $E \in \mathbb{R}^{n\times l}$ such that

$$f(x,u,d) = Ax + Bu + Ed, \tag{2.4}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $d \in D \subseteq \mathbb{R}^l$.

Systems with input delay. A discrete-time system Σ_{delay} with τ-step input delay is in the form of

$$\Sigma_{delay}:\ x(t+1) = f(x(t), u(t-\tau), d(t)), \tag{2.5}$$

with $x(t) \in X$, $u(t) \in U$ and $d(t) \in D$. A system in (2.5) can be written in the form of (2.1) by appending the past τ-step inputs to the state space. The resulting $(n+m\tau)$-dimensional augmented system Σ_τ takes the form:

$$\Sigma_\tau : \begin{bmatrix} x(t+1) \\ u_1(t+1) \\ \vdots \\ u_{\tau-1}(t+1) \\ u_\tau(t+1) \end{bmatrix} = \begin{bmatrix} f(x(t), u_1(t), d(t)) \\ u_2(t) \\ \vdots \\ u_\tau(t) \\ u(t) \end{bmatrix} \tag{2.6}$$

with $x(t) \in X$, $u(t) \in U$ and $d(t) \in D$.

Systems with preview. A system Σ in the form of (2.1) is said to have *p-step preview* if at each time step t, we have the measurements of

- the current state $x(t)$, and

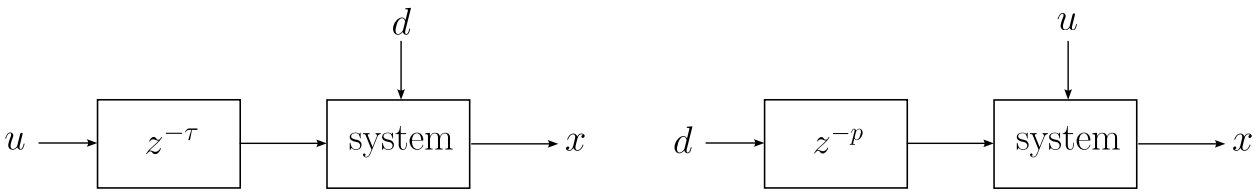


Figure 2.2: Block diagrams of input-delay systems (left) and systems with preview (right), where τ and p are the delay steps and the preview horizon, and z^{-1} is the time-shift operator.

- the current and incoming disturbances $\{d(t+k)\}_{k=0}^{p-1}$ in p steps, denoted by $d_{1:p}(t)$ for short.

The scalar p is called the *preview horizon*. Similar to systems with input delay, we define an augmented system whose states stack the states $x(t)$ and the previewed disturbances $d_{1:p}(t)$ of Σ, called the p-augmented system Σ_p. The p-augmented system of Σ has the form

$$\Sigma_p : \begin{bmatrix} x(t+1) \\ d_1(t+1) \\ \vdots \\ d_{p-1}(t+1) \\ d_p(t+1) \end{bmatrix} = \begin{bmatrix} f(x(t),u(t),d_1(t)) \\ d_2(t) \\ \vdots \\ d_p(t) \\ d(t) \end{bmatrix} \tag{2.7}$$

with state $(x,d_1,\cdots,d_p) \in X \times D^p$, input $u \in U$ and disturbance $d \in D$.

Duality between input delay and preview. Comparing augmented systems in (2.6) and (2.7), one can immediately realize if we switch the role of control input u and disturbance input d, an input-delayed system becomes a system with preview and vice verse. Indeed, one can view a system with p-step preview as a system with p-step time delay in the uncontrolled input $d(t)$, as shown in Figure 2.2. This is what we call the *duality between input delay and preview*.

Due to this duality relation, those two classes of systems share many dual properties. For instance, intuitively, an input-delayed system becomes harder to control as the delay steps τ increases, while a system with preview becomes easier to control as the preview horizon p increases. In Chapter 6, we show that for a system with both input delay and preview, their effects on the safety of the system are canceled. This duality also suggests that the safety control problem for systems with preview can be harder to deal with, as the dual problem, the safety control for input-delayed systems, is easy (Chapter 6). In Chapter 7, we present nontrivial properties of the maximal RCIS for systems with preview by exploring this duality between input delay and preview.

2.3 Safety Control Problem

For a system as in (2.1), suppose that S_{xu} is the *safe set* that contains all the desired state-input pairs (x,u) in $X \times U$. The safety specification of the system is to maintain the state-input pairs (x,u) in the safe set S_{xu} indefinitely, robust to all possible disturbances. More formally, the safety specification can be defined as the set $\mathscr{T}_{safe}$ of state-input trajectories in S_{xu}, that is

$$\mathscr{T}_{safe} = \{(x(t),u(t))_{t=0}^{\infty} \mid x(t+1) \in f(x(t),u(t),D), (x(t),u(t)) \in S_{xu}, \forall t \in \mathbb{N}\}, \tag{2.8}$$

where $f(x,u,D) = \{f(x,u,d) \mid d \in D\}$. Given any state-input trajectory of the system, we say the trajectory *satisfies* the safety specification if this trajectory is contained in $\mathscr{T}_{safe}$. Now, for a given controller $\mathbf{u} : X \to U$ and an initial state $x_0 \in X$, we define the set $\mathscr{T}(\mathbf{u},x_0)$ of the closed-loop state-input trajectories starting at x_0 by

$$\mathscr{T}(x_0,\mathbf{u}) = \{(x(t),\mathbf{u}(x(t)))_{t=0}^{\infty} \mid x(0) = x_0, x(t+1) \in f(x(t),\mathbf{u}(x(t)),D), \forall t \in \mathbb{N}\}. \tag{2.9}$$

Definition 2.1. A controller $\mathbf{u} : X \to U$ is *safe* at the initial state x_0 if all the corresponding closed-loop state-input trajectories starting at x_0 satisfy the safety specification, that is $\mathscr{T}(x_0,\mathbf{u}) \subseteq \mathscr{T}_{safe}$.

Problem 2.1 (Safety control problem)**.** *Given a system model as in* (2.1) *and a safe set S_{xu}, find the initial states x_0 and controllers $\mathbf{u} : X \to U$ such that $\mathscr{T}(x_0,\mathbf{u}) \subseteq \mathscr{T}_{safe}$.*

Remark 2.1. Problem 2.1 has different variants in the literature [38]. For instance, when the disturbance in (2.1) is stochastic, we can relax the inclusion condition $\mathscr{T}(x_0,\mathbf{u}) \subseteq \mathscr{T}_{safe}$ in Problem 2.1 to be satisfied with high probability [38]. In this dissertation, we focus on the deterministic version of the safety control problem, namely Problem 2.1.

2.4 Reachability Analysis and Invariance

In this section, we define RCIS of a system, which is the key for solving Problem 2.1. We first introduce the definition of *one-step backward reachable set (BRS)*.

Definition 2.2. Given a system model Σ as in (2.1), a safe set S_{xu} and a target set $X_0 \subseteq X$, the *one-step BRS* $Pre_{\Sigma}(X_0,S_{xu},D)$ of the target set X_0 with respect to the system Σ, the safe set S_{xu}, and the disturbance set D is the set of states x where there exists an input u such that the state-input pair (x,u) is in the safe set S_{xu} and the next state stays in X_0 robust to any disturbance in D, that is

$$Pre_{\Sigma}(X_0,S_{xu},D) = \{x \mid \exists u, (x,u) \in S_{xu}, f(x,u,D) \subseteq X_0\}. \tag{2.10}$$

In the literature, $Pre_\Sigma(X_0, S_{xu}, D)$ is also called *the controlled predecessors* [72], which is the origin of the acronym "*Pre*". We define the k-step BRS $Pre^k_\Sigma(X_0, S_{xu}, D)$ for all $k \in \mathbb{N}$ recursively by

$$Pre^0_\Sigma(X_0, S_{xu}, D) = X_0, \tag{2.11}$$

$$Pre^k_\Sigma(X_0, S_{xu}, D) = Pre_\Sigma(Pre^{k-1}_\Sigma(X_0, S_{xu}, D), S_{xu}, D),\ \forall k > 0. \tag{2.12}$$

When the safe set S_{xu} and disturbance set D are clear from the context, we denote the k-step BRS of X_0 by $Pre^k_\Sigma(X_0)$ for short.

Definition 2.3. Given a system model Σ as in (2.1) and a safe set S_{xu}, a set $C \subseteq X$ is a *robust controlled invariant set* (RCIS) of the system Σ with respect to the safe set S_{xu} if for all $x \in C$, there exists an control input u such that $(x,u) \in S_{xu}$ and for all $d \in D$, $f(x,u,d) \in C$. Equivalently, C is an RCIS of Σ with respect to S_{xu} if and only if $C \subseteq Pre_\Sigma(C, S_{xu}, D)$.

When the disturbance set $D = \{0\}$, we call an RCIS a *controlled invariant set* (CIS). The maximal RCIS is then equal to the union of all RCISs of Σ with respect to S_{xu}. It can be shown that C_{max} is a fixed point of the *Pre* operator, that is,

$$C_{max} = Pre_\Sigma(C_{max}, S_{xu}, D). \tag{2.13}$$

By definition, for any state x in an RCIS C, one can always find an admissible input $u : C \to \mathbb{R}^m$ such that $(x,u) \in S_{xu}$, and the next state $x^+ = f(x,u,d)$ is in C for any disturbance $d \in D$. The set of all such inputs at a given state is called the *admissible input set*.

Definition 2.4. Given a controlled invariant set C of the system Σ, the *admissible input set* at the state $x \in C$ is defined by

$$\mathscr{A}(x, C) = \{u \in \mathbb{R}^m \mid (x,u) \in S_{xu}, f(x,u,d) \in C\}. \tag{2.14}$$

By definition of RCIS, $\mathscr{A}(x,C)$ is nonempty for any $x \in C$. Next, the connection between the maximal RCIS of Σ and the solutions to Problem 2.1 is summarized in the following:

- A safe controller $\mathbf{u} : X \to U$ exists at a given initial state $x_0 \in \mathbb{R}^n$ if and only if x_0 is in the maximal RCIS C_{max} of Σ with respect to S_{xu};
- A controller $\mathbf{u} : X \to U$ is safe for any initial state $x_0 \in C_{max}$, if and only if $\mathbf{u}(x) \in \mathscr{A}(x, C_{max})$ for all $x \in C_{max}$.

The "if and only if" statements in the two preceding bullet points become "if" statements when we substitute the maximal RCIS C_{max} with an arbitrary RCIS C of Σ with respect to S_{xu}. Therefore, solving Problem 2.1 is equivalent to finding the maximal (or any) RCIS of the system Σ.

As an application of the second bullet point above, if a reference controller $\mathbf{u}_{ref}$ and an RCIS C are given, a safety supervisor $\mathbf{u}$ can be synthesized by minimally modifying the reference controller,

$$\mathbf{u}(x) = \min_{u \in \mathscr{A}(x,C)} \|u - \mathbf{u}_{ref}(x)\|_2^2, \ \forall x \in C. \tag{2.15}$$

By construction, this safety supervisor $\mathbf{u}$ is safe for all initial states $x_0 \in C$.

Finally, a notion similar to RCIS but for autonomous systems is the so-called robust positively invariant set (RPIS), which would be used to construct implicit RCIS later in Chapters 4 and 5.

Definition 2.5. Given an autonomous system $x^+ = f(x,d)$ with state $x \in \mathbb{R}^n$ and disturbance $d \in D \subseteq \mathbb{R}^l$ and a safe set $S_x \subseteq \mathbb{R}^n$, a set $C \subseteq S_x$ is a *robust positively invariant set* (RPIS) of the system within S_x if:

$$\forall x \in C \text{ we have that } f(x,d) \in C, \forall d \in D. \tag{2.16}$$

A set C_{max} is the *maximal RPIS* within S_x if it is positively invariant and contains every robust positively invariant set in S_x.

2.5 Iterative Algorithms for Computing RCISs

Given a system Σ as in (2.1) and a safe set S_{xu}, there are two standard iterative algorithms that can approximate the maximal RCIS:

The outside-in algorithm [20, 103]: Let $C_0 = \pi_{[1,n]}(S_{xu})$ be the projection of the safe set S_{xu} onto the x coordinates. Recursively compute the k-step BRS C_k of C_0 with respect to S_{xu} until the termination condition $C_k = C_{k-1}$ is satisfied.

The inside-out algorithm [32, 103]: Let C_0 be a known RCIS of Σ in S_{xu}. Recursively compute the k-step BRS C_k of C_0 with respect to S_{xu} until the termination condition $C_k = C_{k-1}$ is satisfied or k reaches a user-defined maximum iteration number k_{max}.

Algorithm 1 The outside-in algorithm

Input: System dynamics Σ and the safe set S_{xu}
Initialization: $k \leftarrow -1$, $C_0 \leftarrow \pi_{[1,n]}(S_{xu})$
do
  $k \leftarrow k+0$
  $C_k \leftarrow Pre_\Sigma(C_{k-1}, S_{xu}, D)$
while $C_k \neq C_{k-1}$
return C_k

Algorithm 2 The inside-out algorithm

Input: System dynamics Σ, the safe set S_{xu}, an RCIS C_{-1} of Σ in S_{xu}, and the maximal iteration number k_{max}
Initialization: $k \leftarrow -1$
do
$k \leftarrow k+0$
$C_k \leftarrow Pre_\Sigma(C_{k-1}, S_{xu}, D)$
while $C_k \neq C_{k-1}$ or $k < k_{max}$
return C_k

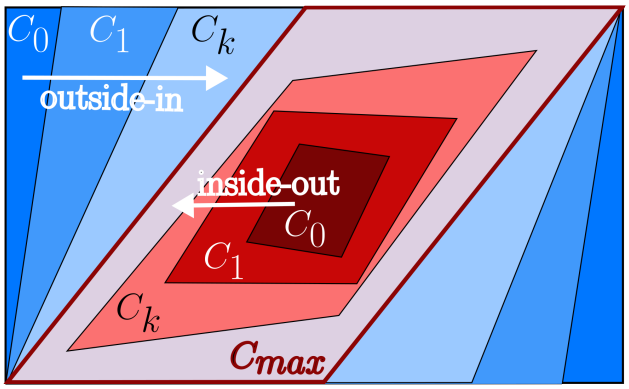


Figure 2.3: Demonstrations of the inside-out and outside-in algorithms. The BRSs C_k of the outside-in algorithm are depicted in blue; those of the inside-out algorithm are in red.

The pseudocode of both algorithms are provided in Algs. 1 and 2. The k-step BRS C_k in the outside-in algorithm monotonically shrinks with k and outer approximates C_{max}, while the set C_k the inside-out algorithm monotonically expands and inner approximates C_{max}. Fig. 2.3 demonstrates how those two algorithms approximate the maximal RCIS from different directions. Under mild conditions, the BRS C_k in the outside-in algorithm converges to the maximal RCIS [20]. The conditions under which the inside-out algorithm converges to the maximal RCIS is studied in Chapter 3.

Both algorithms can not guarantee to terminate within finite steps [103]. However, when prematurely terminated at step k, the set C_k in the outside-in algorithm is not necessarily robust controlled invariant, and thus cannot be used in control synthesis; in contrast, the set C_k in the inside-out algorithm is always an RCIS. Due to this reason, the inside-out algorithm is also called an *anytime* algorithm [103], as users can stop the algorithm at any step k and use the RCIS C_k in control synthesis.

Definition 2.6. A set $C \subseteq \mathbb{R}^n$ is a polytope if there exist a matrix H and a vector h such that $C = \{x \in \mathbb{R}^n \mid Hx \leq h\}$. The tuple (H,h) is called the H-representation of the polytopic set C.

If the system is linear, and the safe set S_{xu}, the disturbance set D and the initial set C_0 are all

polytopic, the outside-in algorithms and the inside-out algorithms can be easily implemented via the MATLAB toolbox MPT3 [48].

Part I

Scalable Safety Controller Synthesis: Design and Analysis

CHAPTER 3

On the Attractivity of the Maximal RCIS in Backward Reachability Analysis

Due to the pivotal role of the backward reachable set (BRS) of RCISs in safety control (see a brief literature review in Section 1.2), it is important to understand its convergence properties, including (i) what conditions ensure that the BRSs of an RCIS converges to the maximal RCIS and (ii) how fast the BRSs of an RCIS converge. In the literature, the existing works in this direction can only deal with systems without disturbances: Ahmadi and Gunluk [3] study asymptotically stable autonomous systems and provide a sufficient condition under which the k-step BRS of an invariant set converges to the maximal invariant set in finite steps. The papers [30, 46, 32, 31] study controllable systems without disturbances and show a sufficient condition under which the k-step BRS of a controlled invariant set (CIS) converges to the maximal CIS in Hausdorff distance, with an exponential convergence rate. When the system is asymptotically stabilizable without disturbances, Santis et al. [32] present a sufficient condition under which the BRSs of a small CIS converge to the maximal CIS, but does not reveal the convergence rate.

As the main contribution of this chapter, for linear systems with both control and disturbance inputs, we develop a mild sufficient condition under which the k-step BRS of an RCIS converges to the maximal RCIS in Hausdorff distance, with an exponential convergence rate. Furthermore, when all sets are represented by polytopes, this sufficient condition can be easily checked by a linear program. To the best of our knowledge, our work is the first result in the literature that shows these convergence properties for systems with additive disturbances. When restricted to systems without disturbances, the existing results in [46, 32, 31, 30] are shown to be special cases of our result. In addition, our result extends the results for stabilizable systems in [32] by showing the convergence rate is exponential.

Chapter Overview. We state the problem in Section 3.1, followed by the main results in Section 3.2 and the proofs in Section 3.3. We illustrate our results with a numerical example in Section 3.4.

After that, we compare our result with existing results in the literature in Section 3.5 and conclude the chapter in Section 3.6. The proofs of all the minor theorems are found in Appendix A.

Notation. We define the distance from a point x to a set Y by $d(x,Y) = \inf\{\|x-y\|_2 \mid y \in Y\}$, and then define the Hausdorff distance of two sets X and Y by

$$d_H(X,Y) = \max(\sup_{x\in X} d(x,Y), \sup_{y\in Y} d(y,X)).$$

For a subset X of $\mathbb{R}^n$, the interior and closure of X are denoted by $int(X)$ and $cl(X)$. The ε-ball centered at x is denoted by $B_\varepsilon(x) = \{y \mid \|x-y\|_2 < \varepsilon\}$.

3.1 Problem Setup

We consider a discrete-time linear system Σ in form of

$$\Sigma : x(t+1) = Ax(t) + Bu(t) + Ed(t), \tag{3.1}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $d \in D \subseteq \mathbb{R}^l$. The state-input constraints of the system Σ are specified by the *safe set* S_{xu}, which contains all the desired state-input pairs (x,u). We make the following assumption throughout this chapter.

Assumption 3.1. *The safe set S_{xu} and the disturbance set D are convex and compact.*

Given C_0 is an RCIS of Σ with respect to S_{xu}, we denote the k-step BRS of C_0 by C_k. This sequence $(C_k)_{k=0}^{\infty}$ of sets is an expanding family of RCISs. That is, for all $k \geq 0$, C_k is robust controlled invariant in S_{xu}, and $C_{k+1} \supseteq C_k$. Thus, the set-theoretical limit C_∞ of the k-step BRS of C_0 as k goes infinity is well defined (see e.g. [101] for the definition of the set-theoretical limit):

$$C_\infty = \cup_{k=0}^{\infty} C_k. \tag{3.2}$$

In this chapter, we also refer to the limit C_∞ as the *infinite-step* BRS of C_0. It can be shown that C_∞ is an RCIS with respect to S_{xu}. In this chapter, we want to answer the following two questions:

(i) What condition ensures that the k-step BRS C_k converges to the maximal RCIS C_{max} in Hausdorff distance, that is, $d_H(C_\infty, C_{max}) = 0$?
(ii) How fast does the k-step BRS C_k converge to its limit C_∞?

3.2 Main Results

Before we state our main result, let us first gain some intuitions from a toy example.

Example 3.1. Consider the 1-dimensional system

$$x(t+1) = \alpha x(t) + u(t) + d(t), \tag{3.3}$$

with $x \in \mathbb{R}$, $u \in \mathbb{R}$, and $d \in [-d_{max}, d_{max}]$. The safe set is $S_{xu} = [-x_{max}, x_{max}] \times [-u_{max}, u_{max}]$. Define

$$c_d = (d_{max} - u_{max})/(1 - |\alpha|). \tag{3.4}$$

Consider symmetric RCIS C_0 in form of $[-c_0, c_0]$ with $c_0 \leq x_{max}$. Then, the k-step BRS C_k of C_0 is symmetric, and if $|\alpha| \notin \{0, 1\}$, C_k is equal to $[-c_k, c_k]$ with

$$c_k = \min\left(\frac{c_0 - c_d}{|\alpha|^k} + c_d,\ x_{max}\right). \tag{3.5}$$

Case 1: Suppose $|\alpha| \in (0,1)$, $u_{max} \leq d_{max}$ and $x_{max} > c_d \geq d_{max}$. For any $c_0 \in [c_d, x_{max}]$, $C_0 = [-c_0, c_0]$ is an RCIS. The maximal RCIS is $[-x_{max}, x_{max}]$.

According to (3.5), if we select $c_0 \in (c_d, x_{max})$, there always exists a finite k such that c_k is equal to x_{max}. That is, the k-step BRS C_k converges to the maximal RCIS $[-x_{max}, x_{max}]$ in finite steps. However, if we select $c_0 = c_d$, then $c_k = c_0 < x_{max}$ for all $k \geq 0$. That is, $C_k = C_0$ fails to converge to the maximal RCIS.

Case 2: Suppose that $|\alpha| > 1$, $u_{max} \geq d_{max}$, and $x_{max} \geq c_d \geq d_{max}$. For any $c_0 \in [d_{max}, c_d]$, $C_0 = [-c_0, c_0]$ is an RCIS. The maximal RCIS is $[-c_d, c_d]$.

According to (3.5), if we select any $c_0 \in [d_{max}, c_d)$, c_k converges to c_d in the limit. That is, the k-step BRS C_k converges to the maximal RCIS in Hausdorff distance in infinite steps. The limit $C_\infty = (-c_d, c_d)$ is the interior of the maximal RCIS. Furthermore, the Hausdorff distance between C_k and the maximal RCIS is

$$d_H(C_k, [-c_d, c_d]) = \frac{c_d - c_0}{|\alpha|^k}, \tag{3.6}$$

which decays to 0 exponentially fast.

Otherwise: For all the other cases where the maximal RCIS is not empty, the behavior of C_k is similar to Case 1. That is, C_k is either equal to C_0 for all $k \geq 0$ or converges to the maximal RCIS in finite steps.

In Example 3.1, we observe three types of limit of C_k: (i) the limit C_∞ is exactly equal to the maximal RCIS C_{max}; (ii) C_∞ is a subset of C_{max}, but the Hausdorff distance $d_H(C_\infty, C_{max}) = 0$; (iii) C_∞ is a subset of C_{max} and the Hausdorff distance $d_H(C_\infty, C_{max}) > 0$. Note that for the limit type (ii), even if $C_\infty \subset C_{max}$, the maximal RCIS C_{max} is equal to the closure of C_∞ since $d_H(C_\infty, C_{max}) = 0$. Thus, among the three types, the limit type (iii) is the least desirable one, as in this case it is

impossible to obtain the maximal RCIS from the BRSs of C_0.

A key observation in Example 3.1 that distinguishes between the limit types (i), (ii) and the limit type (iii) is that if there exists a k such that C_k contains C_0 in the interior, then C_∞ is either in type (i) or in type (ii). Indeed, in Example 3.1, the limit type (iii) only happens when $C_k = C_0$ for all $k \geq 0$. Intuitively, this observation suggests that if the BRSs of C_0 expand in all directions in $\mathbb{R}^n$, then they keep expanding until they reach the maximal RCIS (that is the limit types (i) and (ii)). In other words, the limit type (iii) occurs only if the BRSs of C_0 only expand in certain directions, or do not expand at all (which happens in Example 3.1).

In terms of the convergence rate, another interesting observation in Example 3.1 is that the Hausdorff distance between C_k and the maximal C_{max} decays to 0 at least exponentially fast for the limit types (i) and (ii). Next, we state the main result of this chapter, which shows that the observations made in Example 3.1 actually hold for any linear systems of the form (3.1):

Theorem 3.1. *Under Assumption 3.1, suppose that C_0 is a compact and convex RCIS of the system Σ in S_{xu}. If C_0 is contained in the interior of C_{k_0} for some $k_0 > 0$, then C_k converges to the maximal RCIS C_{max} in Hausdorff distance, and there exists integers $N_0 \geq 0$, $N > 0$ and scalars $c > 0$, $a \in (0,1)$ such that the Hausdorff distance between C_{N_0+kN} and C_{max} satisfies*

$$d_H(C_{N_0+kN}, C_{max}) \leq ca^k. \tag{3.7}$$

That is, the Hausdorff distance $d_H(C_{N_0+kN}, C_{max})$ decays to zero exponentially fast as k goes to infinity.

Note that the bound in (3.7) holds for indices $\{N_0 + kN\}_{k=0}^{\infty}$ increasing by an interval of N. But, due to the fact that $d_H(C_k, C_{max})$ is monotonically non-increasing over k, the inequality in (3.7) implies for all $k \geq 0$,

$$d_H(C_{N_0+k}, C_{max}) \leq c(a^{1/N})^{k-N+1} = (ca^{-1+1/N})a^{k/N}. \tag{3.8}$$

Hence, by Theorem 3.1 and (3.8) , the k-step BRS C_k converge to the maximal RCIS C_{max} in Hausdorff distance exponentially fast whenever C_0 is contained in the interior of C_{k_0} for some $k_0 > 0$. This result validates the two key observations we have in Example 3.1.

When C_0 and C_k are represented by polytopes, the condition $C_0 \subseteq int(C_k)$ can be numerically checked by a linear program[1]. Suppose that $C_0 = \{x \mid H_1 x \leq h_1\}$ and $C_k = \{x \mid H_2 x \leq h_2\}$ for some H_i, h_i in appropriate dimensions, $i = 1, 2$, and x_0 is any interior point of C_k. We construct the

[1]This is a variant of the standard polytope containment problem given the H-representations of two polytopes [104].

following linear program to check if $C_0 \subseteq int(C_k)$:

$$\begin{aligned} \gamma^* = \min_{\gamma \geq 0, \Lambda} \ & \gamma \\ \text{subject to } & \Lambda H_1 = H_2 \\ & \Lambda(h_1 - H_1 x_0) \leq \gamma(h_2 - H_2 x_0). \end{aligned} \tag{3.9}$$

According to [104, Lemma 1], γ^* in (3.9) is less than 1 if and only if $C_0 \subseteq int(C_k)$ if and only if $k_0 \leq k$. We can incorporate this linear program to Alg. 2 to obtain a convergence certificate while inner approximating C_{max}, as shown by the pseudocode in Alg. 3. If Alg. 3 terminates with `isConvergent` = `True`, we know that the inside-out algorithm converges to C_{max} exponentially fast; otherwise, it is inconclusive.

Algorithm 3 The inside-out algorithm with convergence certificate

Input: System dynamics Σ, the safe set S_{xu}, an RCIS C_{-1} of Σ in S_{xu}, and the maximal iteration number k_{max}
Initialization: $k \leftarrow -1$, `isConvergent` $\leftarrow$ `False`, $\gamma^* \leftarrow 1$
do
  $k \leftarrow k+0$
  $C_k \leftarrow Pre_\Sigma(C_{k-1}, S_{xu}, D)$
  if `isConvergent` = `False` **then**
    $\gamma^* \leftarrow$ the optimal cost of (3.9) with respect to C_k
    `isConvergent` $\leftarrow (\gamma^* < 1)$
  end if
while $C_k \neq C_{k-1}$ or $k < k_{max}$
return C_k, `isConvergent`

Next, recall that C_∞ in (3.2) is the limit of the k-step BRS C_k as k goes to infinity. When k_0 in Theorem 3.1 exists, it is obvious that C_0 is contained by the interior $int(C_\infty)$ of C_∞, since C_{k_0} is a subset of C_∞. But conversely, if $C_0 \subseteq int(C_\infty)$, does there always exist a finite k_0 such that $C_0 \subseteq int(C_{k_0})$? It turns out that those two conditions are equivalent, shown by the following theorem:

Theorem 3.2. *Under the same conditions of Theorem 3.1, C_0 is contained in the interior of C_{k_0} for some finite $k_0 > 0$ if and only if C_0 is contained in the interior of C_∞ in* (3.2).

Finally, the readers may wonder what happens if k_0 in Theorem 3.1 does not exist, or equivalently $C_0 \nsubseteq int(C_\infty)$. In Example 3.1, the limit type (iii) occurs when k_0 does not exist. However, there are examples where the limit types (i) and (ii) occur even if a finite k_0 does not exist. Thus, the existence of k_0 is only a sufficient condition for the convergence of the BRS C_k to the maximal RCIS. When the system is disturbance-free (that is $D = \{0\}$), the existence of k_0 is guaranteed if (A, B)

is asymptotically stable and $0 \in int(C_0)$, or if (A,B) is controllable and $0 \in C_0$ and $0 \in int(S_{xu})$. A more in-depth discussion for results of disturbance-free systems can be found in Section 3.5.

3.3 Proof of the Main Result

In this section, we present the main ideas in the proof of Theorem 3.1. First, according to a fixed-point theorem [27, Theorem 12][2], given a compact convex RCIS C_0 in S_{xu}, there always exists a stationary point $(x_e, u_e, d_e) \in S_{xu} \times D$ such that $x_e \in C_0$ and $Ax_e + Bu_e + Ed_e = x_e$. Without loss of generality, we assume that the stationary point (x_e, u_e, d_e) is the origin of the state-input-disturbance space and thus $0 \in C_0$ and $0 \in S_{xu} \times D$ for the remainder of this section. Also, by Theorem 3.2, we convert the condition on the existence of k_0 in Theorem 3.1 into the following equivalent assumption.

Assumption 3.2. *The set C_0 is a compact convex RCIS of Σ in S_{xu} and is contained by the interior $int(C_\infty)$ of the infinite-step BRS C_∞ in* (3.2)*.*

Based on Assumption 3.2, the proof of Theorem 3.1 contains two steps: The first step is to show that the closure $cl(C_\infty)$ of the limit C_∞ is equal to the maximal RCIS C_{max}, that is to prove the following theorem:

Theorem 3.3. *Under Assumptions 3.1 and 3.2, the closure of C_∞ in* (3.2) *is the maximal RCIS C_{max}, that is $cl(C_\infty) = C_{max}$.*

Sketch proof. Note that the maximal RCIS C_{max} is bounded due to Assumption 3.1. Since $C_0 \subseteq int(C_\infty)$ and C_{max} is bounded, intuitively, we can find a small enough $\alpha > 0$ such that the set $C(\alpha) = (1-\alpha)C_0 + \alpha C_{max}$ is contained in $int(C_\infty)$. Due to the linearity of the system, it can be shown that the infinite-step BRS of $C(\alpha)$ contains $(1-\alpha)C_\infty + \alpha C_{max}$. Then, the key to prove Theorem 3.3 is to realize that for any set $C' \subseteq int(C_\infty)$, the infinite-step BRS of C' is contained by C_∞. Thus, for some $\alpha > 0$, we have $(1-\alpha)C_\infty + \alpha C_{max} \subseteq C_\infty \subseteq C_{max}$. That implies $C_{max} = cl(C_\infty)$. The complete proof can be found in Appendix A. □

The second step is to show that the Hausdorff distance between C_k and and the maximal RCIS C_{max} decays to 0 exponentially fast when $C_0 \subseteq int(C_\infty)$. We first introduce an extended notion of λ-contractive sets [22]:

Definition 3.1. Given a scalar $\lambda > 0$, a set X is called *k-step λ-contractive* if the k-step BRS of λX contains the set X, that is $Pre^k_\Sigma(\lambda X, S_{xu}, D) \supseteq X$.

[2] See Appendix A.1 for a detailed discussion.

By definition, a set X is k-step λ-contractive if the system can go from any state in X to some state in λX in k steps without violating any safety constraints. Under Assumptions 3.1 and 3.2, the closure of the infinite-step BRS $cl(C_\infty)$ always contains a N-step λ-contractive set, shown by the following theorem.

Theorem 3.4. *Under Assumptions 3.1 and 3.2, there exist an integer $N > 0$ and scalars $\gamma \in (0,1]$, $\lambda \in [0,1)$ such that $\gamma cl(C_\infty)$ is N-step λ-contractive. Furthermore, there exists a finite integer $N_0 \geq 0$ such that $Pre_\Sigma^{N_0}(C_0, S_{xu}, D) \supseteq \lambda\gamma cl(C_\infty)$.*

Sketch proof. Since $C_0 \subseteq int(C_\infty)$ and C_∞ is convex, there exists positive scalars β_0 and β_1, with $0 < \beta_0 < \beta_1 < 1$, such that $C_0 \subseteq \beta_0 cl(C_\infty) \subset \beta_1 cl(C_\infty) \subseteq int(C_\infty)$. Since $\beta_1 cl(C_\infty) \subseteq int(C_\infty)$ and the k-step BRS of C_0 converges to C_∞, it can be proven that there exists a finite N such that $\beta_1 cl(C_\infty) \subseteq C_N$. Since $C_0 \subseteq \beta_0 cl(C_\infty)$, C_N is contained by the N-step BRS of $\beta_0 cl(C_\infty)$. That implies $\beta_1 cl(C_\infty) \subseteq C_N$ is contained in the N-step BRS of $\beta_0 cl(C_\infty)$, and thus is a N-step (β_0/β_1)-contractive set. By assigning $\gamma = \beta_1$, $\lambda = \beta_0/\beta_1$ and $N_0 = N$, the statement in Theorem 3.4 is proven. A complete proof can be found in Appendix A. □

By Theorems 3.3 and 3.4, we show that $\gamma cl(C_\infty) = \gamma C_{max}$ is N-step λ-contractive. Note that a k-step λ-contractive set X is not necessarily an RCIS unless $k = 1$ and $0 \in X$. Thus, γC_{max} may not be an RCIS.

Recall that our goal in the second step of proving Theorem 3.1 is to show the convergence rate of the k-step BRS C_k to C_{max}. So how is γC_{max} being k-step λ-contractive set related to the convergence rate of BRSs? Let C be an RCIS of Σ in S_{xu}. It turns out that if there exists a factor $\gamma \in (0,1]$ such that the scaled set γC is N-step λ-contractive for some N and $\lambda \in [0,1)$, then the k-step BRS of $\lambda\gamma C$ approaches to C exponentially fast as k increases. The proof of this statement is enabled by the following theorem.

Theorem 3.5. *Under Assumption 3.1, for any convex RCIS C of Σ in S_{xu}, suppose that there exist $\gamma \in (0,1)$, N and $\lambda \in [0,1)$ such that γC is N-step λ-contractive, that is*

$$Pre_\Sigma^N(\lambda\gamma C, S_{xu}, D) \supseteq \gamma C. \tag{3.10}$$

Then, for any scalar ξ with $1 > \xi \geq \lambda\gamma$,

$$Pre_\Sigma^N(\xi C, S_{xu}, D) \supseteq g(\xi) C, \tag{3.11}$$

where

$$g(\xi) = \frac{1-\gamma}{1-\lambda\gamma}\xi + \frac{(1-\lambda)\gamma}{1-\lambda\gamma} \geq \xi. \tag{3.12}$$

For any C satisfying the conditions in Theorem 3.5, (3.11) implies that the N-step BRS of $\lambda\gamma C$ expands at least to $g(\lambda\gamma)C \supseteq \lambda\gamma C$. By applying Theorem 3.5 k times, we obtain that

$$Pre_{\Sigma}^{kN}(\lambda\gamma C, S_{xu}, D) \supseteq g^k(\lambda\gamma)C, \tag{3.13}$$

where $g^k(\cdot) = g(g(...))$ is the function that composes $g(\cdot)$ for k times. According to Theorems 3.3 and 3.4, C in (3.13) can be replaced by C_{max}. That is, for γ, N and λ in Theorem 3.4,

$$Pre_{\Sigma}^{kN}(\lambda\gamma C_{max}, S_{xu}, D) \supseteq g^k(\lambda\gamma)C_{max}. \tag{3.14}$$

By Theorem 3.4, there exists N_0 such that the N_0-step BRS C_{N_0} of C_0 contains $\lambda\gamma C_{max}$. Then, by (3.14),

$$C_{max} \supseteq C_{N_0+kN} \supseteq Pre_{\Sigma}^{kN}(\lambda\gamma C_{max}, S_{xu}, D) \supseteq g^k(\lambda\gamma)C_{max}. \tag{3.15}$$

The inclusion relation in (3.15) implies that the Hausdorff distance between C_{N_0+kN} and the maximal RCIS C_{max} is bounded by

$$d_H(C_{N_0+kN}, C_{max}) \leq (1 - g^k(\lambda\gamma)) \sup_{x \in C_{max}} \|x\|_2. \tag{3.16}$$

Since $g(\cdot)$ in (3.12) is affine, it can be shown that

$$1 - g^k(\lambda\gamma) = \left(\frac{1-\gamma}{1-\lambda\gamma}\right)^k (1 - \lambda\gamma). \tag{3.17}$$

Combining (3.16) and (3.17), we have

$$d_H(C_{N_0+kN}, C_{max}) \leq ca^k,$$

where $c = (1-\lambda\gamma)\sup_{x \in C_{max}} \|x\|_2$ and $a = (1-\gamma)/(1-\lambda\gamma)$. With $\gamma \in (0,1]$, $\lambda \in [0,1)$, it is easy to check that $a \in [0,1)$. Since C_{max} is bounded, c is finite. Thus, the Hausdorff distance between C_{N_0+kN} and C_{max} decays to zero exponentially fast as k goes to infinity. That completes the proof of Theorem 3.1.

3.4 Numerical Example

Consider the two-dimensional system Σ

$$\Sigma : x(t+1) = \begin{bmatrix} 1.1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 1 \\ 1 \end{bmatrix} d, \tag{3.18}$$

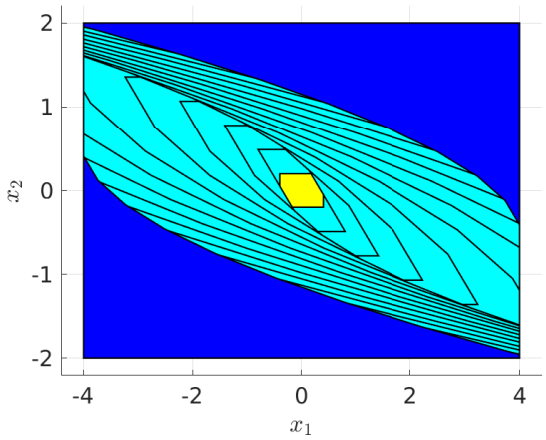
with $x \in \mathbb{R}^2$, $u \in \mathbb{R}$ and $d \in D = [-0.01, 0.01]$. The safe set of the system is $S_{xu} = S_x \times U$, with $S_x = [-4,4] \times [-2,2]$ and $U = [-0.3, 0.3]$. Denote the maximal RCIS of Σ in S_{xu} by C_{max}. The set C_0 is selected to be the maximal RCIS in a scaled safe set $\tilde{S}_{xu} = (0.1S_x) \times U$, shown by the yellow polytope in Fig. 3.1a.

MPT3 toolbox [48] and YALMIP [84], equipped with GUROBI 9.5.0 [44], are used to implement the computation of the BRS in (2.10) and the linear program in (3.9). The computed k-step BRSs C_k of C_0 for $k = 1, \cdots, 16$ are visualized in Fig. 3.1a, which converges to the maximal C_{max} in 16 steps. By solving the linear program in (3.9), we checked that C_0 is contained in the interior of C_2. Thus, Theorem 3.1 implies that the k-step BRS C_k converges to the maximal RCIS C_{max} at least exponential fast. The Hausdorff distance between C_k and C_{max} is shown by the red curve in Fig. 3.1b. We also manually fit an exponential function $y(k) = 4.61 \times 0.8^k$ (the blue curve in Fig. 3.1b) that bounds the actual Hausdorff distance from above. The existence of this exponential decaying upper bound is predicted by Theorem 3.1. Given any system and C_0, how to find such an exponential decaying function without computing all the BRSs would be part of our future work.

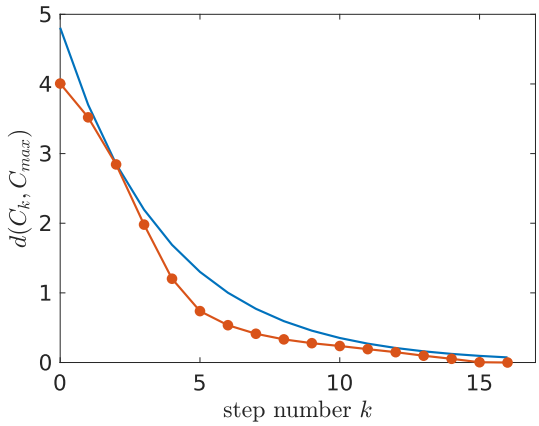
3.5 Discussion

In this section, we compare our result with the existing ones in [32, 31, 46, 30]. Here we adopt the notation in [32, 31] and call a set X a *C-set* if X is convex, compact and contains 0 in the interior. Note that the results in [32, 31, 46, 30] are all based on the condition that the disturbance set $D = \{0\}$ and S_{xu} is a C-set. Hence, we assume that the above condition holds for the remainder of this section.

First, [32, 31, 46] identify the two sufficient conditions under which the k-step BRS C_k converges to the maximal CIS in Hausdorff distance. The first sufficient condition [46, 32, 31] is that (A,B) is controllable and $C_0 = \{0\}$; the second one [32, Proposition 30] is that (A,B) is asymptotically stabilizable and C_0 is a controlled invariant C-set. When the first sufficient condition is satisfied, [32, 31] show that $C_0 = \{0\}$ also satisfies the conditions in Theorem 3.1. When the second sufficient condition is satisfied, the corresponding C_0 may not satisfy the condition in Theorem 3.1. However, the proof of [32, Proposition 30] shows that C_0 must contain a smaller CIS that satisfies the conditions in Theorem 3.1. Thus, both of the sufficient conditions are corollaries of our result.



(a) The k-step BRS of C_0 over k



(b) The Hausdorff distance $d_H(C_k, C_{max})$ over k

Figure 3.1: Left: The k-step BRS C_k for $k = 1, \cdots, 16$ are the nested cyan polytopes, where a larger one corresponds to a larger k. The set C_0 is the yellow polytope in the middle. The maximal RCIS C_{max} is equal to the largest cyan polytope C_{16}. The dark blue rectangle is the safe set S_{xu}. Right: The Hausdorff distance $d_H(C_k, C_{max})$ between the k-step BRS C_k and the maximal RCIS C_{max} versus the number of steps k (the red curve with dots). An exponential function $y(k) = 4.61 \times 0.8^k$ that bounds the Hausdorff distance from above is manually fitted (the blue curve).

Though [32, 31, 46] prove the convergence of C_k, none of those works show the convergence rate of C_k. To our best knowledge, [30] is the only work that shows the convergence rate of C_k, under the condition that (A, B) is controllable and $C_0 = \{0\}$ (the first sufficient condition above). Originally [30] derives the convergence rate in a different metric. But by converting the metric in [30] to Hausdorff distance, we can show that the convergence rate of C_k in [30] is equivalent to the exponential convergence in Hausdorff distance proven in this chapter. That is, under the first sufficient condition in the previous paragraph, our result coincides with the result in [30].

3.6 Conclusion

In this chapter, we consider linear systems with disturbance and show a sufficient condition under which the k-step BRS of RCISs converges to the maximal RCIS exponentially fast. When all sets are represented by polytopes, this sufficient condition can be numerically checked via the linear program in (3.9). When restricted to disturbance-free systems, our result implies the existing results in [32, 31, 46, 30].

In terms of applications, our result provides convergence guarantees for inside-out algorithm (and its variants) [46, 32, 103, 36, 119, 8], and sheds some light on novel analysis for RCIS-based control synthesis algorithms. For instance, in constrained MPC equipped with a controlled invariant

terminal set [23], our results imply that under mild conditions, the DoA of MPC enlarges to the maximal DoA exponentially fast as the prediction horizon T increases, which provides new insights for selecting the prediction horizon T. Moreover, if parameters N_0, N, a and c in Theorem 3.1 are known, we can quantitatively evaluate the Hausdorff distance between the DoA of MPC with respect to any given T and the maximal DoA via (3.7). Algorithms for estimating parameters N_0, N, a in Theorem 3.1 for nominal systems are studied in Chapter 7.

CHAPTER 4

Controlled Invariant Sets: Implicit Closed-form Representations and Applications

The iterative algorithms for computing RCISs introduced in Section 2.5 are known to suffer from poor scaling with the system's dimension and no guarantees of termination. An alternative approach is to construct an implicit representation for an RCIS. The specific implicit representation used in this chapter is a set in the higher dimensional space of states and finite input sequences. We argue that in many practical, safety-critical applications, such as MPC and supervisory control, knowledge of the explicit RCIS is not required and the implicit representation suffices. Consequently, by exploiting the efficiency of the implicit representation the aforementioned ideas are suitable for systems with large dimensions.

In this chapter, we propose a general framework for *computing (implicit) RCISs* for discrete-time linear systems with additive disturbances, under polytopic state-input constraints. We consider RCISs parameterized by collections of *eventually periodic* input sequences and prove that this choice leads to a closed-form expression for an implicit RCIS in the space of states and finite input sequences. Moreover, this choice results in a systematic way to obtain larger RCISs, which we term a *hierarchy*. Essentially, the computed sets include all states for which there exist eventually periodic input sequences that lead to a trajectory that remains within the safe set indefinitely. Once the (implicit) RCIS is computed, any controller rendering the RCIS invariant can be used in practice and a fixed periodic input is not chosen or used. Moreover, we show that this parameterization is rich enough, such that: 1) in the absence of disturbances, our method is complete and sufficient to approximate the maximal CIS arbitrarily well; 2) in the presence of disturbances, a weak completeness result is established, along with a bound for the computed RCIS that can be approximated arbitrarily well. Finally we study, both theoretically and experimentally, safety-critical scenarios and establish that the efficient implicit representation suffices in place of computing the exact RCIS. In practice, the use of implicit RCISs can be done via optimization programs, e.g., a Linear Program (LP), a Mixed-Integer (MI) program, or a Quadratic Program (QP), and is only limited by the size of the program afforded to solve.

Chapter Overview. We state the problem in Section 4.1, and then provide a closed-form construction of implicit RCISs in Section 4.2. Section 4.3 shows how to systematically expand the implicit RCIS to reduce conservativeness, followed by usages of the resulting implicit RCIS in planning and control in Section 4.4. In Section 4.5, we derive a performance bound of our approach, showing that the implicit RCIS converges to a maximal set exponentially fast with the parameterization size. Finally, we demonstrate the effectiveness of the proposed method via several case studies in Section 4.6 and conclude this chapter in Section 4.7.

Notation. The Hausdorff distance between two subsets P and Q of $\mathbb{R}^n$ is denoted by $d(P,Q)$ and is induced from the Euclidean norm in $\mathbb{R}^n$. We denote a block-diagonal matrix M with blocks $M_1,\dots,M_N$ by $M=\text{blkdiag}(M_1,\dots,M_N)$. For any $N\in\mathbb{Z}^+$, let $[N]=\{1,2,\cdots,N\}$. Let $\mathbb{I}$ and $\mathbf{0}$ be the identity and zero matrices of appropriate sizes respectively, while $\mathbf{1}$ is a vector with all entries equal to 1.

4.1 Problem Setup

In this chapter, we consider a discrete-time linear system Σ

$$x^+ = Ax+Bu+Ew, \tag{4.1}$$

with state $x\in\mathbb{R}^n$, input $u\in\mathbb{R}^m$, and disturbance $w\in W\subseteq\mathbb{R}^d$. Let $S_{xu}\subseteq\mathbb{R}^{n+m}$ be the safe set of the system Σ.

Assumption 4.1. *In this chapter, we focus on systems and safe sets that satisfy the following:*
1) There exists a suitable state feedback transformation that makes the matrix A of system Σ nilpotent. For a nilpotent matrix, there exists a $\nu\in\mathbb{Z}^+$ such that $A^\nu=0$.
2) The safe set *$S_{xu}\subset\mathbb{R}^n\times\mathbb{R}^m$ and the* disturbance set *$W\subset\mathbb{R}^d$ are both polytopes.*

Remark 4.1. For any controllable system Σ, there exists a state feedback transformation satisfying Assumption 4.1 [11, Ch.3]. In this case, the nilpotency index ν is equal to the largest controllability index of Σ.

For any system Σ satisfying Assumption 4.1, let $K\in\mathbb{R}^{m\times n}$ be the feedback gain such that $A+BK$ is nilpotent. We construct a system Σ' by pre-feedbacking Σ with $u=Kx+u'$:

$$x^+ = (A+BK)x+Bu'+Ew,$$

where $u'\in\mathbb{R}^m$ is the input of the system Σ'. The safe set for Σ' is the polytope induced from the safe set S_{xu} of Σ as $S'_{xu}=\{(x,u')\in\mathbb{R}^n\times\mathbb{R}^m \mid (x,Kx+u')\in S_{xu}\}$.

Proposition 4.1. Any RCIS $\mathscr{C}$ of Σ with respect to S_{xu} is an RCIS of Σ' with respect to S'_{xu} and vice versa.

Proof. The proof is based on the fact that the map from (x,u) to $(x,u') = (x,u-Kx)$ is a bijection from S_{xu} to S'_{xu}.

Consider an RCIS $\mathscr{C}$ of Σ with respect to S_{xu}, and (x,u) with $x \in \mathscr{C}$ and $Ax+Bu+EW \subseteq \mathscr{C}$. Take $u' = u-Kx$. Then, $(x,u') \in S'_{xu}$ since $(x,Kx+u') = (x,u) \in S_{xu}$. Advancing the state x with input u' in Σ' gives $(A+BK)x+Bu'+EW = Ax+BKx-BKx+Bu+EW = Ax+Bu+EW \subseteq \mathscr{C}$. Hence, $\mathscr{C}$ is also an RCIS for Σ' with respect to S'_{xu}. The other direction is shown in a similar way. □

Based on Proposition 4.1, it can be seen that the problem of finding an RCIS of Σ with respect to S_{xu} is *exactly equivalent* to the problem of finding an RCIS of Σ' with respect to S'_{xv}. That is, for any procedure that takes in (Σ, S_{xu}) and produces a RCIS $\mathscr{C}$, there exists an equivalent procedure that takes in (Σ', S'_{xu}) and produces the same RCIS $\mathscr{C}$, and vice versa. Therefore, in the remainder of this chapter, we simply assume that the system in (4.1) (and its safe set S_{xu}) is already transformed to this equivalent form where the matrix A is nilpotent.

The main goal of this chapter is to compute an *implicit representation of an RCIS in closed-form*. Hereafter, we refer to this representation as the *implicit RCIS*.

Definition 4.1 (Implicit RCIS)**.** Given a system Σ, a safe set $S_{xu} \subset \mathbb{R}^n \times \mathbb{R}^m$, and some integer $q \in \mathbb{Z}^+$, a set $\mathscr{C}_{xv} \subseteq \mathbb{R}^n \times \mathbb{R}^q$ is an *Implicit RCIS* for Σ if its projection $\pi_{[1:n]}(\mathscr{C}_{xv})$ onto the first n dimensions is an RCIS for Σ with respect to S_{xu}.

The following result stems directly from Definition 2.3.

Proposition 4.2. The union of RCISs and the convex hull of an RCIS are robustly controlled invariant.

We define the *accumulated disturbance set* at time t by:

$$\overline{W}_t = \sum_{i=1}^{t} A^{i-1} EW. \tag{4.2}$$

By nilpotency of A we have that:

$$\overline{W}_\infty = \sum_{i=1}^{\infty} A^{i-1} EW = \sum_{i=1}^{\nu} A^{i-1} EW. \tag{4.3}$$

In the literature, $\overline{W}_\infty$ is called the *Minimal RPIS* of the system $x^+ = Ax+Ew$ [99].

The next operator is used throughout this chapter.

Definition 4.2 (Reachable set)**.** Given a system Σ and a set $X \subset \mathbb{R}^n$, define the *reachable set* from X under input sequence $\{u_i\}_{i=0}^{t-1}$ as:

$$\mathscr{R}_\Sigma\big(X, \{u_i\}_{i=0}^{t-1}\big) = A^t X + \sum_{i=1}^{t} A^{i-1} B u_{t-i} + \overline{W}_t. \tag{4.4}$$

Intuitively, $\mathscr{R}_\Sigma\big(X, \{u_i\}_{i=0}^{t-1}\big)$ maps a set X and an input sequence $\{u_i\}_{i=0}^{t-1}$ to the set of all states that can be reached from X in t steps when applying said input sequence. Conventionally, $\mathscr{R}_\Sigma(X, \emptyset) = X$ and when X is a singleton, i.e., $X = \{x\}$, we abuse notation to write $\mathscr{R}_\Sigma\big(x, \{u_i\}_{i=0}^{t-1}\big)$.

4.2 Implicit Representation of RCISs for Linear Systems

The classical algorithm that computes the *maximal* RCIS consists of an iterative procedure [20, 34] and theoretically works for any discrete-time system and safe set. However, this approach is known to suffer from the curse of dimensionality and its termination is not guaranteed. To alleviate these drawbacks, we propose an algorithm that is guaranteed to terminate and computes an implicit RCIS efficiently in closed-form, thus being suitable for high dimensional systems. Moreover, by optionally projecting the implicit RCIS back to the original state-space one computes an explicit RCIS. Overall, the proposed algorithm computes controlled invariant sets in one and two moves respectively.

The goal of this section is to present a *finite implicit representation* of an RCIS. That is, we provide a *closed-form* expression for an *implicit RCIS* characterized by constraints on the state and on a finite input sequence, whose length is the design parameter. This results in a polytopic RCIS in a higher dimensional space. Intuitively, the implicit RCIS contains the pairs of states and appropriate finite input sequences that guarantee that the state remains in the safe set indefinitely.

4.2.1 General implicit robust controlled invariant sets

We begin by discussing a general construction of a polytopic implicit RCIS. First, we consider inputs u_t to Σ that evolve as the output of a linear dynamical system, Σ_C, whose state is a *sequence of q inputs, v*, i.e.:

$$\Sigma_C \quad : \quad \begin{aligned} v_{t+1} &= P v_t, \\ u_t &= H v_t, \end{aligned} \tag{4.5}$$

where $v \in \mathbb{R}^{mq}$, $P \in \mathbb{R}^{mq \times mq}$, and $H \in \mathbb{R}^{m \times mq}$. The choice of a linear dynamical system stems from our safe set being a polytope per Assumption 4.1. By using system Σ_C we preserve the linearity

of the safe set constraints and we are, hence, able to compute polytopic RCISs with respect to polytopic safe sets. The resulting input to Σ can be expressed as:

$$u_t = Hv_t = HP^t v_0, \tag{4.6}$$

for an initial choice of $v_0 \in \mathbb{R}^{mq}$. We can then lift system Σ, after closing the loop with Σ_C, to the following companion dynamical system:

$$\Sigma_{xv} \quad : \quad \begin{bmatrix} x^+ \\ v^+ \end{bmatrix} = \begin{bmatrix} A & BH \\ \mathbf{0} & P \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} E \\ \mathbf{0} \end{bmatrix} w. \tag{4.7}$$

Given the safe set S_{xu}, we construct the companion safe set $S_{xv} = \{(x,v) \in \mathbb{R}^n \times \mathbb{R}^{mq} \mid (x, Hv) \in S_{xu}\}$. The companion system of (4.1) is the closed-loop dynamics of (4.1) with a control input in (4.6). Then, the companion safe set simply constrains the closed-loop state-input pairs in the original safe set, i.e., $(x_t, Hv_t) \in S_{xu}$.

Theorem 4.3 (Generalized implicit RCIS)**.** *Let $\mathscr{C}_{xv}$ be an RPIS of the companion system Σ_{xv} with respect to the companion safe set S_{xv}. The projection of $\mathscr{C}_{xv}$ onto the first n coordinates, $\pi_{[1:n]}(\mathscr{C}_{xv})$, is an RCIS of the original system Σ with respect to S_{xu}. In other words, $\mathscr{C}_{xv}$ is* an implicit RCIS *of Σ.*

Proof. Let $x \in \pi_{[1:n]}(\mathscr{C}_{xv})$. Then, there exists a $v \in \mathbb{R}^{mq}$ such that $(x,v) \in \mathscr{C}_{xv}$. Define $u = Hv$ and pick an arbitrary $w \in W$. By construction of S_{xv}, $(x,u) \in S_{xu}$. Since $\mathscr{C}_{xv}$ is an RPIS, we have that $(x^+, v^+) = (Ax + Bu + Ew, Pv) \in \mathscr{C}_{xv}$ and thus $x^+ \in \pi_{[1:n]}(\mathscr{C}_{xv})$. By Definition 2.3, $\pi_{[1:n]}(\mathscr{C}_{xv})$ is an RCIS of Σ in S_{xu}. □

In principle, Theorem 4.3 holds even if Σ_C is nonlinear. However, the choice of a linear system Σ_C, as in (4.5), makes the computation of the maximal RPIS of Σ_{xv} more efficient. In what follows, we study the conditions on P and H such that the maximal RPIS of Σ_{xv} is represented in closed-form.

4.2.2 Finite reachability constraints

By definition of the companion safe set S_{xv} and Definition 2.5, we have that any state (x,v) belongs to the maximal RPIS of Σ_{xv} with respect to S_{xv}, if and only if, the input sequence $\{u_i\}_{i=0}^{t-1}$, with each input as in (4.6), satisfies:

$$\left(\mathscr{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu},\ t \geq 0, \tag{4.8}$$

where $\mathscr{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right) \subseteq \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, and the pair $\left(\mathscr{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq \mathbb{R}^n \times \mathbb{R}^m$. By Theorem 4.3, the above constraints characterize the states and input sequences within an implicit RCIS of

Σ, such that the pair (x,u) stays inside the safe set S_{xu} indefinitely. Notice that (4.8) defines an *infinite* number of constraints in general. In this section, we investigate under what conditions we can reduce the above constraints into a *finite* number and compute them explicitly. Then, we use these constraints to construct the promised implicit RCIS.

Definition 4.3 (Eventually periodic behavior)**.** Consider two integers $\tau \in \mathbb{Z}^+ \cup \{0\}$ and $\lambda \in \mathbb{Z}^+$. A control input u_t follows an *eventually periodic behavior* if:

$$u_{t+\lambda} = u_t, \text{ for all } t \geq \tau. \tag{4.9}$$

We call τ the *transient* and λ the *period*.

Proposition 4.4 (Finite reachability constraints). Consider a system Σ satisfying Assumption 4.1. If the input u_t follows an eventually periodic behavior with transient $\tau \in \mathbb{Z}^+ \cup \{0\}$ and period $\lambda \in \mathbb{Z}^+$, then the infinite constraints in (4.8) are reduced to a finite number of constraints.

Proof. Under Assumption 1 the matrix A is nilpotent with nilpotency index ν. Consequently, given (4.4), the reachable set from a state x for $t \geq \nu$ depends only on the past ν inputs. We abuse notation to write $\mathscr{R}_\Sigma\big(\{u_i\}_{i=0}^{t-1}\big)$ and omit the state x to denote dependency only on the inputs. Then, for $t \geq \nu + \tau$:

$$\mathscr{R}_\Sigma\big(\{u_i\}_{i=0}^{t-1}\big) = \sum_{i=1}^{\nu} A^{i-1}Bu_{t-i} + \overline{W}_\infty \overset{(4.9)}{=} \sum_{i=1}^{\nu} A^{i-1}Bu_{t+\lambda-i} + \overline{W}_\infty = \mathscr{R}_\Sigma\Big(\{u_i\}_{i=0}^{t+\lambda-1}\Big).$$

Therefore, under inputs with eventually periodic behavior the reachability constraints repeat themselves after $t = \nu + \tau + \lambda$. As a result, we can split the constraints in (4.8) as:

$$\Big(\mathscr{R}_\Sigma\Big(x, \{u_i\}_{i=0}^{t-1}\Big), u_t\Big) \subseteq S_{xu},\ t = 0, \ldots, \nu - 1, \tag{4.10}$$

$$\Big(\mathscr{R}_\Sigma\Big(\{u_i\}_{i=0}^{t-1}\Big), u_t\Big) \subseteq S_{xu},\ t = \nu, \ldots, \nu + \tau + \lambda - 1. \tag{4.11}$$

The above suggests that $\big(\mathscr{R}_\Sigma\big(x, \{u_i\}_{i=0}^{t-1}\big), u_t\big) \subseteq S_{xu}$ for all $t \geq 0$ can be replaced with only $\nu + \tau + \lambda$ constraints. □

Proposition 4.4 provides a finite representation of the constraints in (4.8) under the eventually periodic input behavior in (4.9). The next question we address concerns characterizing the classes of policies that guarantee the behavior in (4.9).

4.2.3 Implicit robust controlled invariant sets in closed-form

Recall that our goal is to derive a closed-form expression for an implicit RCIS of Σ, which is essentially the maximal RPIS of the companion system Σ_{xv} by Theorem 4.3. So far we proved that, in general, inputs with eventually periodic behavior result in finite reachability constraints. Clearly, the parameterized input in (4.6) follows an eventually periodic behavior as in (4.9) if:

$$P^t = P^{t+\lambda},\ t \geq \tau, \tag{4.12}$$

i.e., P is an *eventually periodic* matrix with transient τ and period λ.

Proposition 4.5 (Structure of eventually periodic matrices). Any *eventually periodic* matrix $P \in \mathbb{R}^{n\times n}$ has eigenvalues that are either 0 or λ-th roots of unity. If $\tau \neq 0$, i.e., P is not purely periodic, then P has at least one 0 eigenvalue with algebraic multiplicity equal to τ and geometric multiplicity equal to 1. If $P^\tau \neq 0$, i.e., P is not nilpotent, then P has at least one eigenvalue that is a λ-th root of unity.

Proof. Let $\mathrm{v} \neq 0$ be an eigenvector of P and δ the corresponding eigenvalue, i.e., $P\mathrm{v} = \delta\mathrm{v}$. Then, (4.12) for $t \geq \tau$ yields:

$$P^t = P^{t+\lambda} \Rightarrow P^t\mathrm{v} = P^{t+\lambda}\mathrm{v} \Leftrightarrow \delta^t\mathrm{v} = \delta^{t+\lambda}\mathrm{v} \overset{\mathrm{v}\neq 0}{\Leftrightarrow} \delta^t = \delta^{t+\lambda} \Leftrightarrow \delta^t\left(1-\delta^\lambda\right) = 0,$$

that is, the eigenvalues δ of P are only 0 or λ-th roots of unity.

Consider now the Jordan normal form $P = MJM^{-1}$ [64]. This form is unique up to the order of the Jordan blocks, and $P^t = MJ^tM^{-1}$. Without loss of generality, we write:

$$J = \begin{bmatrix} J_1 & \mathbf{0} \\ \mathbf{0} & J_2 \end{bmatrix},$$

where J_1 is the Jordan block corresponding to the eigenvalues of P that are 0, and J_2 is the Jordan block corresponding to the eigenvalues of P that are the λ-th roots of unity. Thus, J_1 is nilpotent. Then, when $\tau \neq 0$, equality (4.12) is equivalent to:

$$P^t = P^{t+\lambda} \Leftrightarrow MJ^tM^{-1} = MJ^{t+\lambda}M^{-1},\ t \geq \tau.$$

Matrix J_1 vanishes in exactly τ steps, i.e., $J_1^\tau = 0$ and $J_1^t \neq 0$, for $t < \tau$. This implies that P has at least one 0 eigenvalue with algebraic multiplicity equal to τ and geometric multiplicity equal to 1, but no 0 eigenvalues of geometric multiplicity 1 and algebraic multiplicity greater than τ.

Moreover, when P is not nilpotent, i.e., $P^\tau \neq \mathbf{0}$, for $t \geq \tau$:

$$J^t = J^{t+\lambda} \overset{J_1^t=\mathbf{0}, t\geq\tau}{\Leftrightarrow} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & J_2^t \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & J_2^{t+\lambda} \end{bmatrix} \Leftrightarrow J_2^t = J_2^{t+\lambda}.$$

Thus, P has at least one eigenvalue that is a λ-th root of unity. □

Corollary 4.5.1. The class of matrices described by Proposition 4.5 that satisfies (4.12) can be written, up to a similarity transformation, in the following form:

$$P = \begin{bmatrix} N & Q \\ \mathbf{0} & R \end{bmatrix}, \tag{4.13}$$

where N is a nilpotent matrix with nilpotency index τ, R is a matrix whose eigenvalues are all λ-th roots of unity, i.e., $R^\lambda = \mathbb{I}$, and Q is an arbitrary matrix.

Proposition 4.5 and Corollary 4.5.1 guide the designer to effortlessly select matrix P via its eigenvalues or its submatrices. Moreover, it is reasonable to select the projection matrix H to be *surjective* in order to obtain a non-trivial input in (4.6).

We now show that we can compute the desired *closed-form* expression for an implicit RCIS parameterized by collections of *eventually periodic* input sequences.

Theorem 4.6 (Closed-form implicit RCIS)**.** *Consider a system* Σ *and a safe set* S_{xu} *for which Assumption 4.1 holds. Select an eventually periodic matrix* $P \in \mathbb{R}^{mq\times mq}$ *and a surjective projection matrix* $H \in \mathbb{R}^{m\times mq}$*. An* implicit RCIS *for* Σ *with respect to* S_{xu}*, denoted by* $\mathscr{C}_{xv}$*, is defined by the constraints:*

$$\begin{aligned} \left(A^t x + \sum_{i=1}^{t} A^{i-1} BHP^{t-i} v, HP^t v\right) &\subseteq S_{xu} - \overline{W}_t \times \{0\},\ t = 0,\dots,\nu-1, \\ \left(\sum_{i=1}^{\nu} A^{i-1} BHP^{t-i} v, HP^t v\right) &\subseteq S_{xu} - \overline{W}_\infty \times \{0\},\ t = \nu,\dots,\nu+\tau+\lambda-1. \end{aligned} \tag{4.14}$$

That is, the set $\mathscr{C}_{xv} \subset \mathbb{R}^n \times \mathbb{R}^{mq}$*:*

$$\mathscr{C}_{xv} = \left\{(x,v) \in \mathbb{R}^n \times \mathbb{R}^{mq} \mid (x,v) \textit{ satisfy } (4.14)\right\}, \tag{4.15}$$

is computed in closed-form. Moreover, $\mathscr{C}_{xv}$ *is the maximal RPIS of the companion dynamical system in* (4.7)*.*

Proof. By Proposition 4.4, the set $\mathscr{C}_{xv}$ defined by (4.14) in closed-form satisfies the constraints in (4.8) and, thus, is the maximal RPIS of the companion system Σ_{xv} in S_{xv}. Then, by Theorem 4.3,

C_{xv} is an implicit RCIS of Σ in S_{xu}. □

Theorem 4.6 provides an implicit RCIS, $\mathscr{C}_{xv}$, in closed-form. This set defines pairs of states and finite input sequences such that the state remains in the safe set indefinitely.

Remark 4.2 (On the choice of input behavior). Notice that the open-loop eventually periodic policy used to parameterize the implicit RCIS is only a means towards its computation in closed-form. In practice, after computing an RCIS, we can use any controller appropriate for the task at hand. This is illustrated in our case studies in Section 4.6, where the controller of the system is independent of the RCIS implicit representation. For instance, once an RCIS is available one defines a closed-loop non-periodic and memoryless controller $K : \mathbb{R}^n \to \mathbb{R}^m$ for which $Ax + BK(x)$ belongs to the RCIS when x is an element of the RCIS.

Corollary 4.6.1 (Computation of explicit RCIS). By selecting an eventually periodic matrix $P \in \mathbb{R}^{mq \times mq}$ and a projection matrix $H \in \mathbb{R}^{m \times mq}$, one computes an *explicit* RCIS $C_x = \pi_{[1:n]}(\mathscr{C}_{xv})$ with a single projection step.

The size of the lifted space leads to a trade-off: on the one hand it can result to larger RCISs, as we detail in the next section, but on the other it requires more effort if the optional projection step is taken.

4.3 A Hierarchy of Controlled Invariant Sets

Our main result, Theorem 4.6, provides a closed-form expression for an implicit RCIS, $\mathscr{C}_{xv}$, with constraints on the state of the system, x, and on a finite sequence of inputs, v. The resulting sets depend on the choice of the eventually periodic matrix P in (4.5) and the projection matrix H.

In this section, we show how to *systematically* construct a sequence of RCISs that form a *hierarchy*, i.e., a non-decreasing sequence. Our goal is to provide a closed-form expression for the implicit RCISs corresponding to this hierarchy. Towards this, we identify special forms of matrices P and H.

Definition 4.4 ((τ, λ)-lasso sequence)**.** Consider two integers $\tau \in \mathbb{Z}^+ \cup \{0\}$ and $\lambda \in \mathbb{Z}^+$, and let $q = \tau + \lambda$. The control input u generated by the dynamical system Σ_C in (4.5) forms a (τ, λ)-*lasso sequence* with respect to the inputs v, if:

$$\begin{aligned} P = P_{(\tau,\lambda)} &= \text{blkdiag}\,(\bar{P}, \ldots, \bar{P}) \in \mathbb{R}^{mq \times mq}, \\ H = H_{(\tau,\lambda)} &= \text{blkdiag}\,(\bar{H}, \ldots, \bar{H}) \in \mathbb{R}^{m \times mq}, \end{aligned} \tag{4.16}$$

with m blocks each and $\bar{P}$, $\bar{H}$ defined as:

$$\begin{aligned} \bar{P} &= \begin{bmatrix} \mathbf{0} & & \mathbb{I} & & \\ 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{q\times q}, \\ \bar{H} &= \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{1\times q}. \end{aligned} \tag{4.17}$$

In the last row of $\bar{P}$ the 1 occurs at the τ-th position. It is easy to verify that $P_{(\tau,\lambda)}$ in (4.16) is of the form (4.13). A (τ,λ)-lasso sequence has a transient of τ inputs followed by periodic inputs with period λ.

We utilize the (τ,λ)-lasso sequence to formalize a hierarchy of RCISs with a single decision parameter q.

Definition 4.5 (Lassos of same length)**.** Select $q \in \mathbb{Z}^+$. Define the set of all pairs $(\tau,\lambda) \in \mathbb{Z}^+ \cup \{0\} \times \mathbb{Z}^+$ corresponding to lassos of length q as:

$$\Theta_q = \left\{ (\tau,\lambda) \in \mathbb{Z}^+ \cup \{0\} \times \mathbb{Z}^+ \mid \tau + \lambda = q \right\}. \tag{4.18}$$

The cardinality of Θ_q is exactly q.

The next result provides a way to systematically construct implicit RCISs in closed-form such that the corresponding explicit RCISs form a hierarchy.

Theorem 4.7 (Hierarchy of RCISs)**.** *Consider a system Σ and a safe set S_{xu} for which Assumption 4.1 holds, and select an integer $q \in \mathbb{Z}^+$. Given q, the set $\mathscr{C}_{xv,\mathrm{q}} \subset \mathbb{R}^n \times \mathbb{R}^{mq}$:*

$$\mathscr{C}_{xv,\mathrm{q}} = \bigcup_{(\tau,\lambda)\in\Theta_q} \mathscr{C}_{xv,(\tau,\lambda)}, \tag{4.19}$$

is the implicit RCIS induced by the q-level of the hierarchy, where each $\mathscr{C}_{xv,(\tau,\lambda)}$ is computed in closed-form in (4.15) *with P and H as in* (4.16)*. In addition, the explicit RCIS:*

$$\mathscr{C}_{x,\mathrm{q}} = \pi_{[1:n]}\left(\mathscr{C}_{xv,\mathrm{q}}\right) = \bigcup_{(\tau,\lambda)\in\Theta_q} \pi_{[1:n]}\left(\mathscr{C}_{xv,(\tau,\lambda)}\right) = \bigcup_{(\tau,\lambda)\in\Theta_q} \mathscr{C}_{x,(\tau,\lambda)}, \tag{4.20}$$

corresponding to the q-level of the hierarchy contains any RCIS lower in the hierarchy, i.e.:

$$\mathscr{C}_{x,\mathrm{q}} \supseteq \mathscr{C}_{x,\mathrm{q}'}, \text{ for any } q, q' \in \mathbb{Z}^+ \text{ with } q' < q. \tag{4.21}$$

Proof. First, the sets $\mathscr{C}_{xv,\mathrm{q}}$ and $\mathscr{C}_{x,\mathrm{q}}$ are implicit and explicit RCISs respectively as the unions of, implicit and explicit, RCISs by Proposition 4.2. Next we prove (4.21) for the case of q and $q+1$, while the more general statement follows by a simple induction argument.

For any $\lambda \in \mathbb{Z}^+$ such that $(\tau,\lambda) \in \Theta_q$, we have by (4.18) that $(\tau+1,\lambda) \in \Theta_{q+1}$. It is easy to conceptualize that:

$$\mathscr{C}_{x,(\tau+1,\lambda)} \supseteq \mathscr{C}_{x,(\tau,\lambda)}, \tag{4.22}$$

as $\mathscr{C}_{x,(\tau,\lambda)}$ contains the set of states rendered invariant by a (τ,λ)-lasso sequence of inputs, and any (τ,λ)-lasso sequence is also a $(\tau+1,\lambda)$-lasso sequence. Hence, by (4.20):

$$\begin{aligned}\mathscr{C}_{x,(\mathrm{q}+1)} = \bigcup_{(\tau,\lambda)\in\Theta_{q+1}} \mathscr{C}_{x,(\tau,\lambda)} &= \left(\bigcup_{(\tau,\lambda)\in\Theta_q} \mathscr{C}_{x,(\tau+1,\lambda)}\right) \bigcup \mathscr{C}_{x,(0,q+1)} \\ &\overset{(4.22)}{\supseteq} \left(\bigcup_{(\tau,\lambda)\in\Theta_q} \mathscr{C}_{x,(\tau,\lambda)}\right) \bigcup \mathscr{C}_{x,(0,q+1)} \overset{(4.20)}{=} \mathscr{C}_{x,\mathrm{q}} \bigcup \mathscr{C}_{x,(0,q+1)}.\end{aligned}$$

The above entails that $\mathscr{C}_{x,(\mathrm{q}+1)} \supseteq \mathscr{C}_{x,\mathrm{q}}$. □

Corollary 4.7.1. Using the standard big-M formulation, the implicit RCIS $\mathscr{C}_{xv,\mathrm{q}}$ can be expressed as a projection of a higher-dimensional polytope:

$$\mathscr{C}_{xv\zeta,\mathrm{q}} = \left\{ (x,v,\zeta) \,\middle|\, \sum_{i=1}^{q} \zeta_i = 1, G_i(x,v) \leq f_i + (1-\zeta_i)M\mathbf{1} \right\}, \tag{4.23}$$

where $\zeta \in \{0,1\}^q$, G_i and f_i describe each of the q polytopes $\mathscr{C}_{xv,(\tau,\lambda)}$ in (4.19), and $M \in \mathbb{R}_+$ is sufficiently large. The set $\mathscr{C}_{xv\zeta,\mathrm{q}}$ is a polytope in $\mathbb{R}^n \times \mathbb{R}^{mq} \times \{0,1\}^q$, and its projection on $\mathbb{R}^n \times \mathbb{R}^{mq}$ is exactly the union in (4.19), while its projection on $\mathbb{R}^n$ is exactly the explicit RCIS in (4.20).

Theorem 4.7 defines the promised hierarchy and provides an implicit RCIS for each level of the hierarchy that can also be computed in closed-form in (4.23) at the cost of an additional lift. Fig. 4.1 illustrates the relation in (4.21), that is, how the sets induced by each hierarchy level contain the ones induced by lower hierarchy levels.

Remark 4.3 (Convex hierarchy). We can replace the union in (4.19) by the convex hull

$$\operatorname{conv}\left(\bigcup_{(\tau,\lambda)\in\Theta_q} \mathscr{C}_{xv,(\tau,\lambda)}\right).$$

Then, in an analogous manner, all the above results go through resulting in a *hierarchy of convex RCISs*. Similarly to (4.23), by standard set-lifting techniques, one obtains a closed-form expression for the convex hull.

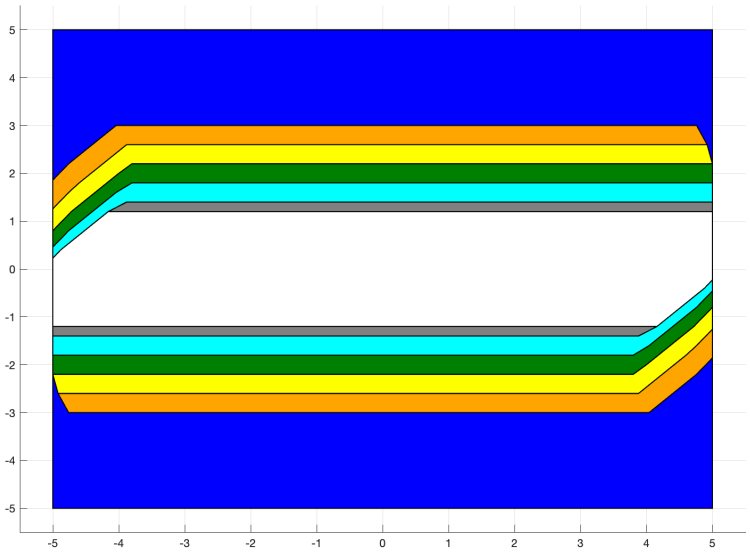


Figure 4.1: RCIS corresponding to $q = 1$ (white), $q = 2$ (gray), $q = 3$ (teal), $q = 4$ (green), $q = 5$ (yellow), and $q = 6$ (orange) for a double integrator. Safe set in blue.

Remark 4.4 (Partial hierarchies without union). The proposed hierarchy involves handling a union of sets. However, one might prefer to avoid unions of sets and rather use a single convex set. As each implicit RCIS $\mathscr{C}_{xv,(\tau,\lambda)}$ involved in the hierarchy is computed in closed-form by Theorem 4.6, we provide two more refined guidelines for obtaining larger RCISs, based on the choice of (τ,λ):

1. Given any $\lambda \in \mathbb{Z}^+$, it holds that $\mathscr{C}_{x,(\tau+1,\lambda)} \supseteq \mathscr{C}_{x,(\tau,\lambda)}$ for any $\tau \in \mathbb{Z}^+ \cup \{0\}$.
2. Given any $\tau \in \mathbb{Z}^+ \cup \{0\}$, it holds that $\mathscr{C}_{x,(\tau,\lambda)} \supseteq \mathscr{C}_{x,(\tau,\lambda')}$ for any $\lambda, \lambda' \in \mathbb{Z}^+$ such that $\lambda = k\lambda'$, $k \in \mathbb{Z}^+$, i.e., λ is a multiple of λ', see [10, Section 4.6] when $\tau = 0$.

The above can direct the designer in search of larger RCISs that are based on a single implicit RCIS.

4.4 Implicit Invariant Sets in Practice: Controlled Invariant Sets in One Move

Using the proposed results, one has the option to project the implicit RCIS back to the original space and obtain an explicit RCIS as proposed in the two-move approach [9, 10, 8]. However, the required projection from a higher dimensional space becomes the bottleneck of this approach.

One of the goals of this chapter is to establish that in a number of key control problems explicit knowledge of the RCIS is not required and the implicit RCIS suffices. We show how the proposed methodology can be used *online* as the implicit RCIS which admits a closed-form expression.

4.4.1 Extraction of admissible inputs

For many applications in this section, we need to extract a set of admissible inputs of the RCIS $\pi_n(C_{xv})$ at a given state x, i.e, $\mathscr{A}(x, \pi_n(C_{xv}))$ as given in Definition 2.4. Given only the implicit RCIS C_{xv}, we provide here three linear encodings of $\mathscr{A}(x, \pi_n(\mathscr{C}_{xv}))$ or its nonempty subsets.

1) The first linear encoding of $\mathscr{A}(x, \pi_n(\mathscr{C}_{xv}))$ is given by the polytope:

$$\mathscr{U}_1(x) = \left\{(u, v_{1:N}) \in \mathbb{R}^{(1+Nq)m} \,\middle|\, (x,u) \in S_{xu}, (Ax+Bu+Ew_i, v_i) \in \mathscr{C}_{xv}, \forall i \in [N]\right\}, \tag{4.24}$$

where $v_{1:N}$ denotes the vector $(v_1, v_2, \cdots, v_N)$. It follows that $\pi_m(\mathscr{U}_1(x)) = \mathscr{A}(x, \pi_n(C_{xv}))$.

2) The second linear encoding is:

$$\mathscr{U}_2(x) = \left\{v \in \mathbb{R}^{qm} \,\middle|\, (x,v) \in \mathscr{C}_{xv}\right\}, \tag{4.25}$$

with H and P as in (4.5). Note that $\mathscr{U}_2(x)$ is the slice of $\mathscr{C}_{xv}$ at x and is nonempty for $x \in \pi_n(\mathscr{C}_{xv})$. Then, the linear transformation $H\mathscr{U}_2(x)$ is a nonempty subset of $\mathscr{A}(x, \pi_n(\mathscr{C}_{xv}))$.

3) Finally, define the polytope:

$$\mathscr{U}_3(x) = \left\{(u, v) \in \mathbb{R}^{(1+q)m} \,\middle|\, (x,u) \in S_{xu}, (Ax+Bu+Ew_i, v) \in \mathscr{C}_{xv}, \forall i \in [N]\right\}, \tag{4.26}$$

where $w_i \in \mathscr{V}$ with $\mathscr{V}$ the vertices of W. It follows that $\pi_m(\mathscr{U}_3(x)) \subseteq \mathscr{A}(x, \pi_n(\mathscr{C}_{xv}))$. It is easy to check that $(Hv, Pv) \in \mathscr{U}_3(x)$ for all $v \in \mathscr{U}_2(x)$, which implies that $\mathscr{U}_3(x)$ is guaranteed to be nonempty for any $x \in \pi_n(\mathscr{C}_{xv})$.

All three linear encodings are easily computed online given $\mathscr{C}_{xv}$. Moreover, it holds that:

$$H\mathscr{U}_2(x) \subseteq \pi_m(\mathscr{U}_3(x)) \subseteq \pi_m(\mathscr{U}_1)(x) = \mathscr{A}(x, \pi_n(\mathscr{C}_{xv})).$$

That is, $\mathscr{U}_2(x)$ is the most conservative encoding, while $\mathscr{U}_1(x)$ is the least conservative one. However, $\mathscr{U}_2$ is of lower dimension, while $\mathscr{U}_1$ has the highest dimension. More conservative encodings are easier to compute. Depending on the available compute, a user can select the most appropriate encoding.

4.4.2 Supervision of a nominal controller

In many scenarios, when synthesizing a controller for a plant, the objective is to meet a performance criterion while always satisfying a safety requirement. This gives rise to the problem of *supervision*.

Problem 4.1 (Supervisory Control)**.** *Consider a system* Σ*, a safe set* S_{xu}*, and a nominal controller that meets a performance objective. The supervisory control problem asks at each time step to evaluate if, given the current state, the input* $\tilde{u}$ *from the nominal controller keeps the next state of* Σ *in the safe set. If not,* correct $\tilde{u}$ *by selecting an input that does so.*

To solve Problem 4.1 one has to guarantee *at every step* that the pairs of states and inputs respect the safe set S_{xu}. A natural way to do so is by using an RCIS. The supervision framework operates as follows. Given an RCIS $\mathscr{C}$, assume that the initial state of Σ lies in $\mathscr{C}$. The nominal controller provides an input $\tilde{u}$ to be executed by Σ. If $\tilde{u} \in \mathscr{A}(x,\mathscr{C})$, then its execution is allowed. Else $\tilde{u}$ is corrected by selecting an input $u_{safe} \in \mathscr{A}(x,\mathscr{C})$. Existence of u_{safe} is guaranteed in any RCIS by Definition 2.3.

In practice an explicit RCIS is not needed. One can exploit the three linear encodings of admissible inputs from the proposed implicit RCISs to perform supervision. Furthermore, the nominal controller can be designed independently of the implicit RCIS parameterization. Consider an implicit RCIS $\mathscr{C}_{xv}$ for Σ with respect to S_{xu}, as in Theorem 4.6. Then supervision of an input $\tilde{u}$ is performed by solving the following QP:

$$\begin{aligned} \min_{u,v} \quad & ||u-\tilde{u}||_2^2 \\ \text{s.t.} \quad & (x,u) \in S_{xu} \\ & (Ax+Bu+Ew,v) \in \mathscr{C}_{xv}, \forall w \in W \end{aligned} \tag{4.27}$$

Notice that the feasible domain of the QP in (4.27) is equal to the third linear encoding $\mathscr{U}_3(x)$ of admissible inputs; similar QPs are easily formulated with the feasible domain being $\mathscr{U}_1(x)$ or $\mathscr{U}_2(x)$. By solving optimization problem (4.27) we compute the *minimally intrusive safe input*.

4.4.3 Safe online planning

Based on the discussed supervision framework, we utilize the proposed implicit RCIS to enforce safety constraints in online planning tasks. The goal here is to navigate a robot through unknown environments without collision with any obstacles. The map is initially unknown, and it is built and updated online based on sensor measurements, such as LiDAR. The robot must only operate in the detected obstacle-free region. To ensure this, given a path planning algorithm and a tracking

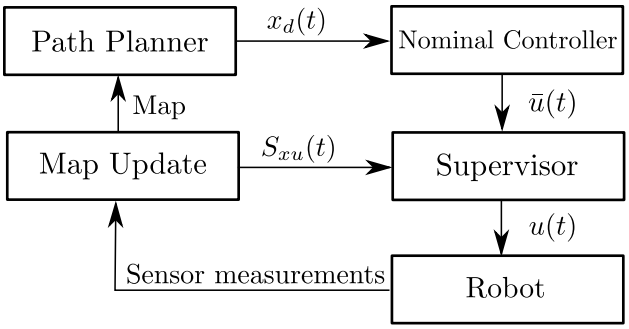


Figure 4.2: The overall safe online planning framework.

controller, we supervise the controller inputs based on the implicit RCIS. The overall framework is shown in Figure 4.2.

The safe set for the robot imposes bounds on states and inputs, which do not change over time, and also constraints, e.g., on the robot's position, which are given by the obstacle-free region in the current map. As the detected obstacle-free region expands over time, the corresponding part of the safe set does as well. Thus, differently from Section 4.4.2, we have a time-varying safe set $S_{xu}(t)$ satisfying $S_{xu}(t) \subseteq S_{xu}(t+1)$, $t \geq 0$. As the implicit RCIS is constructed in closed-form, we can generate a new implicit RCIS $\mathscr{C}_{xv}(t)$ for each $S_{xu}(t)$. Then, at each time step t, for any $t' \leq t$, we supervise the nominal input $\tilde{u}(t)$ by solving the optimization problem:

$$\mathscr{P}(t,t'): \begin{aligned} \min_{u,v} \quad & ||u-\tilde{u}||_2^2 \\ \text{s.t.} \quad & (x,u) \in S_{xu}(t) \\ & (Ax+Bu+Ew,v) \in \mathscr{C}_{xv}(t'), \forall w \in W. \end{aligned}$$

As $S_{xu}(t) \subseteq S_{xu}(t+1)$, $\mathscr{C}_{xv}(t')$ is a valid implicit RCIS in $S_{xu}(t)$ for all $t \geq t'$. Thus, as long as $\mathscr{P}(t,t')$ is feasible, the optimizer v^* of $\mathscr{P}(t,t')$ is a safe input that guarantees the next state lies in the RCIS. Furthermore, if $P(t,t')$ is feasible, by definition of RCIS, $P(t+1,t')$ is also feasible. Thus, if $P(0,0)$ is feasible, for all $t > 0$, there exists $t' \leq t$ such that $P(t,t')$ is feasible. That is, the recursive feasibility of $P(t,t')$ is guaranteed. In practice, to take advantage of the latest map, we always select t' to be the latest time instant t^* for which $P(t,t^*)$ is feasible.

To summarize, at each time step, we first construct the implicit RCIS $\mathscr{C}_{xv}(t)$ based on the current map. Then, given the state and nominal control input, we solve $\mathscr{P}(t,t^*)$ to obtain the minimally intrusive safe input. This input guarantees that the state of the robot stays within $S_{xu}(t)$ for all $t \geq 0$, provided that $\mathscr{P}(0,0)$ is feasible.

4.4.4 Safe hyper-boxes

For high dimensional systems, the exact representation of an RCIS $\mathscr{C}_x$ can be a set of thousands of linear inequalities. This reduces insight as it is quite difficult to clearly identify regions of each state that lie within the RCIS. In contrast, hyper-boxes are easy to grasp in any dimension and immediately provide information about the regions of states they contain. Based on this, we explore how implicit RCISs can be used to find hyper-boxes that can be considered *safe* in the following sense.

Definition 4.6 (Safe hyper-boxes)**.** Consider a system Σ, a safe set S_x, and the maximal RCIS $\mathscr{C}_{max} \subseteq S_x$. Define a hyper-box $\mathscr{B} = [\underline{b}_1, \overline{b}_1] \times \cdots \times [\underline{b}_n, \overline{b}_n] = [\underline{b}, \overline{b}] \subset \mathbb{R}^n$. We call a hyper-box $\mathscr{B}$ *safe* if $\mathscr{B} \subseteq \mathscr{C}_{max}$.

To simplify the presentation we only consider state constraints, S_x, instead of S_{xu}. Notice that by Definition 4.6, a safe hyper-box is not necessarily invariant. A safe hyper-box $\mathscr{B}$ entails the guarantee that the trajectory starting therein can remain in S_x forever, but not necessarily within $\mathscr{B}$. We now aim to address the following problem.

Problem 4.2. *Find the largest*[1] safe *hyper-box $\mathscr{B}$ within $\mathscr{C}_x$.*

A hyper-box $\mathscr{B}$ can be described by a pair of vectors $\left(\underline{b}, \overline{b}\right) \in \mathbb{R}^n \times \mathbb{R}^n$. Then, using similar arguments to Section 4.2, we compute in closed-form an implicit RCIS $\mathscr{C}_{\mathscr{B}}$ characterizing all hyper-boxes $\left(\underline{b}, \overline{b}\right)$ that remain in S_x under eventually periodic inputs. The eventually periodic inputs are given by a vector $v \in \mathbb{R}^{mq}$ with $q = \tau + \lambda$. Then, the set $\mathscr{C}_{\mathscr{B}}$ lives in $\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{mq}$ and is described by:

$$
\begin{aligned}
&A^t \left[\underline{b}, \overline{b}\right] + \sum_{i=1}^{t} A^{i-1} B H P^{t-i} v \subseteq S_x - \overline{W}_t, t = 0, \ldots, \nu - 1, \\
&\sum_{i=1}^{\nu} A^{i-1} B H P^{t-i} v \subseteq S_x - \overline{W}_\infty, t = \nu, \ldots, \nu + q - 1.
\end{aligned}
$$

The above constraints can all be written as linear inequalities in $\left(\underline{b}, \overline{b}, v\right) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{mq}$. Then, the implicit RCIS $\mathscr{C}_{\mathscr{B}}$ is a polytope and one solves Problem 4.2 by the following convex optimization program:

$$
\begin{aligned}
&\max_{\left(\underline{b}, \overline{b}, v\right)} \gamma\left(\overline{b} - \underline{b}\right) \\
&\text{s.t.} \quad \left(\underline{b}, \overline{b}, v\right) \in \mathscr{C}_{\mathscr{B}},
\end{aligned}
$$

[1]The largest, as measured by volume, hyper-box within a set might not be unique. We choose a heuristic for maximizing the volume of a set that yields a well-defined convex optimization problem. Hence, the term "largest" refers to the heuristic used.

where $\gamma(y) = \left(\Pi_{i=1}^{n} y_i\right)^{\frac{1}{n}}$ is the geometric mean function, which is used as a heuristic for the volume of the hyper-box. Function γ is concave, and maximizing a concave function can be cast as a convex minimization problem [24].

Remark 4.5 (Invariant and recurrent hyper-boxes). Two special cases of the above are *invariant* hyper-boxes, when $\tau = 0$, $\lambda = 1$, see also [9], and *recurrent* hyper-boxes, when $\tau = 0$, $\lambda > 0$, see also [10, 8].

A related question to Problem 4.2 is to evaluate if a proposed hyper-box is safe. This is of interest when evaluating whether the initial condition of a problem or an area around a configuration point x_c where the system is required to operate is safe. If both the above are modeled by hyper-boxes $(\underline{b}, \overline{b})$, we can simply ask whether there exists a v, such that $(\underline{b}, \overline{b}, v) \in \mathscr{C}_{\mathscr{B}}$. Similarly, more complicated questions can be formulated, e.g., to find the largest safe box around a configuration point.

Remark 4.6 (Complexity when using implicit RCISs). In this section we showed how several key problems in control are solved without the need of projection and of an explicit RCIS, which results in extremely efficient computations since the implicit RCISs are computed in closed-form. The decision to be made is the size of the lift, i.e., the length of the input sequence. From a computational standpoint, this choice is only limited by how large an optimization problem one affords solving given the application.

4.5 Performance Bound for the Proposed Method

Numerical examples, to be presented later, will show that the projection of the proposed implicit RCIS onto the original state-space can coincide with the maximal RCIS. However, this is not always the case. When there is a gap between our projected set and the maximal RCIS, one may wonder if that gap is fundamental to our method. In other words, can we arbitrarily approximate the maximal RCIS with the projection of our implicit RCIS by choosing better P and H matrices?

In this section we aim to answer the above question and provide insights into the completeness of our method. Given (4.3), define the nominal system $\overline{\Sigma}$ and the nominal safe set $\overline{S}_{xu}$:

$$\overline{\Sigma} \; : \; x^+ = Ax + Bu, \tag{4.28}$$

$$\overline{S}_{xu} = S_{xu} - \overline{W}_{\infty} \times \{0\}, \tag{4.29}$$

where A and B are the same as in (4.1). Let $\overline{\mathscr{C}}_{\max}$ be the maximal CIS of the nominal system $\overline{\Sigma}$ with

respect to $\overline{S}_{xu}$ and define:

$$\mathscr{C}_{outer,\nu} = \Big\{ x \in \mathbb{R}^n \,\Big|\, \exists \{u_i\}_{i=0}^{\nu-1} \in \mathbb{R}^{m\nu}, \big(\mathscr{R}_\Sigma\big(x, \{u_i\}_{i=0}^{t-1}\big), u_t\big) \subseteq S_{xu}, t = 0, \ldots, \nu-1, \\ \mathscr{R}_\Sigma\big(x, \{u_i\}_{i=0}^{\nu-1}\big) \subseteq \overline{\mathscr{C}}_{\max} + \overline{W}_\infty \Big\}, \tag{4.30}$$

where ν is the nilpotency index of A.

Proposition 4.8. $\mathscr{C}_{outer,\nu}$ is an RCIS of Σ with respect to S_{xu}.

Proof. In this proof, we use the order cancellation lemma, as a special case of [42, Thm. 4].

Lemma 4.9. Let $X, Y \subset \mathbb{R}^n$ be two closed convex sets with Y bounded. A point $x \in \mathbb{R}^n$ is in X if and only if $x + Y \subseteq X + Y$.

To prove that $\mathscr{C}_{outer,\nu}$ is an RCIS, we show that for any $x_0 \in \mathscr{C}_{outer,\nu}$, there exists u such that $(x_0, u) \in S_{xu}$ and for all $w \in W$, $Ax_0 + Bu + Ew \in \mathscr{C}_{outer,\nu}$. By definition of $\mathscr{C}_{outer,\nu}$, there exists a sequence $\{u_i\}_{i=0}^{\nu-1}$ that, along with x_0, satisfies the conditions in (4.30). We aim to show that u_0 in $\{u_i\}_{i=0}^{\nu-1}$ is a feasible choice for u. Given (4.30), the reachable set from x_0 at time ν is:

$$\mathscr{R}_\Sigma\Big(x_0, \{u_i\}_{i=0}^{\nu-1}\Big) = \sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i + \overline{W}_\infty \subseteq \overline{\mathscr{C}}_{\max} + \overline{W}_\infty,$$

with $\overline{W}_\infty$ and $\overline{\mathscr{C}}_{\max}$ convex and $\overline{W}_\infty$ bounded. By Lemma 4.9 we have that $\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i \in \overline{\mathscr{C}}_{\max}$. Since $\overline{\mathscr{C}}_{\max}$ is controlled invariant with respect to $\overline{S}_{xu}$ for the nominal system $\overline{\Sigma}$, there exists u_ν such that:

$$\left(\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i, u_\nu \right) \in \overline{S}_{xu}, \tag{4.31}$$
$$A \left(\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i \right) + B u_\nu = \sum_{i=1}^{\nu} A^{\nu-1-i} B u_i \in \overline{\mathscr{C}}_{\max}.$$

Consider any $w \in W$ and define $x_1 = Ax + Bu_0 + Ew$:

$$\mathscr{R}_\Sigma(x_1, \{u_i\}_{i=1}^{\nu}) = \sum_{i=1}^{\nu} A^{\nu-1-i} B u_i + \overline{W}_\infty \subseteq \overline{\mathscr{C}}_{\max} + \overline{W}_\infty. \tag{4.32}$$

From (4.31) we have that:

$$(\mathscr{R}_\Sigma(x_1, \{u_i\}_{i=1}^{\nu-1}), u_\nu) \subseteq S_{xu}. \tag{4.33}$$

Finally, note that for $t = 0, \cdots, \nu - 2$, we have:

$$\left(\mathscr{R}_{\Sigma}(x_1, \{u_i\}_{i=1}^{t}), u_{t+1}\right) \subseteq \left(\mathscr{R}_{\Sigma}(x_0, \{u_i\}_{i=0}^{t}), u_{t+1}\right) \subseteq S_{xu}. \tag{4.34}$$

From (4.32), (4.33), and (4.34) we verify that $x_1 \in \mathscr{C}_{outer,\nu}$. Thus, $\mathscr{C}_{outer,\nu}$ is an RCIS. □

The following theorem shows that $\mathscr{C}_{outer,\nu}$ is an outer bound of the projection of the proposed implicit RCIS.

Theorem 4.10 (Outer bound on $\pi_{[1:n]}(\mathscr{C}_{xv})$)**.** *For a companion system Σ_{xv} as in* (4.7)*, with arbitrary matrices P and H, let $\mathscr{C}_{xv}$ be an RPIS of Σ_{xv} within the companion safe set S_{xv}. The RCIS $\pi_{[1:n]}(\mathscr{C}_{xv})$ is a subset of $\mathscr{C}_{outer,\nu}$, that is $\pi_{[1:n]}(\mathscr{C}_{xv}) \subseteq \mathscr{C}_{outer,\nu}$.*

Proof. Let $x \in \pi_{[1:n]}(\mathscr{C}_{xv})$. We show that $x \in \mathscr{C}_{outer,\nu}$. By definition of $\mathscr{C}_{xv}$, there exists a vector v such that:

$$\left(\mathscr{R}_{\Sigma}\left(x, \left\{HP^i v\right\}_{i=0}^{t-1}\right), HP^t v\right) \subseteq S_{xu}, \text{ for all } t \geq 0. \tag{4.35}$$

Define $u_t = HP^t v$. We want to verify that x and $\{u_i\}_{i=0}^{\nu-1}$ satisfy the two conditions in the definition of (4.30). The first condition is immediately satisfied by (4.35). It is left to show that $\mathscr{R}_{\Sigma}(x, \{u_i\}_{i=0}^{\nu-1}) \subseteq \overline{\mathscr{C}}_{\max} + \overline{W}_{\infty}$. That is:

$$\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i + \overline{W}_{\infty} \subseteq \overline{\mathscr{C}}_{\max} + \overline{W}_{\infty}.$$

By Lemma 4.9, it is equivalent to prove that:

$$\bar{x} \equiv \sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i \in \overline{\mathscr{C}}_{\max}.$$

By (4.35), we have that for $t \geq 0$:

$$\begin{aligned} &\left(\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_{i+t} + \overline{W}_{\infty}, u_{\nu+t}\right) \subseteq S_{xu} \Leftrightarrow \left(\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_{i+t}, u_{\nu+t}\right) \in \overline{S}_{xu} \\ \Leftrightarrow &\left(\mathscr{R}_{\overline{\Sigma}}(\bar{x}, \{u_i\}_{i=\nu}^{\nu+t-1}), u_{\nu+t}\right) \in \overline{S}_{xu} \end{aligned} \tag{4.36}$$

According to (4.36), the control sequence $\{u_i\}_{i=\nu}^{\nu+t-1}$ guarantees that the trajectory of $\overline{\Sigma}$ starting at $\bar{x}$ stays within $\overline{S}_{xu}$ for all $t \geq 0$. Thus, $\bar{x}$ must belong to the maximal CIS of Σ in $\overline{S}_{xu}$. That is, $\bar{x} \in \overline{\mathscr{C}}_{\max}$. □

Note here that the set $\mathscr{C}_{outer,v}$, which serves as an outer bound for the set computed by our method, is as hard to compute as the maximal RCIS. Given Theorem 4.10 we have:

$$\pi_{[1:n]}(\mathscr{C}_{xv}) \subseteq \mathscr{C}_{outer,v} \subseteq \mathscr{C}_{\max}. \tag{4.37}$$

Thus, the projection of our implicit RCIS can coincide with the maximal RCIS, for appropriately selected matrices P and H, only if $\mathscr{C}_{outer,v} = \mathscr{C}_{\max}$ in (4.37). This potential gap between our approximation and the maximal RCIS is due to the fact that our method uses open-loop forward reachability constraints under disturbances. Finally, the following theorem establishes weak completeness of our method.

Theorem 4.11 (Weak completeness)**.** *The set $\mathscr{C}_{outer,v}$ is nonempty, if and only if, there exist matrices P and H such that the corresponding implicit RCIS $\mathscr{C}_{xv}$ is nonempty. Specifically, $\mathscr{C}_{outer,v} \neq \emptyset$, if and only if, $\mathscr{C}_{xv,(0,1)} \neq \emptyset$, that is P and H are as in* (4.16) *with* $(\tau, \lambda) = (0,1)$.

Proof. We want to show that $\mathscr{C}_{outer,v}$ is nonempty if and only if $\mathscr{C}_{xv,(0,1)}$ is nonempty, where $\mathscr{C}_{xv,(0,1)}$ is defined in (4.19) with respect to system Σ and safe set S_{xu}.

Since $\pi_{[1:n]}\left(\mathscr{C}_{xv,(0,1)}\right) \subseteq \mathscr{C}_{outer,v}$, immediately nonemptyness of $\mathscr{C}_{xv,(0,1)}$ implies nonemptyness of $\mathscr{C}_{outer,v}$.

For the converse, suppose that $\mathscr{C}_{outer,v}$ is nonempty. Then $\overline{\mathscr{C}}_{\max}$ is nonempty. By [27, Theorem 12], we know that $\overline{\mathscr{C}}_{\max}$ is nonempty, if and only if, there exists a fixed point $x \in \overline{\mathscr{C}}_{\max}$ along with a u such that $(x,u) \in \overline{S}_{xu}$ and $Ax + Bu = x$. Also, note that $A\overline{W}_\infty + EW = \overline{W}_\infty$. Thus, we have:

$$\begin{aligned} &(x + \overline{W}_\infty, u) \subseteq S_{xu}, \\ &A(x + \overline{W}_\infty) + Bu + EW = x + \overline{W}_\infty. \end{aligned} \tag{4.38}$$

According to (4.38), for any $y \in x + \overline{W}_\infty$, we have $(y,u) \in S_{xu}$ and $Ay + Bu + EW \subseteq x + \overline{W}_\infty$, which implies that $x + \overline{W}_\infty$ is an RCIS of Σ with respect to S_{xu}. By the definition of $\mathscr{C}_{xv,(0,1)}$, it is easy to check that $(x + \overline{W}_\infty, u) \subseteq \mathscr{C}_{xv,(0,1)}$. Thus, $\mathscr{C}_{xv,(0,1)}$ is nonempty. □

Corollary 4.11.1 (Completeness in absence of disturbances). In the absence of disturbances, $\mathscr{C}_{outer,v} = \mathscr{C}_{max}$ and, thus, there exist P and H such that $\mathscr{C}_{xv}$ is nonempty, if and only if, $\mathscr{C}_{max}$ is nonempty. That is, the proposed method is complete.

The significance of Theorem 4.11 lies in allowing to quickly check nonemptiness of $\mathscr{C}_{outer,v}$ by computing $\mathscr{C}_{xv,(0,1)}$, which we can do in closed-form. Even though the gap between $\mathscr{C}_{outer,v}$ and $\mathscr{C}_{\max}$ is still an open question at the writing of this chapter, we show that $\pi_{[1:n]}(\mathscr{C}_{xv})$ can actually converge to its outer bound for a specific choice of H and P matrices.

Theorem 4.12 (Convergence to $\mathscr{C}_{outer,\nu}$)**.** *Assume that the disturbance set W contains 0, and the interior of $\overline{S}_{xu}$ contains a fixed point (x,u) of $\overline{\Sigma}$. There exist matrices H and P such that $\pi_{[1:n]}(\mathscr{C}_{xv})$ approaches $\mathscr{C}_{outer,\nu}$. Specifically, if H and P are as in* (4.16)*, by increasing τ in* (4.16)*, $\pi_{[1:n]}(\mathscr{C}_{xv})$ converges to $\mathscr{C}_{outer,\nu}$ in Hausdorff distance exponentially fast.*

Proof. Without loss of generality, assume that the fixed point (x,u) of $\overline{\Sigma}$ in the interior of $\overline{S}_{xu}$ is the origin of the state-input space. We define a set operator $\mathscr{U}(\mathscr{C})$ that maps a subset $\mathscr{C}$ of $\mathbb{R}^n$ to a subset of $\mathbb{R}^{\nu m}$:

$$\mathscr{U}(\mathscr{C}) = \left\{ u_{0:\nu-1} \in \mathbb{R}^{\nu m} \;\middle|\; \sum_{i=1}^{\nu} A^{i-1} B u_{\nu-i} \in \mathscr{C} \right\}, \tag{4.39}$$

where $u_{0:\nu-1}$ denotes the vector $(u_0, u_1, \cdots, u_{\nu-1}) \in \mathbb{R}^{\nu m}$.

To maintain a streamlined presentation, we make the following claims that we prove in Appendix B.1.

Claim 1: The polytope $\mathscr{C}_{xv,0}$ contains the origin, where:

$$\mathscr{C}_{xv,0} = \{(x, u_{0:\nu-1}) \in \mathbb{R}^{n+\nu m} \mid \left(\mathscr{R}_{\Sigma}\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu}, t = 0, \ldots, \nu-1\}.$$

Claim 2: For the set $\mathscr{C}_{outer,\nu}$ in (4.30) it holds that:

$$\mathscr{C}_{outer,\nu} = \pi_n(\mathscr{C}_{xv,max}), \tag{4.40}$$

where $\mathscr{C}_{xv,max} = \mathscr{C}_{xv,0} \cap (\mathbb{R}^n \times \mathscr{U}(\overline{\mathscr{C}}_{max}))$.

Claim 3: Let $\overline{\mathscr{C}}_{xv,(\tau,\lambda)}$ be the implicit CIS of the nominal system $\overline{\Sigma}$ with respect to $\overline{S}_{xu}$ with H and P as in (4.16) and let $\overline{\mathscr{C}}_{x,(\tau,\lambda)} = \pi_n(\overline{\mathscr{C}}_{xv,(\tau,\lambda)})$. The implicit RCIS $\mathscr{C}_{xv,(\tau,\lambda)}$ of Σ with respect to S_{xu} with H and P as in (4.16) satisfies:

$$\pi_n(\mathscr{C}_{xv,(\tau,\lambda)}) = \pi_n(\widehat{\mathscr{C}}_{xv,(\tau,\lambda)}), \text{ for any } \tau \geq \nu, \tag{4.41}$$

where $\widehat{\mathscr{C}}_{xv,(\tau,\lambda)} = \mathscr{C}_{xv,0} \cap (\mathbb{R}^n \times \mathscr{U}(\overline{\mathscr{C}}_{x,(\tau-\nu,\lambda)}))$.

Claim 4: There exist $c_0 > 0$, $a \in [0,1)$, and some $\tau_1 \geq 0$ such that for any $\lambda \geq 1$ and for any $\tau \geq \tau_1$:

$$\overline{\mathscr{C}}_{x,(\tau,\lambda)} \supseteq (1 - c_0 a^{\tau}) \overline{\mathscr{C}}_{max}, \tag{4.42}$$

with τ_1 big enough such that $1 - c_0 a^{\tau_1} \geq 0$ and thereby the right hand side of (4.42) is well-defined.

We use these claims to prove the desired convergence rate. The operator $\mathscr{U}(\cdot)$ in (4.39) is linear with respect to scalar multiplication, i.e, $\mathscr{U}(\xi\mathscr{C}) = \xi\mathscr{U}(\mathscr{C})$, $\xi \geq 0$, and monotonic, i.e.,

$\mathscr{U}(\mathscr{C}_1) \supseteq \mathscr{U}(\mathscr{C}_2)$, $\mathscr{C}_1 \supseteq \mathscr{C}_2$. According to (4.42), for $\tau \geq \tau_1$:

$$\mathscr{U}(\overline{\mathscr{C}}_{x,(\tau,\lambda)}) \supseteq (1 - c_0 a^{\tau}) \mathscr{U}(\overline{\mathscr{C}}_{max}). \tag{4.43}$$

Note $\tau_0 = \nu + \tau_1$. By (4.41), for $\tau \geq \tau_0$:

$$\begin{aligned} \widehat{\mathscr{C}}_{xv,(\tau,\lambda)} &\supseteq \mathscr{C}_{xv,0} \cap (1 - c_0 a^{\tau-\nu})(\mathbb{R}^n \times \mathscr{U}(\overline{\mathscr{C}}_{max})) \supseteq (1 - c_0 a^{\tau-\nu})(\mathscr{C}_{xv,0} \cap (\mathbb{R}^n \times \mathscr{U}(\overline{\mathscr{C}}_{max}))) \\ &\supseteq (1 - c_0 a^{\tau-\nu}) \mathscr{C}_{xv,max}. \end{aligned} \tag{4.44}$$

The second inclusion above holds since $0 \in \mathscr{C}_{xv,0}$ and thus $(1 - c_0 a^{\tau}) \mathscr{C}_{xv,0} \subseteq \mathscr{C}_{xv,0}$. Note that $\pi_n(\cdot)$ is also linear with respect to scalar multiplication. By (4.40), (4.41) and (4.44), for $\tau \geq \tau_0$:

$$\mathscr{C}_{x,(\tau,\lambda)} = \pi_n(\mathscr{C}_{xv,(\tau,\lambda)}) = \pi_n(\widehat{\mathscr{C}}_{xv,(\tau,\lambda)}) \supseteq \pi_n((1 - c_0 a^{\tau-\nu}) \mathscr{C}_{xv,max}) = (1 - c_0 a^{\tau-\nu}) \mathscr{C}_{outer,\nu}. \tag{4.45}$$

By Theorem 4.10 and (4.45), for any $\tau \geq \tau_0$:

$$(1 - c_0 a^{\tau-\nu}) \mathscr{C}_{outer,\nu} \subseteq \mathscr{C}_{x,(\tau,\lambda)} \subseteq \mathscr{C}_{outer,\nu}. \tag{4.46}$$

Let $c_1 = \max_{x_1, x_2 \in \mathscr{C}_{outer,\nu}} \|x_1 - x_2\|_2$ be the diameter of $\mathscr{C}_{outer,\nu}$, which is finite since S_{xu} is bounded. Then, by (4.46), the Hausdorff distance between $\mathscr{C}_{x,(\tau,\lambda)}$ and $\mathscr{C}_{outer,\nu}$ satisfies:

$$d(\mathscr{C}_{x,(\tau,\lambda)}, \mathscr{C}_{outer,\nu}) \leq c a^{\tau}, \text{ for } c = c_0 c_1 a^{-\nu} \text{ and } \tau \geq \tau_0.$$

□

Note that $\mathscr{C}_{outer,\nu}$ contains the union of the projections $\pi_n(\mathscr{C}_{xv})$ for all general implicit RCISs $\mathscr{C}_{xv}$ suggested by Theorem 4.3 (that is, the matrices H and P can be arbitrary, not necessarily the eventually periodic ones in Section 4.2.3). Hence, intuitively the set $\mathscr{C}_{outer,\nu}$ should be much larger than the projection of any specific implicit RCIS $\mathscr{C}_{xv}$ corresponding to an eventually periodic H and P in Section 4.2.3. However, Theorem 4.12 shows that the proposed implicit RCIS can approximate $\mathscr{C}_{outer,\nu}$ arbitrarily well by just using the simple H and P matrices as in (4.17). Moreover, the approximation error decays exponentially fast as we increase the parameter τ in (4.17). This result implies that the eventually periodic input structure explored in Section III.B and III.C is rich enough, and not as conservative as what it may look at first sight.

Corollary 4.12.1. In the absence of disturbances, if the interior of S_{xu} contains a fixed point of Σ, then for any $\lambda > 0$, then $\mathscr{C}_{x,(\tau,\lambda)}$ converges to the maximal CIS $\mathscr{C}_{max}$ in Hausdorff distance exponentially fast as τ increases.

The condition that the interior of S_{xu} (resp. $\overline{S}_{xu}$) contains a fixed point of Σ (resp. $\overline{\Sigma}$) in Corollary 4.12.1 (resp. Theorem 4.12) is critical to our method:

Example 4.1. Let Σ be $x_1^+ = x_2$, $x_2^+ = u$ and the safe set $S_{xu} = \{(x,u)| -1 \leq x_1,\ 1.5x_2 \leq x_1 \leq 2x_2, u \in [-1,1]\}$. The only fixed point of Σ in S_{xu} is the origin in $\mathbb{R}^3$, which is also a vertex of S_{xu}. It is easy to check that $\mathscr{C}_{max} = \pi_n(S_{xu})$, but the largest CIS $\pi_n(C_{xv})$ computed by our method is equal to the singleton set $\{0\}$.

If we expand S_{xu} slightly so that its interior contains the origin, there immediately exist H and P such that $\pi_n(\mathscr{C}_{xv})$ approximates $\mathscr{C}_{max}$ arbitrarily well, as expected by Corollary 4.12.1. Conversely, if we slightly shrink S_{xu} so that it does not contain any fixed point, then $\mathscr{C}_{max}$ is empty [27, Theorem12].

Remark 4.7. Under the assumption that $0 \in W$, let $\mathbb{S}_{xu}$ be the set of all the polytopic safe sets S_{xu} that have a nonempty $\mathscr{C}_{outer,v}$. Moreover, let $\partial\mathbb{S}_{xu}$ be the set of all safe sets $S_{xu} \in \mathbb{S}_{xu}$, whose corresponding nominal safe set $\overline{S}_{xu}$ does not contain a fixed point of $\overline{\Sigma}$ in the interior. It can be shown that $\partial\mathbb{S}_{xu}$ must be contained by the boundary of $\mathbb{S}_{xu}$ in the topology induced by Hausdorff distance. Consequently, for any safe set in the interior of $\mathbb{S}_{xu}$, there exists H and P such that $\pi_n(\mathscr{C}_{xv})$ approximates $\mathscr{C}_{outer,v}$ arbitrarily well.

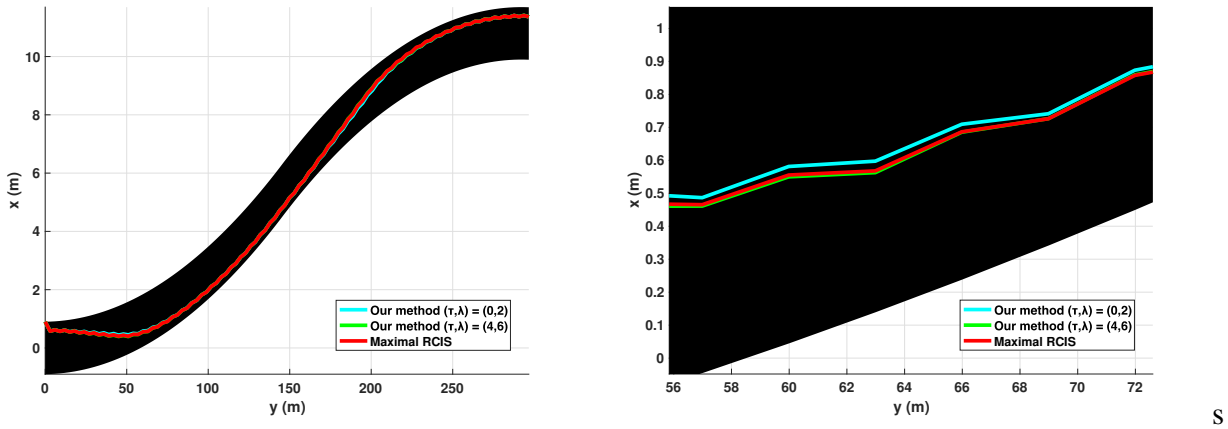
4.6 Case Studies

A MATLAB implementation of the proposed method, along with instructions to replicate our case studies, can be found at https://github.com/janis10/cis2m.

4.6.1 Lane keeping supervision using implicit RCIS

We begin by tackling the supervision problem, defined in Section 4.4.2, for the task of vehicle lane keeping control. That is, we filter vehicle steering inputs to ensure the lateral position of the vehicle to stay within lane boundaries. We consider a 4-dimensional linearized bicycle model with respect to a constant longitudinal velocity $\bar{v}$ [112], discretized with time step $T_s = 0.1s$. The states consist of the lateral displacement y, the lateral velocity v, the yaw angle $\Delta\Psi$, and the yaw rate r. The input δ is the steering angle. The disturbance r_d is the road curvature. The safe set is given by constraints $y \in [-0.9, 0.9]$, $v \in [-1.2, 1.2]$, $\Delta\Psi \in [-0.05, 0.05]$, $r \in [-0.3, 0.3]$, and $\delta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. The disturbance bounds are $r_d \in [-0.015, 0.015]$.

The nominal controller to be supervised is $u_d(t) = -0.1812y - 0.0373v - 4.5996\Delta\Psi - 0.6649r$. We implement the safety supervisor in (4.27) using the implicit RCIS $\mathscr{C}_{xv,(\tau,\lambda)}$ for (τ,λ) equal to $(0,2)$ and $(4,6)$, and compare them with the safety supervisor in (2.15) using the maximal RCIS.



s

Figure 4.3: Vehicle maneuvers under the safety supervision with the maximal RCIS (red) and the implicit RCIS $\mathscr{C}_{xv,(\tau,\lambda)}$ for $(\tau,\lambda) = (0,2)$ (cyan) and $(\tau,\lambda) = (4,6)$ (green). The black region depicts the lane shape.

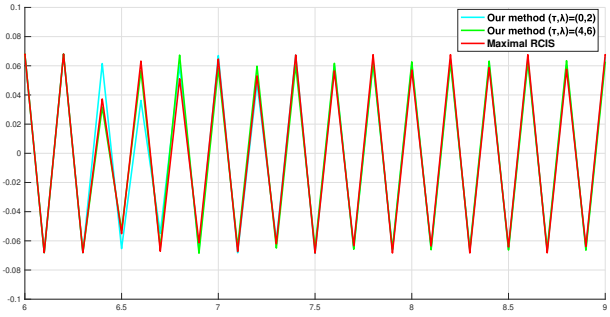


Figure 4.4: Filtered vehicle steering inputs ($6 \leq t \leq 9$) with respect to the maximal RCIS (red) and the implicit RCIS $\mathscr{C}_{xv,(\tau,\lambda)}$ for $(\tau,\lambda) = (0,2)$ (cyan) and $(\tau,\lambda) = (4,6)$ (green).

We run simulations for each supervisor under the same initial states and disturbance inputs. Fig. 4.3 compares the simulated vehicle maneuvers under the three safety supervisors. As shown by the zoomed-in plot in Fig. 4.3, even when supervised with $\mathscr{C}_{xv,(0,2)}$, the trajectory already follows tightly with the trajectory supervised with the maximal RCIS, and as we increase τ and λ, it gets even closer. Similar phenomenon can be observed in the filtered steering inputs shown in Fig. 4.4. As we increase τ and λ from $(0,2)$ to $(4,6)$, the difference between the steering signal supervised with the implicit RCIS $\mathscr{C}_{xv,(\tau,\lambda)}$ and that with the maximal RCIS becomes significantly smaller, indicating that our implicit RCIS approximates the maximal RCIS better as τ and λ increase.

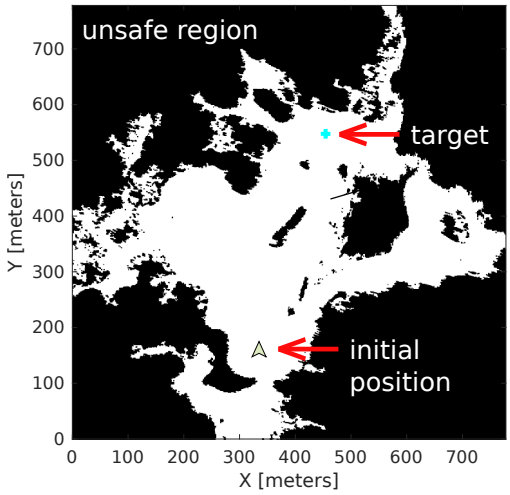


Figure 4.5: Robot operational space: initial position (yellow arrowhead), target position (cyan), unsafe region (dark area).

4.6.2 Safe online planning using implicit RCIS

Next, we solve the safe online planning problem, discussed in Section 4.4.3, for ground robot navigation. The map is initially unknown and is built online based on LiDAR measurements. While navigating the robot needs to avoid the obstacles, indicated by the dark area in Fig. 4.5, and reach the target point. This case study is inspired by the robot navigation problem in [17].

The robot's motion, using forward Euler discretization, is:

$$x^+ = \begin{bmatrix} \mathbb{I} & \mathbb{I}T_s \\ 0 & \mathbb{I} \end{bmatrix} x + \begin{bmatrix} 0 \\ \mathbb{I}T_s \end{bmatrix} u,$$

where the state $x = (p_x, p_y, v_x, v_y) \in \mathbb{R}^4$ is the robot's position and velocity and the input $u = (u_1, u_2) \in \mathbb{R}^2$ is the acceleration. The safe set consists of two parts:

1) The time-invariant constraints $v_x, v_y \in [-\overline{v}, \overline{v}]$ and $u_1, u_2 \in [-\overline{u}, \overline{u}]$.

2) The time-varying constraint of (p_x, p_y) within the obstacle-free region, shown by the white nonconvex area in Fig. 4.5. The obstacle-free region, denoted by $M(t) \subseteq R^2$, is determined by a LiDAR sensor using data up to time t. Combining the two constraints, the safe set at time t is:

$$\begin{aligned} S_{xu}(t) = \{&(p_x, p_y, v_x, v_y, u_1, u_2) \mid (p_x, p_y) \in M(t), \\ &v_x, v_y \in [-\overline{v}, \overline{v}],\ u_1, u_2 \in [-\overline{u}, \overline{u}]\}. \end{aligned}$$

Since $M(t) \subseteq M(t+1)$, we have $S_{xu}(t) \subseteq S_{xu}(t+1)$, $t \geq 0$.

The overall control framework is shown in Fig. 4.2. Initially, the map is blank and the path planner

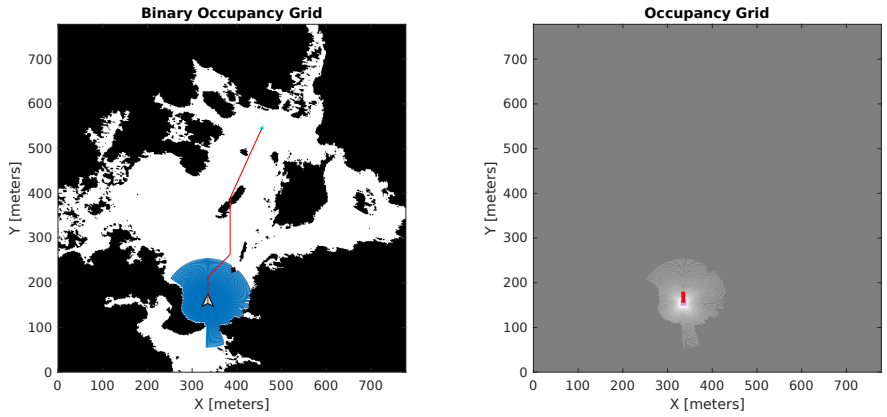
generates a reference trajectory assuming no obstacles. At each time t, the map is updated based on the latest LiDAR measurements and the path planner checks if the reference trajectory collides with any obstacles in the updated map. If so, it generates a new, collision-free, reference path. Then, the nominal controller provides a candidate input $\tilde{u} = (\tilde{u}_1(t), \tilde{u}_2(t))$ tracking the reference path. When updating the reference trajectory, a transient period is needed for the robot to converge to the new reference. Moreover, the path planner cannot guarantee satisfaction of the input constraints. To resolve these issues, we add a supervisory control to the candidate inputs. Based on the updated obstacle-free region $M(t)$, we construct the safe set $S_{xu}(t)$ and compute an implicit CIS $\mathscr{C}_{xv,(\tau,\lambda)}(t)$ with respect to $S_{xu}(t)$. To handle the nonconvexity of $S_{xu}(t)$, we first compute a convex composition of $S_{xu}(t)$. When constructing $\mathscr{C}_{xv,(\tau,\lambda)}(t)$, we let the reachable set at each time belong to one of the convex components in $S_{xu}(t)$, encoded by mixed-integer linear inequalities. For details see [79]. The convex decomposition of $S_{xu}(t)$ becomes more complex over time, which slows down the algorithm. To lighten the computational burden, we replace the full convex composition by the union of the 10 largest hyper-boxes in $S_{xu}(t)$ as the safe set. Given the constructed implicit CIS $\mathscr{C}_{xv,(\tau,\lambda)}(t)$ at time t, we supervise the nominal control input $\tilde{u}(t)$ by solving $\mathscr{P}(t,t^*)$ as discussed in Section 4.4.3. Note that $P(t,t^*)$ becomes a mixed-integer program as we introduced binary variables for the convex composition of the safe set and, therefore, in the implicit CIS.

In our simulations, we use a linear feedback controller as the nominal controller. The MATLAB Navigation Toolbox is used to simulate a LiDAR sensor with sensing range of 100 m, update the map, and generate the reference path based on the A* algorithm. The simulation parameters are $(\tau,\lambda) = (6,4)$, $T_s = 0.1s$, $\bar{v} = 5m/s$, $\bar{u} = 5m/s^2$. The mixed-integer program $\mathscr{P}(t,t^*)$ is implemented via YALMIP [84] and solved by GUROBI [44]. The average computation time for constructing $\mathscr{C}_{xv,(\tau,\lambda)}(t)$ and solving $\mathscr{P}(t,t^*)$ at each time step is $2.87s$. The average computation time shows the efficiency of our method, considering the safe set is nonconvex and being updated at every time step.

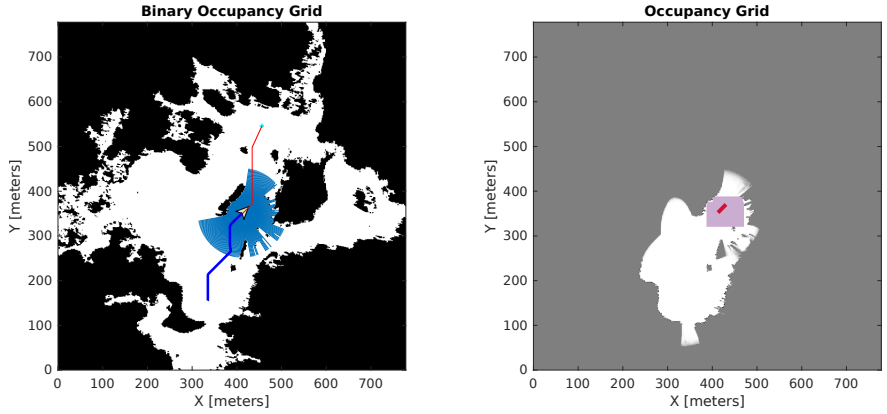
The simulation results are shown in Fig. 4.6. The robot reaches the target region at $t = 78.6s$, and thanks to the supervisor, it satisfies the input and velocity constraints, while always staying within the time-varying safe region. As a comparison, when the supervisor is disabled, the velocity constraint is violated at time $t = 1.2s$. The full simulation video can be found at https://youtu.be/mB9ir0R9bzM.

4.6.3 Scalability and quality

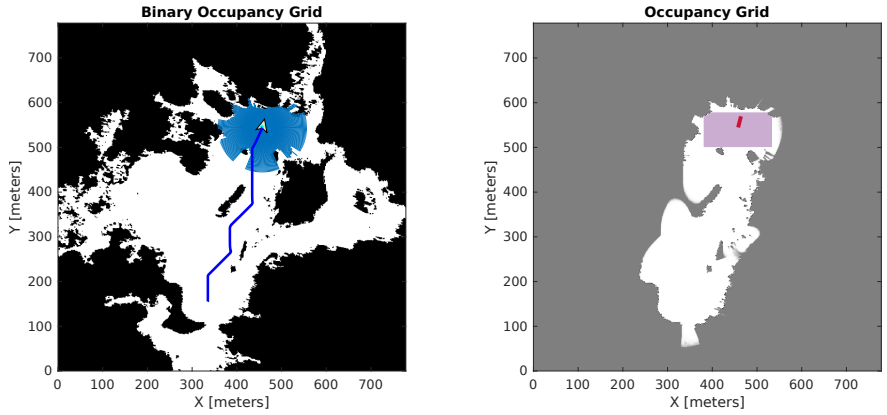
In this subsection we illustrate the scalability of the proposed method and compare with other methods in the literature. We consider a system of dimension n as in (4.1) that is already in



(a) $t = 0s$

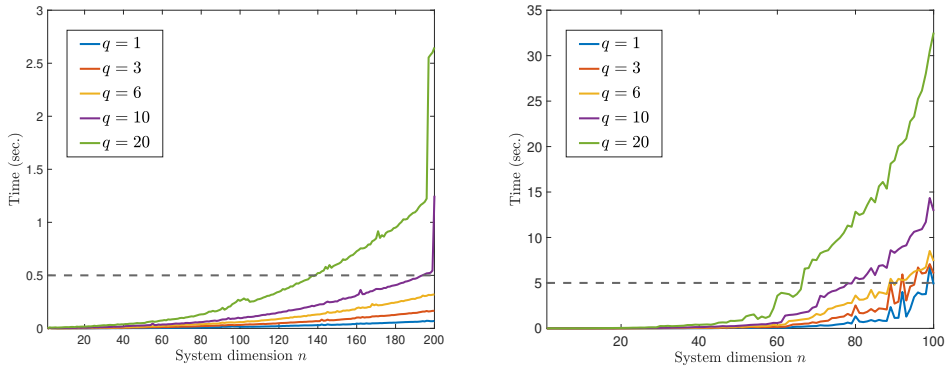


(b) $t = 40s$



(c) $t = 78.6s$

Figure 4.6: Simulation screenshots at times $t = 0s$, $40s$ and 78.6s.
Left: reference path (red) and actual trajectory (blue); the disk of blue rays is the LiDAR measurements; the arrowhead indicates the position and moving direction of the robot.
Right: obstacle-free region $M(t)$ (white) and unknown region (grey); purple boxes are the 10 largest boxes in $M(t)$ that contain the current robot position.



(a) (b)

Figure 4.7: Absence of disturbances. Computation times for implicit CISs for different levels q of the full hierarchy, i.e., computing q Implicit CISs per level. (a) Safe sets with $2n$ constraints, $n \leq 200$. (b) Safe sets with n^2 constraints, $n \leq 100$.

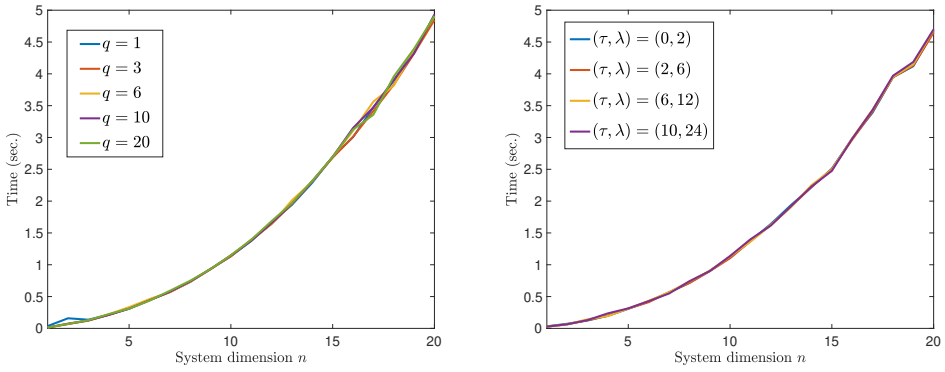
Brunovsky normal form [26].

$$A_n = \begin{bmatrix} 0 & \mathbb{I} \\ 0 & 0 \end{bmatrix}, \; B_n = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix},$$

where $A_n \in \mathbb{R}^{n \times n}$ and $B_n \in \mathbb{R}^n$. This assumption does not affect empirical performance measurements as the transformation that brings a system in the above form is system-dependent and, thus, can be computed offline just once. To generalize the assessment of performance, we generate the safe set as a random polytope of dimension n and we average the results over multiple runs. Moreover, we constraint our input to $[-0.5, 0.5]$ and the disturbance to $[-0.1, 0.1]$.

Scalability of implicit invariant sets We begin with the case of no disturbances. Fig. 4.7a and Fig. 4.7b show the times to compute the implicis CIS $\mathscr{C}_{xv,\mathrm{q}}$ for safe sets with $2n$ and n^2 constraints respectively. $\mathscr{C}_{xv,\mathrm{q}}$ can be computed in less than $0.5s$ for systems of size $n = 200$ when the safe set has $2n$ constraints, and in around $5s$ for $n = 100$ and safe sets with n^2 constraints, that is 10000 constraints in this example.

We now proceed to the case where system disturbances are present. In Fig. 4.8a and Fig. 4.8b, we observe that in the presence of disturbances computations are slower and, actually, are almost identical for different values of q. This is attributed to the presence of the Minkowsky difference in the closed-form expression (4.14) that dominates the runtime and depends on the nilpotency index of the system. Still, we are able to compute implicit RCISs in closed-form for systems with up to 20 states fairly efficiently in this experiment.



(a) (b)

Figure 4.8: Presence of disturbances. Safe sets with $2n$ constraints, $n \leq 20$. Computation times for Implicit RCISs. (a) Different levels q of the hierarchy. (d) Individual implicit RCIS, $\mathscr{C}_{xv,(\tau,\lambda)}$, for different values of (τ, λ).

The above results suggest the efficiency and applicability of our approach to scenarios involving online computations. Moreover, in our experience, the numerical result of a projection operation, depending on the method used, can be sometimes unreliable. Contrary to this, our closed-form implicit representation does not suffer from such drawback.

Quality of the computed sets and comparison to other methods We now compare our method with different methods in the literature, both in runtime and quality of the computed sets as measured by the percentage of their volume compared to the maximal (R)CIS. Even though, we already provided a comprehensive analysis in terms of runtime for our method, we still present a few cases for the shake of comparison. We compare our approach to the Multi-Parametric Toolbox (MPT3) [48] that computes the maximal (R)CIS, $\mathscr{C}_{max}$, the iterative approach in [113] that computes low-complexity (R)CISs, and the one in [67] that computes ellipsoidal CISs.

The runtimes of each method are reported in Fig. 4.9. The difficulty of computing $\mathscr{C}_{max}$ is apparent from the steep corresponding curve. The low-complexity methods in [113] and [67] are considerably faster, and [67] is slightly faster than even our implicit representation. However, our sets are superior in quality as we detail next.

First, in the absence of disturbances, the relative volume of the computed sets with respect to $\mathscr{C}_{max}$ is presented in Table 4.1. Since for $n \geq 7$ MPT3 does not terminate after several hours and the computed set before termination is not invariant, we present the relative volumes only for $2 \leq n \leq 6$. Our method returns a very close approximation of $\mathscr{C}_{max}$ even with small values of (τ, λ) and computes substantially larger sets compared to the other techniques. This supports our theoretical result in Corollary 4.12.1. In other words, our implicit representation retains the best out

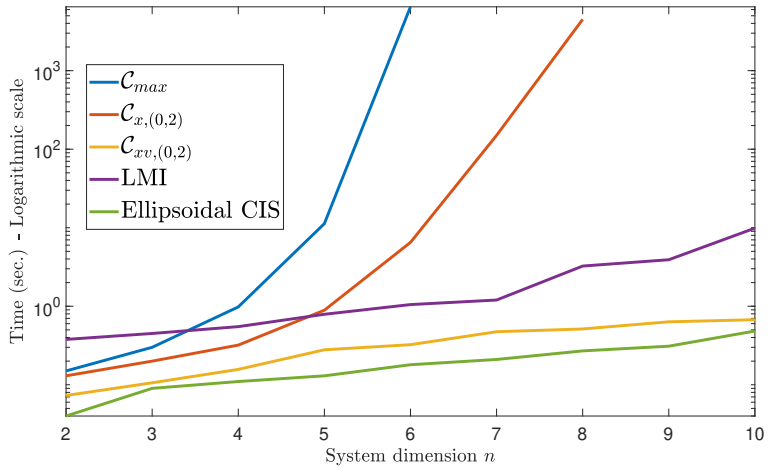


Figure 4.9: Computation times for $\mathscr{C}_{xv,(0,2)}$, its projection $\mathscr{C}_{x,(0,2)}$, the LMI method in [113], the ellipsoidal CIS in [67], and $\mathscr{C}_{max}$. Logarithmic scale. Note: [67] is evaluated in the absence of disturbances as it considers only nominal systems. For the other methods the performance without disturbance is similar or better.

Table 4.1: Absence of disturbances. Volume percentage with respect to the maximal CIS. Algorithms: Our method for different implicit CISs $\mathscr{C}_{xv,(\tau,\lambda)}$, the LMI method in [113], and the method in [67] computing ellipsoidal CISs. (S) denotes a singleton set.

	Our method		LMI method [113]	Ellipsoidal CIS method [67]
System dimension	$\mathscr{C}_{xv,(0,2)}$	$\mathscr{C}_{xv,(4,2)}$		
$n=2$	100	100	42.43	45.69
$n=3$	100	100	16.31	24.66
$n=4$	99.92	100	3.69	14.41
$n=5$	99.75	100	0.47	10.50
$n=6$	97.81	100	0 (S)	3.89

of two worlds: computational efficiency and close approximations of $\mathscr{C}_{max}$.

In the presence of disturbances, the results are similar and are reported in Table 4.2, where we omit [67] that is not designed for the case of disturbances. Theorem 4.12 proves that our method converges to its outer bound $\mathscr{C}_{outer,v}$. Here, we can appreciate that empirically $\mathscr{C}_{outer,v}$ approximates very closely $\mathscr{C}_{max}$, even in the presence of disturbances, based on the size of the sets computed by our method. However, the gap between $\mathscr{C}_{outer,v}$ and $\mathscr{C}_{max}$ depends on the size of the disturbance as shown next.

We illustrate how the size of the disturbance set affects our performance. We fix the safe set to be a random polytope in $\mathbb{R}^4$ and constraint the input to $[-0.5, 0.5]$. The disturbance set is

Table 4.2: Presence of disturbances. Volume percentage with respect to the maximal RCIS. Algorithms: Our method for different implicit RCISs $\mathscr{C}_{xv,(\tau,\lambda)}$ and the LMI method in [113]. (S) denotes a singleton set.

Volume (%)	Our method			LMI method [113]
System dimension	$\mathscr{C}_{xv,(0,2)}$	$\mathscr{C}_{xv,(2,2)}$	$\mathscr{C}_{xv,(4,2)}$	
$n=2$	100	100	100	31.99
$n=3$	98.24	99.67	99.96	16.35
$n=4$	99.02	99.42	99.88	4.36
$n=5$	98.75	99.74	99.81	3.64
$n=6$	91.17	96.07	97.91	0 (S)

Table 4.3: Increasing the size of the disturbance set $W=[-\overline{w},\overline{w}]$. Volume percentage of $\mathscr{C}_{x,(2,2)}$ with respect to the maximal RCIS and volume percentage of $\overline{S}_{xu}$ with respect to S_{xu}. (NE) set is nonempty. (E) set is empty.

$\overline{w}$	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40
$\frac{vol\ \mathscr{C}_{x,(2,2)}}{vol\ \mathscr{C}_{max}}$	99.9	99.7	99.3	98.1	91.8	10.5	$\mathscr{C}_x$ empty	$\mathscr{C}_{max}$ empty
$\frac{vol\ \overline{S}_{xu}}{vol\ S_{xu}}$	63.9	38.2	20.5	9.2	2.6	0.1	$\overline{S}_{xu}$ empty	$\overline{S}_{xu}$ empty
$\overline{S}_{xu} \cap \overline{\Delta}_{xu}$	NE	NE	NE	NE	NE	NE	E	E

$W=[-\overline{w},\overline{w}]$ and we increase $\overline{w}$ as in Table 4.3. Recall the nominal system $\overline{\Sigma}$ and the nominal safe set $\overline{S}_{xu}=S_{xu}-\overline{W}_{\infty}$, and let $\overline{\Delta}_{xu}$ be the set of fixed points of $\overline{\Sigma}$, which is in Brunovsky normal form. We can show that $\overline{\Delta}_{xu}=\{(x,u)\in\mathbb{R}^4\times\mathbb{R} | x_1=x_2=x_3=x_4=u\}$. As Table 4.3 details, by increasing the size of W the our RCIS shrinks at a faster rate compared to C_{max}, until finally $\overline{S}_{xu}$ is empty and, hence, does not contain any fixed points from $\overline{\Delta}_{xu}$. This is when the set we compute becomes empty as well.

4.7 Conclusion

In this chapter, we propose a novel approach to compute implicit RCISs in closed form. For systems without disturbances, we show that our approach is complete and converges to the maximal CIS exponentially fast as the parameterization size grows. Similarly, for systems with disturbances, our approach is weak complete and converges to a well-defined maximal set with the parameterization size. In addition, we show how to apply the implicit RCIS in safety supervisory control and safe planning, and demonstrate the effectiveness of our approach via several case studies.

One potential drawback of this approach is that the implicit RCIS can become overly conservative

when the disturbance set D becomes large, due to the open-loop nature of the eventually periodic control inputs used in the construction of the implicit RCIS. This issue can be alleviated by replacing the eventually periodic control inputs with a disturbance-feedback controller, presented in the next chapter.

CHAPTER 5

Automaton-based Implicit RCIS Computation for Discrete-Time Linear Systems

Several recent works [9, 10, 8, 119], including the one presented in Chapter 4, develop approaches for constructing implicit RCIS in closed form. To be precise, an implicit RCIS is a set in a lifting space whose projection onto the original state space gives an RCIS. By avoiding computing RCISs explicitly, these methods can work for systems with higher dimensions. It is indeed the case in many practical applications, such as model predictive control and supervision control, that knowledge of the explicit RCIS is not required and the implicit representation suffices.

Inspired by the recent progress on implicit RCISs, in this chapter we propose a novel approach to compute implicit RCISs for discrete-time linear systems. In addition, the aforementioned works consider non-measurable disturbances only, however, in many safety-critical applications, incoming disturbances can be measured in advance [120] and, hence, are considered measurable [90]. Thus, unlike the existing works, we develop a method that works for both measurable and non-measurable disturbances, achieved by introducing an automaton-based controller whose input is exactly the measurable disturbance. Specifically, our contributions include:

1) We propose a automaton-based method for computing *implicit RCISs* for a class of linear systems that contains the class of controllable linear systems, with measurable disturbances.

2) We derive conditions on the structure of the automaton such that the computation of the implicit RCIS is done exactly in closed-form.

3) We present a generic connection between measurable and non-measurable disturbances, enabling the proposed method to work with systems with non-measurable disturbances.

4) For systems with non-measurable disturbances, the proposed approach generalizes the method in Chapter 4. Numerical results show that the proposed approach significantly reduces the conservativeness inherent in the method in Chapter 4.

In addition to the above, we demonstrate the practicality of the implicit RCIS in the task of supervision control for the lane keeping problem. The goal is to modify nominal control inputs, as needed, to keep the system's trajectory within a set of safe states. We show that this is achieved by

solving a single optimization problem using the implicit RCIS.

Chapter Overview. In Section 5.1, the problem is mathematically set up, along with the essential definitions and assumptions. Section 5.2 lays down the ideas for computing an implicit RCIS for systems with measurable disturbances. Subsequently, Section 5.3 investigates when the implicit RCIS can be computed in closed-form, while Section 5.4 connects the proposed method to the case of non-measurable disturbances. Section 5.5 provides a computational evaluation of the proposed method. To keep a streamlined presentation, the proofs of all theorems are found in Appendix C.

5.1 Problem Setup

We consider a discrete-time linear system Σ:

$$\Sigma : x(t+1) = Ax(t) + Bu(t) + d(t), \tag{5.1}$$

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$, and disturbance $d \in D \subseteq \mathbb{R}^n$. The set D contains all possible values of d. The disturbance d is *measurable* if the measurement $d(t)$ is available before $u(t)$ is determined; otherwise, d is *non-measurable*. Unless specified explicitly, the disturbance is non-measurable by default in this dissertation.

Let $S \subseteq \mathbb{R}^{n+m}$ be the safe set of the system Σ. Depending on whether the disturbance is measurable or not, we define RCIS differently. The definition of RCIS for non-measurable disturbance is provided in Definition 2.3. We define RCIS for a system with measurable disturbance in the following.

Definition 5.1 (RCIS with measurable disturbance)**.** For the disturbance d being *measurable*, a set $C \subseteq \mathbb{R}^n$ is an RCIS for the system Σ within the safe set S if:

$$\forall x \in C, \forall d \in D, \exists u \text{ such that } (x,u) \in S, Ax + Bu + d \in C. \tag{5.2}$$

Note that the order of the quantifiers $\forall d \in D$ and $\exists u$ is swapped in Definitions 2.3 and 5.1, which means an RCIS C with respect to a non-measurable disturbance is guaranteed to be an RCIS for the same system with measurable disturbance but not vice versa. Finally, we call a set C_{max} the *maximal RCIS* within S if it is controlled invariant and contains every RCIS in S.

In the first part of this chapter, we focus on the computation of RCISs for systems with measurable disturbances. More specifically, we propose a method that computes the desired implicit representation for an RCIS in closed-form based on the following assumptions. In the second part, we extend our method to compute controlled invariant set for non-measurable disturbance. We make

the following assumptions throughout this chapter. The same assumptions are make in Chapter 4, where a discussion on their conservativeness can be found in Section 4.1.

Assumption 5.1. *The matrix A is nilpotent. That is, there exists an integer $h \leq n$ such that $A^h = 0$.*

Assumption 5.2. *The safe set $S \subset \mathbb{R}^{n+m}$ and the disturbance set $D \subset \mathbb{R}^n$ are both polytopes.*

The next theorem shows that to compute an RCIS for systems with a measurable disturbance we only need to consider the (finite) vertices of D.

Proposition 5.1. with a measurable disturbance $d \in D$, and let D_v be the set of vertices of D. Let Σ' be a system the same as Σ but with $d \in D_v$. Then, a convex set C is an RCIS for Σ if and only if C is an RCIS Σ'.

According to Proposition 5.1, given any polytopic disturbance set D, we can substitute D by th finite set D_v without loss of generality. Thus, for the remaining of this chapter, we directly assume that D is a finite set, as stated below.

Assumption 5.3. *The disturbance set $D \subset \mathbb{R}^n$ is given as a finite set of vertices.*

Problem 1: For a system Σ with measurable disturbance, a safe set S, and a disturbance set D satisfying Assumptions 5.1, 5.2, and 5.3, compute a convex implicit RCIS of Σ within S in closed form.

We end this section with a technical definition used later. Consider an autonomous system Σ_a:

$$\Sigma_a : x(t+1) = f(x(t), d(t)) \tag{5.3}$$

with state $x \in \mathbb{R}^n$ and a disturbance term $d \in D$. Let S be the safe set of Σ_a. The maximal RPIS of Σ_a within S is defined in Definition 2.5. The following proposition provides a equivalent characterization of the maximal RPIS.

Definition 5.2 (Reachable set for autonomous systems)**.** The *reachable set* $\mathscr{R}(\Sigma_a, x_0)$ of the autonomous system Σ_a from state x_0 is the set of all possible states that the system may visit. Formally, $x \in \mathscr{R}(\Sigma_a, x_0)$ if and only if there exists a trajectory $(x(t))_{t=0}^{K}$, $K \geq 0$, of Σ_a under disturbance sequence $(d(t))_{t=0}^{K-1} \in D^K$ with $x(0) = x_0$ and $x(K) = x$.

Proposition 5.2. The set $C = \{x \in \mathbb{R}^n \mid \mathscr{R}(\Sigma_a, x) \subseteq S\}$, i.e., the set of states whose corresponding reachable set is contained in the safe set S, is the maximal RPIS for Σ_a within S.

5.2 Controlled Invariant Set Computation Framework

The maximal RCIS for a system in (5.1) with measurable and/or non-measurable disturbances can be computed by standard iterative methods in Section 2.5, which is not guaranteed to terminate in finite time and does not scale to high-dimensional systems. To reduce the computation burden, many existing works close the loop with a linear feedback controller and then compute a RPIS of the closed-loop system as an under-approximation of the maximal RCIS. In this section, we extend this idea to a more general case: We close the loop with a parametrized nonlinear disturbance-feedback controller. Then, by computing a RPIS of an augmented closed-loop system, we search for the feasible initial states and controller parameters simultaneously such that the closed-loop trajectory satisfies the safety constraints. Lastly, we retrieve a RCIS from the RPIS of the augmented system.

First, we want to determine an appropriate controller structure. A popular choice in the literature is linear state feedback controllers, as linear structures make computation easier. But, if we ignore the computation tractability, would linear state feedback controllers be the optimal choice? We draw some inspirations from Definition 5.1: Given any RCIS C for Σ under measurable disturbances, by definition, there exists a *memoryless state-disturbance feedback controller* $u(t) = k(x(t), d(t))$ such that C is the maximal RPIS of the closed-loop system. In other words, any RCIS, including the maximal RCIS, is the maximal RPIS of a closed-loop system with respect to some memoryless state-disturbance controller. Thus, to minimize the conservativeness of the closed-loop RPIS, it is enough to consider the class of memoryless state-disturbance feedback controllers. Furthermore, it is well-known that any memoryless state-disturbance feedback controller is equivalent to a *disturbance feedback controller with memory*, explained by the following example.

Example 5.1. Consider a memoryless state-disturbance feedback controller $u(t) = k(x(t), d(t))$. This controller can be expressed in the form of (5.5) as:

$$\begin{cases} s(t+1) & = As(t) + Bk(s(t), d(t)) + d(t), \\ u(t) & = k(s(t), d(t)), \end{cases} \tag{5.4}$$

with $s(0) = x(0)$. That is, the internal dynamics of the controller forms a state estimator. ■

For above reasons, we consider a *parameterized disturbance-feedback controller with memory* Σ_c:

$$\Sigma_c : \begin{cases} s(t+1) & = \mathscr{T}(s(t), d(t); \theta), \\ u(t) & = o(s(t), d(t); \theta). \end{cases} \tag{5.5}$$

In the above, s is the internal state (memory) of the controller that distills useful information from

the disturbance input $d \in D$, and u is the output of the controller. The same d and u correspond to the disturbance and the control input of Σ respectively. The state transition function $\mathscr{T}$ and the output function o map the current state $s(t)$ and the disturbance input $d(t)$ into the next state $s(t+1)$ and the current output $u(t)$ respectively. Finally, θ is a constant vector that parameterizes the state transition function $\mathscr{T}$ and the output function o. The value of θ can depend on the initial state x_0 of the system Σ, such as in Example 5.1, $\theta = x(0)$. In what follows, we assume that the functions $\mathscr{T}$ and o in Σ_c are known. We discuss how to select $\mathscr{T}$ and o in the next section.

Closing the loop of Σ in (5.1) with the controller in (5.5), we obtain the following closed-loop system augmented with the controller internal state s and the constant vector θ:

$$\Sigma_{cl} : \begin{bmatrix} x(t+1) \\ \theta(t+1) \\ s(t+1) \end{bmatrix} = \begin{bmatrix} Ax(t) + Bo(s(t), d(t); \theta(t)) + d(t) \\ \theta(t) \\ \mathscr{T}(s(t), d(t); \theta(t)) \end{bmatrix}, \tag{5.6}$$

with the augmented state (x, θ, s) and the disturbance $d \in D$.

In the above augmented system we can calculate feasible initial states x_0 and controller parameters θ simultaneously. The control input $u(t)$ of Σ is equal to $o(x(t), d(t); \theta(t))$, as a function of the augmented state. Then, given the safe set S of the system Σ, the safe set for the closed-loop system is defined by:

$$S_{cl} = \{(x, \theta, s) \mid (x, o(s, d; \theta)) \in S, \forall d \in D)\}. \tag{5.7}$$

The following theorem connects the problem of computing an RCIS for the system Σ with the problem of computing an RPIS for the closed-loop system Σ_{cl}.

Theorem 5.3. *Let C_{cl} be an RPIS for the closed-loop system Σ_{cl} within the safe set S_{cl}. Then, the convex hull* $\mathrm{conv}(\pi_{[1,n]}(C_{cl}))$ *of the projection of C_{cl} onto the first n coordinates is a convex RCIS for the system Σ within the safe set S.*

According to Theorem 5.3, in order to compute RCISs for Σ, we first compute the maximal RPIS C_{cl} of the augmented system Σ_{cl} within S_{cl} and then compute the convex hull of its projection.

5.3 Closed-form Construction of Implicit RCISs

In the previous section we presented a framework for computing RCISs. Still, there are two main questions to be addressed. First, can we compute the maximal RPIS C_{cl} efficiently? Second, in practice the convex hull computation is expensive, can we avoid this operation?

Recall that in Section 5.2, we introduced the general controller structure in (5.5), but did not discuss how to select the functions $\mathscr{T}$ and o. In this section, we show that by carefully designing $\mathscr{T}$ and o we address both questions.

5.3.1 Computing C_{cl} in closed-form

From Proposition 5.2 we have that:

$$C_{cl} = \{(x, \theta, s) \mid \mathscr{R}(\Sigma_{cl}, (x, \theta, s)) \subseteq S_{cl}\}. \tag{5.8}$$

According to (5.8), if we express the reachable set $\mathscr{R}(\Sigma_{cl}, (x, \theta, s))$ in closed-form, then we obtain a closed-form expression of C_{cl}. The next theorem gives a sufficient condition for the reachable set $\mathscr{R}(\Sigma_{cl})$ to admit a closed-form expression.

Theorem 5.4. *Under Assumptions 5.1 and 5.3, if the state s of the controller belongs to a finite set Q, then given any initial state $(x, \theta, s) \in \mathbb{R}^{n+mL} \times Q$, the reachable set $\mathscr{R}(\Sigma_{cl}, (x, \theta, s))$ is finite. Moreover, it can be expressed in closed-form.*

Note that Theorem 5.4 is the only result that requires Assumption 5.1 in this chapter. Given Theorem 5.4, we want to design the functions $\mathscr{T}$ and o such that the internal state s of the controller belongs to a finite set Q. Recall that by Proposition 5.1 and Assumption 5.3, the input d of the controller also belongs to a finite set D. Thus, the controller Σ_c is a system with finite states and inputs. In the literature, this type of system is called a *mealy machine*, a special class of automata.

Definition 5.3 (Mealy Machine)**.** A *Mealy Machine* Σ_{fts} is a quintuple $(Q, D, \mathscr{T}, \Theta, o)$, where:

- Q is a finite set of discrete states;
- D is a finite set of actions;
- $\mathscr{T} : Q \times D \to Q$ is the state transition function that maps each state-action pair to the next state;
- Θ is a finite set of outputs;
- $o : Q \times D \to \Theta$ is the output function that maps each state-action pair to an element in the set Θ.

With slightly abusing notations, we denote both the action set of a mealy machine and the disturbance set of Σ by D, since in this chapter we only consider the disturbance set D as the action set of a mealy machine. The transition function $\mathscr{T}$ and the output function o are designed by the

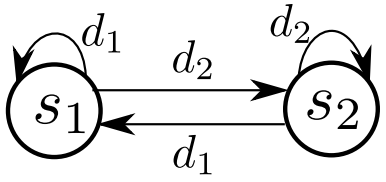


Figure 5.1: A toy mealy machine controller.

user. We parameterize the output set Θ of a mealy machine by the parameter vector θ: Suppose that $\Theta = \{u_i\}_{i=1}^{L}$, where each u_i is a vector of variables in $\mathbb{R}^m$. Given a vector $\theta = (\overline{u}_1, \cdots, \overline{u}_L) \in \mathbb{R}^{mL}$, we define a parametrized output function $o(s,d;\theta) : Q \times D \to \mathbb{R}^m$ such that $o(s,d;\theta) = \overline{u}_i$ for $o(s,d) = u_i$. A simple example of a mealy machine and the parametrized output function is shown below.

Example 5.2. Let $D = \{d_1, d_2\}$, d_1, $d_2 \in \mathbb{R}^n$, and $Q = \{s_1, s_2\}$. The state transition function $\mathscr{T}$ is shown in Figure 5.1. The output set is $\Theta = \{u_1, u_2\}$. The output function is $o(s_1,d_1) = o(s_2,d_1) = u_1$ and $o(s_1,d_2) = o(s_2,d_2) = u_2$. Let the controller parameter be $\theta = (\overline{u}_1, \overline{u}_2) = (1,2)$. Then, the parametrized output function is $o(s_1,d_1;\theta) = o(s_2,d_1;\theta) = 1$ and $o(s_1,d_2;\theta) = o(s_2,d_2;\theta) = 2$. ■

Next, we provide guidance on how to select $\mathscr{T}$ and o such that C_{cl} can be computed efficiently. Since $s \in Q$, with Q finite, we can decompose C_{cl} into $|Q|$ subsets:

$$C_{cl} = \bigcup_{s_i \in Q} C_{sub}(s_i) \times \{s_i\}, \tag{5.9}$$

where:

$$C_{sub}(s_i) = \left\{ (x,\theta) \in \mathbb{R}^{n+mL} \mid \mathscr{R}(\Sigma_{cl}, (x,\theta,s_i)) \subseteq S_{cl} \right\}. \tag{5.10}$$

For each state $(x',\theta,s') \in \mathscr{R}(\Sigma_{cl}, (x,\theta,s_i))$:

$$(x',\theta,s') \in S_{cl} \Leftrightarrow (x', o(s',d;\theta)) \in S,\ \forall d \in D. \tag{5.11}$$

Notice that x' and $o(s',d;\theta)$ are linear functions of x, θ, and $d \in D$. Then, the condition $(x', o(s',d;\theta)) \in S$ is a set of linear inequality constraints on (x,θ). Thus, by (5.11) and the fact that $\mathscr{R}(\Sigma_{cl}, (x,\theta,s_i))$ is finite, the set $C_{sub}(s_i)$ in (5.10) can be expressed by a set of linear inequality constraints, that is a polytope in $\mathbb{R}^{n+mL}$. We use the following example to illustrate the computation of $C_{sub}(s_i)$.

Example 5.3. Consider the mealy machine controller in Example 5.2. Suppose that the nilpotent matrix A of system Σ satisfies $A^2 = 0$. Then, the reachable set $\mathscr{R}(\Sigma_{cl}, (x, \theta, s_i))$ contains 7 elements, that is

$$\mathscr{R}(\Sigma_{cl}, (x, \theta, s_i)) = \left\{ \bigcup_{j,k=1,2} (ABu_j + Bu_k + Ad_j + d_k, \theta, s_k) \right\} \cup \left\{ (x, \theta, s_i) \right\} \cup \left\{ \bigcup_{j=1,2} (Ax + Bu_j + d_j, \theta, s_j) \right\},$$

where $\theta = (u_1, u_2)$. Suppose that the safe set is $S = \{(x,u) \in \mathbb{R}^{n+m} \mid G_x x + G_u u \leq h\}$, then:

$$\begin{aligned} &C_{sub}(s_1) = C_{sub}(s_2) = \\ &\quad \{(x,u) \in \mathbb{R}^{n+m} \mid & G_x x + G_u u_i \leq h, \\ &&G_x(Ax + Bu_j + d_j) + G_u u_j \leq h, \\ &&G_x(ABu_j + Bu_k + Ad_j + d_k) + G_u u_i \leq h, \\ &&\forall i, j, k \in \{1,2\}\}. \end{aligned}$$

Finally, $C_{cl} = \bigcup_{i=1,2} C_{sub}(s_i) \times \{s_i\}$.

So far we constructed C_{cl} in closed-form. However, to obtain an RCIS from C_{cl}, we have to project C_{cl} onto the first n coordinates and then compute the convex hull of the projected set. Both projection and convex hull operations are time consuming and thus undesirable. In what follows we derive an *implicit expression* of the resulting RCISs.

5.3.2 Implicit Controlled Invariant Set Expression (Method 1)

Assumption 5.4. *The safe set S of Σ is bounded.*

Given that S is bounded, the projection of $C_{sub}(s_i)$ onto the first n coordinates is also bounded. Thus, we can always find a large enough hypberbox B such that:

$$\pi_{[1,n]}(C_{sub}(s_i) \cap B) = \pi_{[1,n]}(C_{sub}(s_i)).$$

Denote the intersection of the hyperbox B and the polytope $C_{sub}(s_i)$ by $\overline{C}_{sub}(s_i) = B \cap C_{sub}(s_i)$. Note that the projection of C_{cl} is exactly the union of the projections of polytopes $\overline{C}_{sub}(s_i)$ for $s_i \in Q$, that

is:

$$\pi_{[1,n]}(C_{cl}) = \bigcup_{s_i \in Q} \pi_{[1,n]} \left(\overline{C}_{sub}(s_i) \right). \tag{5.12}$$

Furthermore, since the order of convex hull operation and the projection can be swapped, we have that:

$$\text{conv}(\pi_{[1,n]}(C_{cl})) = \pi_{[1,n]} \left(\text{conv} \left(\bigcup_{s_i \in Q} \overline{C}_{sub}(s_i) \right) \right). \tag{5.13}$$

Since $\overline{C}_{sub}(s_i)$ is a polytope, it can be written as:

$$C_{sub}(s_i) = \{(x, \theta) \mid G_i(x, \theta) \leq h_i\}.$$

Then, we construct the polytope:

$$\begin{aligned} C_\lambda = \Big\{ & \left(x, \theta, x_1, \theta_1, \cdots, x_{|Q|}, \theta_{|Q|}, \lambda_1, \cdots, \lambda_{|Q|}\right) \mid \\ & \lambda_i \geq 0, G_i(x_i, \theta_i) \leq \lambda_i h_i, \forall 1 \leq i \leq |Q|, \\ & \sum_{i=1}^{|Q|} \lambda_i = 1, \sum_{i=1}^{|Q|} x_i = x, \sum_{i=1}^{|Q|} \theta_i = \theta \Big\}. \end{aligned} \tag{5.14}$$

Under Assumption 5.4, given that $(x, \theta) \in R^{n+mL}$, we have:

$$\text{conv} \left(\bigcup_{s_i \in Q} \overline{C}_{sub}(s_i) \right) = \pi_{[1,(n+mL)]} (C_\lambda), \tag{5.15}$$

$$\pi_{[1,n]} \left(\text{conv} \left(\bigcup_{s_i \in Q} \overline{C}_{sub}(s_i) \right) \right) = \pi_{[1,n]}(C_\lambda). \tag{5.16}$$

By (5.13) and (5.16), the RCIS $\text{conv}(\pi_{[1,n]}(C_{cl}))$ is the projection of C_λ onto the first n coordinates. In other words, C_λ is an *implicit expression* of the RCIS $\text{conv}(\pi_{[1,n]}(C_{cl}))$.

Remark 5.1. Assumption 5.4 is only required if we want the equality $\pi_{[1,n]}(C_\lambda) = \text{conv}(\pi_{[1,n]}(C_{cl}))$ to hold. In the next section, we introduce an alternative implicit expression which does not need a bounded safe set S.

5.3.3 Implicit Controlled Invariant Set Expression (Method 2)

In Example 5.3, the projection of C_{cl} is already convex and, thus, the convex hull computation is omitted. It turns out that the convexity of $\pi_{[1,n]}(C_{cl})$ is not a coincidence. We define the nested state transition function by:

$$\mathscr{T}^*(s,(d(t))_{t=0}^k)=\begin{cases}\mathscr{T}(s,d(0)), & k=0,\\ \mathscr{T}(\mathscr{T}^*(s,(d(t))_{t=0}^{k-1}),d(k)), & k>0.\end{cases} \tag{5.17}$$

Similarly, the nested output function $o^*(s,(d(t))_{t=0}^k)$ is:

$$o^*(s,(d(t))_{t=0}^k)=\begin{cases}o(s,d(0)), & k=0,\\ o(\mathscr{T}^*(s,(d(t))_{t=0}^{k-1}),d(k)), & k>0.\end{cases} \tag{5.18}$$

Define a preorder relation "$\succeq$" on Q as follows. For any $s_1,s_2\in Q$, we have that $s_1\succeq s_2$ if for all $(d_1(t))_{t=0}^{k_1}\in D^{k_1}$ and $(d_2(t))_{t=0}^{k_2}\in D^{k_2}$ with non-negative integers $k_1,k_2\leq |Q|^2$, $o^*(s_1,(d_1(t))_{t=0}^{k_1})=o^*(s_1,(d_2(t))_{t=0}^{k_2})$ implies $o^*(s_2,(d_1(t))_{t=0}^{k_1})=o^*(s_2,(d_2(t))_{t=0}^{k_2})$.

Here, the "=" sign in $o^*(s_i,(d_1(t))_{t=0}^{k_1})=o^*(s_i,(d_2(t))_{t=0}^{k_2})$ is interpreted as the function o^* mapping two inputs to the same element in Θ (regardless of the parameter θ). Given the definition of the relation $\succeq$ on Q, we can algorithmically check if two states s and s' satisfy $s\succeq s'$ with worst case time complexity $O(|Q|^2)$.

Note that the "$\succeq$" relation is not a partial order as it does not satisfy the antisymmetry condition, namely it is possible to have $s_1\succeq s_2$ and $s_2\preceq s_1$ but $s_1\neq s_2$. However, the following theorem shows that the $\succeq$ relation in Q actually implies the partial order on the sets $\{\pi_{[1,n]}(C_{sub}(s))\}_{s\in Q}$ defined by the set inclusion.

Theorem 5.5. *Given the $\succeq$ relation defined on the set Q, for any two states s_1, $s_2\in Q$, $s_1\succeq s_2$ implies that $\pi_{[1,n]}(C_{sub}(s_1))\supseteq\pi_{[1,n]}(C_{sub}(s_2))$.*

We call a state $s_{max}\in Q$ a *maximal state* if for any $s'\in Q$, $s'\succeq s_{max}$ implies $s_{max}\succeq s'$, and call a state s_{dom} a *dominant state* if $s_{dom}\succeq s$ for all $s\in Q$. Denote Q_{max} as the set of all the maximal states in Q.

Corollary 5.5.1. Suppose that there exists a dominant state $s_{max}\in Q$. Then,

$$\pi_{[1,n]}(C_{cl})=\pi_{[1,n]}(C_{sub}(s_0)).$$

Corollary 5.5.1 explains our observation in Example 5.3.

Algorithm 4 Compute Implicit Controlled Invariant Set

inputs: Σ, S, $\Sigma_c = (Q, D, \mathscr{T}, \Theta, o)$.

if a dominant state $s_{dom} \in Q$ exists **then**
  Compute $C_{sub}(s_{dom})$ as in (5.10).
  return $C_{sub}(s_{dom})$.
else
  for $s_i \in Q_0 \subseteq Q$ **do**
    Compute $C_{sub}(s_i)$ as in (5.10).
  end for
  Compute C_λ as in (5.14).
  return C_λ.
end if

Example 5.4. For the mealy machine in Example 5.2, $s_1 \succeq s_2$ and $s_2 \succeq s_1$. Thus, both s_1 and s_2 are dominant states. Then, $\pi_{[1,n]}(C_{cl}) = \pi_{[1,n]}(C_{sub}(s_1)) = \pi_{[1,n]}(C_{sub}(s_2))$.

Corollary 5.5.2. Define a partition over Q_{max} as follows: For s and $s' \in Q$, s and s' belong to the same component if $s \succeq s'$ and/or $s' \succeq s$. Let a set $Q_0 \subseteq Q_{max}$ contain exactly one state from each component of this partition. Then, $\pi_{[1,n]}(C_{cl}) = \cup_{s_{max} \in Q_0} \pi_{[1,n]}(C_{sub}(s_{max}))$.

According to Corollary 5.5.1, if a dominant state s_{dom} exists in Q, the RCIS $\text{conv}(\pi_{[1,n]}(C_{cl}))$ is simply the projection of $C_{sub}(s_{dom})$ onto the first n coordinates. In this case, we can directly take $C_{sub}(s_{dom})$ as the implicit representation of the RCIS $\text{conv}(\pi_{[1,n]}(C_{cl}))$; otherwise, by Section 5.3.2, we construct C_λ as the implicit RCIS. Note that according to Corollary 5.5.2, we can replace Q by Q_0 in the definition of C_λ. The overall procedure of computing implicit RCISs is summarized in Algorithm 4.

5.3.4 Classes of Mealy Machines with Dominant States

In this subsection, we present three classes of mealy machines with at least one dominant state, which we use later to construct implicit RCISs in case studies. It is important to note, however, that there exists many mealy machines with dominant states that do not belong to those three classes.

Furthermore, within this subsection, we demonstrate that when Method 2 (outlined in Section 5.3.3) is employed with a specific class of mealy machine controllers, it yields the construction of the implicit RCIS $\mathscr{C}_{xv,(\tau,\lambda)}$ in Chapter 4. This observation underscores that the method presented in Chapter 4 can be regarded as a distinct subset of Method 2, as described in Section 5.3.3.

Simple Loop Given a positive integer $\lambda > 0$, let $Q = \{s_i\}_{i=1}^{\lambda}$ and $\Theta = \{u_i\}_{i=1}^{\lambda}$. Define, for all $d \in D$, the state transition and output functions as:

$$\mathscr{T}(s_i,d) = \begin{cases} s_{i+1} & i < \lambda, \\ s_1 & i = \lambda. \end{cases}, \; o(s_i,d) = \begin{cases} u_{i+1} & i < \lambda, \\ u_1 & i = \lambda. \end{cases} \tag{5.19}$$

For such a structure, any $s \in Q$ is a dominant state.

Simple Loop with a Tail Given two integers $\tau \geq 0$ and $\lambda > 0$, let $Q = \{s_i\}_{i=1}^{\tau+\lambda}$ and $\Theta = \{u_i\}_{i=1}^{\tau+\lambda}$. Define, for all $d \in D$, the state transition and output functions as:

$$\mathscr{T}(s_i,d) = \begin{cases} s_{i+1} & i < \tau+\lambda, \\ s_{\tau+1} & i = \tau+\lambda. \end{cases}, \; o(s_i,d) = \begin{cases} u_{i+1} & i < \tau+\lambda, \\ u_{\tau+1} & i = \tau+\lambda. \end{cases} \tag{5.20}$$

For such a structure, the domain state is s_1. Note that a simple loop structure is a special case of a simple loop with a tail ($\tau = 0$). Since all the transitions in a simple loop with a tail structure is independent of the disturbance measurement, when a system with measurable disturbance is equipped with a controller in such a structure, it can be shown that the resulting implicit RCIS $C_{sub}(s1)$ is also controlled invariant for the same system with non-measurable disturbance. In fact, the resulting implicit RCIS $C_{sub}(s1)$ is exactly equal to the implicit RCIS $\mathscr{C}_{xv,(\tau,\lambda)}$ in Section 4.3 of Chapter 4. In other words, the method for constructing implicit RCISs in Chapter 4 is a special case of Method 2 described in Section 5.3.3.

Tree Structure Suppose the cardinality of the disturbance set $|D| = K$. Given an integer $L > 0$, define $N = (K^L - 1)/(K-1)$. Let $Q = \{s_0\} \cup \bigcup_{i=1}^{l} D^L$. That is, Q is the union of s_0 and all finite sequences of elements in D with length less than or equal to L. We assign one output for each $s \in Q$ denoted by $u(s)$. Thus, $\Theta = \{u(s)\}_{s \in Q}$. The state transition function is defined as for all $d \in D$:

$$\mathscr{T}(s,d) = \begin{cases} d & s = s_0, \\ sd & s \in D^k, k < L, \\ s(2:L)d & s \in D^L, \end{cases} \tag{5.21}$$

where $sd \in D^{k+1}$ denotes the concatenation of $s \in D^k$ and $d \in D$, and $s(2:L)d$ denotes the concatenation of the subsequence $s(2:L)$ of s and $d \in D$. For instance, if $s = d_1 d_2 \cdots d_L$, then $s(2:L)d = d_2 \cdots d_L d$.

For $L = K = 2$, the state transition function is shown in Figure 5.2. We call this class of mealy

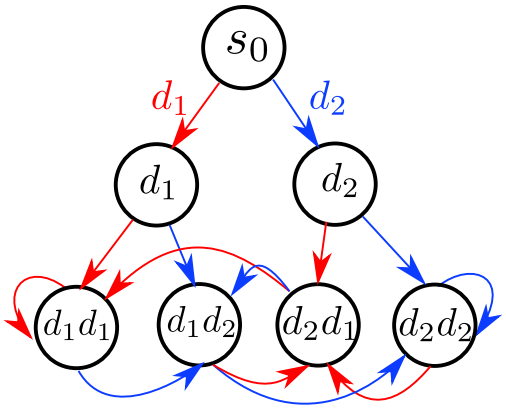


Figure 5.2: The tree-structure mealy machine ($L = 3$, $D = \{d_1, d_2\}$). The red arrow and blue arrow indicate transitions under d_1 and d_2 respectively.

machines *tree structure* since the mealy machine transition graph, as shown in Figure 5.2, embeds a tree with s_0 the root node.

Given the state transition function, the output function is simply defined as:

$$o(s,d) = u(\mathscr{T}(s,d)). \tag{5.22}$$

For any tree-structure mealy machine, s_0 is the dominant state. Intuitively, the tree-structure mealy machine memorizes the past L disturbance measurements and assigns a control input to each possible combination of the past L disturbances.

Finally, for the three classes of mealy machines introduced here, it can be proven that by increasing the number of discrete states (complexity), that is increasing L, τ, or λ, we tend to obtain larger RCISs.

5.4 Bridge Measurable and Non-measurable Disturbances

According to the discussion of Section 5.3.4, one way to handle non-measurable disturbance is to use Method 2 in Section 5.3.3 with a mealy machine controller whose transitions are independent of the disturbance measurement, such as the simple loop with tail structure. In this case, the proposed method degrades to the method in Chapter 4, which yields implicit RCISs for non-measurable disturbance. But this workaround is conservative since we restrict the method to only use open-loop controllers to construct implicit RCISs. In this section, we show a generic connection between measurable and non-measurable disturbance, which enables our method to compute RCISs for systems with any type of disturbances using any mealy machine controller.

Suppose a system Σ in (5.1) has a non-measurable disturbance $d \in D$. We construct a system Σ'

with a measurable disturbance by adding a one-step delay:

$$\Sigma': \begin{bmatrix} x(t+1) \\ u(t+1) \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} + \begin{bmatrix} 0 \\ I_m \end{bmatrix} v(t) + \begin{bmatrix} I_n \\ 0 \end{bmatrix} d(t), \tag{5.23}$$

with state $(x,u) \in \mathbb{R}^{n+m}$, input $v \in \mathbb{R}^m$, a measurable disturbance $d \in D \subseteq \mathbb{R}^n$, and I_n, I_m being the $n \times n$ and $m \times m$ identity matrices respectively.

Let the safe set of Σ be $S \subset \mathbb{R}^{n+m}$. We want to compute a RCIS for Σ within S. The next theorem reveals that this can be achieved by computing a RCIS for Σ' within $S \times \mathbb{R}^m$.

Theorem 5.6. *Given the systems Σ in* (5.1) *and Σ' in* (5.23)*, if C' is an RCIS for Σ' in $S \times \mathbb{R}^m$, the projection $\pi_{[1,n]}(C')$ of C' onto the first n coordinates is an RCIS for Σ in S.*

If C' is the maximal RCIS for Σ' in $S \times \mathbb{R}^m$, then $\pi_{[1,n]}(C')$ is the maximal RCIS for Σ in S.

Thanks to Theorem 5.6, in terms of computing RCISs, any method designed for measurable disturbances can be applied to systems with non-measurable disturbances.

5.5 Case Studies

5.5.1 Lane Keeping Supervision

Consider a 4-dimensional linearized bicycle vehicle dynamics with respect to a constant longitudinal velocity $30m/s$ in [112], discretized with time step $\Delta t = 0.1s$. The system states consist of the lateral displacement y, lateral velocity v, yaw angle $\Delta\Psi$ and yaw rate r. The control input u is the steering angle. The disturbance is $d = (0, 0, -r_d \Delta t, 0)$, where $r_d \in \mathbb{R}$ is the road curvature within a range $|r_d| \leq r_{d,max}$. The safe set is given by constraints $|y| \leq 0.9$, $|v| \leq 1.2$, $|\Delta\Psi| \leq 0.05$, $|r| \leq 0.3$ and $|u| \leq \pi/2$.

The future road curvature can be measured in advance and thus d is a measurable disturbance [120]. We compare our method with the method proposed in Chapter 4, LMI-based low-complexity RCIS in [113], full-complexity RCIS in [43] and the maximal RCIS. Our method uses the tree structure with $L = 4$ in Section 5.3.4 as the mealy machine controller. For the method in Chapter 4, we use $\mathscr{C}_{xv,(\tau,\lambda)}$ defined in Section 4.3 with $\tau = 0$ and $\lambda = 14$ as the implicit RCIS, with is equivalent to our method equipped with simple loop controller with parameter $\lambda = 14$. For the LMI-based method in [113], we set the parameter $\rho = 1$ and run the iterative algorithm until convergence. The methods in [8], [113] consider non-measurable disturbances only. To make a fair comparison, our method computes RCISs for d being measurable and/or non-measurable respectively. We evaluate the algorithm performance by their computation time and the volume percentage of the resulting

Table 5.1: Computation Time and Volume Percentage of Computed RCIS to the maximal RCIS. (Lane Keeping)

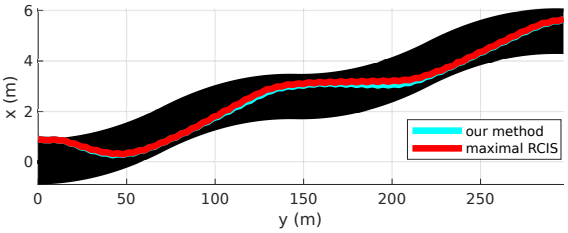
$r_{d,max}$		0.01	0.015	0.03	0.05	0.07
Our method (d meas.)	Time (s)	0.042	0.035	0.037	0.035	0.032
	Vol (%)	100.00	99.99	99.89	98.91	74.75
Our method (d non-meas.)	Time (s)	0.071	0.062	0.072	0.063	0.060
	Vol (%)	100.00	100.00	100.00	99.89	0
Method in Chapter 4	Time (s)	0.506	0.443	0.404	0.397	0.484
	Vol (%)	99.82	87.64	0	0	0
LMI Method [113] ($\rho = 1$)	Time (s)	0.449	0.519	0.562	0.500	0.564
	Vol (%)	0	0	0	0	0
Maximal RCIS	Time (s)	13.084	18.918	15.525	15.698	21.513

RCISs to the maximal RCIS. The volume percentage is estimated by monte carlo method with sample size $N = 10^4$.

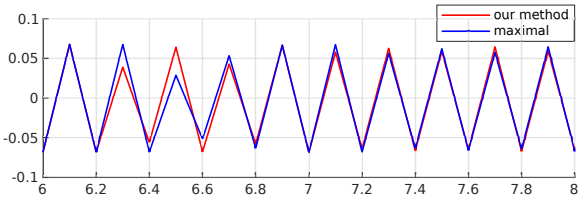
The comparison results are shown in Table 5.1: According to the 2nd, 3rd and 4th rows of Table 5.1, when dealing with non-measurable disturbances only, our method outperforms the method in Chapter 4 and LMI-based method in [113] in both the computation time and the volume of the resulting RCIS for all $r_{d,max}$, showing a strong robustness to non-measurable disturbances. The LMI-based method encounters an infeasible optimization problem in all test cases and thus has 0 volume percentage. The method in Chapter 4, as a special case of the proposed method, has a decent volume percentage when the disturbance range is small. But as $r_{d,max} > 0.015$, the implicit RCIS $\mathscr{C}_{xv,(0,14)}$ in Chapter 4 becomes empty, while our method still has volume percentage greater than 98% for both measurable and non-measurable cases. It is worth recalling that, by Theorem 4.11, the emptiness of $\mathscr{C}_{xv,(0,14)}$ implies the emptiness of any implicit RCIS generated using the method in Chapter 4. This observation underscores the significantly reduced conservativeness inherent in the method introduced in this chapter when compared to the one in Chapter 4.

Shown by the first 2 rows of Table 5.1, when $r_{d,max} = 0.7$, our method returns an nonempty RCIS for d being measurable, but returns an empty RCIS for d being non-measurable. Thus, by considering d as a measurable disturbance, our method is robust to a larger range of disturbances. Finally, comparing the first 2 rows with the last row of Table 5.1, when $r_{d,max} < 0.07$, our method computes implicit RCISs with almost the same size as the maximal RCISs, using less than 0.3% computation time of the maximal RCISs.

Next, we illustrate how the computed implicit RCIS can be used to supervise a nominal controller. Suppose the current state $x(t)$ belongs to the RCIS $\pi_{[1,n]}(C_{sub}(s_0))$. Given the nominal steering input $u_d(t)$ and the disturbance $d(t)$ at time t, we minimally change the input $u_d(t)$ such that the



(a) Vehicle maneuvers



(b) Steering inputs

Figure 5.3: (a) Vehicle maneuvers under control inputs supervised by our implicit RCIS (cyan curve) and the maximal RCIS (red curve). The black region indicates the road surface. (b)Vehicle steering inputs supervised by our implicit RCIS (blue curve) and the maximal RCIS (red curve) ($6 \leq t \leq 8$).

next state $x(t+1)$ stays in the RCIS $\pi_{[1,n]}(C_{sub}(s_0))$ by solving the following linear program:

$$\begin{aligned} &\min_{u(t),\theta} \|u_d(t) - u(t)\|_2^2 \\ &\text{subject to } (Ax(t) + Bu(t) + d(t), \theta) \in C_{sub}(s_0), \end{aligned} \tag{5.24}$$

where A, B are the system matrices and θ is a slack variable. We use the solution $u(t)$ of (5.24) as the actual steering input to the vehicle. The feasibility of (5.24) is guaranteed since $\pi_{[1,n]}(C_{sub}(s_0))$ is a RCIS and $x(t) \in \pi_{[1,n]}(C_{sub}(s_0))$.

We compare the corrected inputs when using our method to the ones obtained based on the maximal RCIS C_{max} via the following linear program:

$$\begin{aligned} &\min_{u(t)} \|u_d(t) - u(t)\|_2^2 \\ &\text{subject to } Ax(t) + Bu(t) + d(t) \in C_{max}. \end{aligned} \tag{5.25}$$

The nominal controller is $u_d(t) = -0.1812y - 0.0373v - 4.5996\Delta\Psi - 0.6649r$. We run two simulations with the same initial states and control inputs obtained from (5.24) and (5.25) respectively ($r_{d,max} = 0.015$). As shown in Figures 5.3a and 5.3b, the vehicle maneuvers and steering inputs supervised by our implicit RCIS $C_{sub}(s_0)$ and the maximal RCIS are very close to each other. The maximal difference between the control inputs from (5.24) and (5.25) is around 0.035 at $t = 6.5s$. This observation is consistent with the results shown in Table 5.1, where the volume of our implicit RCIS is approximately 99.96% of the volume of the maximal RCIS.

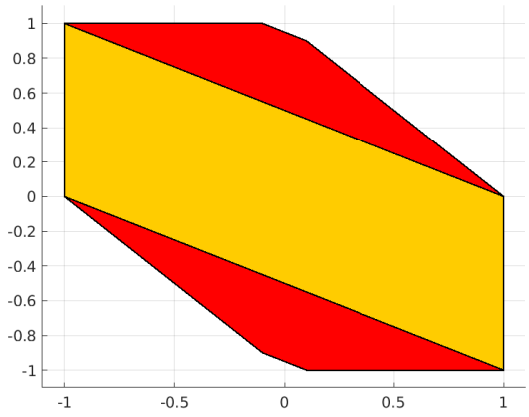


Figure 5.4: RCISs for double integrator. Yellow: RCIS from LMI-based method [113]. Red: the maximal RCIS computed by both our method and [8].

5.5.2 Chain of Integrators

Consider a discrete-time n-th order integrator:

$$x(t+1) = \left(I_n + \begin{bmatrix} 0 & I_{n-1} \\ 0 & 0 \end{bmatrix} \right) x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (u(t) + d(t)) \tag{5.26}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}$ and $d \in \mathbb{R}$. I_n indicates the identity matrix in $\mathbb{R}^{n \times n}$. d is considered as a measurable disturbance within range $|d| \leq 0.1$. The safe set is $S = \{(x,u) \mid |x_i| \leq 1, \forall i = 1,\ldots,n,\ |u| \leq 1\}$.

The comparison results of our approach (tree structure, $L = 4$) with the method in Chapter 4 ($\mathscr{C}_{\tau,\lambda}$ with $\tau = 0$ and $\lambda = 14$) and the LMI-based method in [113] ($\rho = 1$) are shown in Table 5.2. For $n \leq 4$, our method outperforms the other 2 methods in computation time and volume percentage. For $n = 2$, our method returns exactly the maximal RCIS, depicted in Fig. 5.4. For $n \geq 6$, the maximal RCIS does not terminate within 1 hour. Thus we only check if the computed RCISs are empty or not instead of comparing their volume to the maximal RCIS. When $n \geq 6$, our method is the only one that returns non-empty RCISs. Note that even though the implicit RCIS has closed-form expression, the number of constraints in the implicit RCIS grow exponentially as n increases. In this example, for $n = 10$, it takes about 339s for our method to generate the implicit RCIS, which is a polytope in $\mathbb{R}^{24}$ with about 36×10^4 constraints.

5.5.3 Truck with N Trailers

Consider a continuous-time model for a truck with N trailers [102]. The state consists of the $N+1$ velocity values, each for the truck and the N trailers, and the N spring elongations in between them.

Table 5.2: Computation Time and Volume Percentage of Computed RCIS to the maximal RCIS (Chain of Integrators).

n		2	4	6	8	10
Our method (d meas.)	Time (s)	0.005	0.025	0.368	9.287	339.060
	Vol (%)	100	98.79	>0	>0	>0
Method in Chapter 4	Time (s)	0.183	0.341	2.420	7.168	37.105
	Vol (%)	74.49	0	0	0	0
LMI Method [113] ($\rho=1$)	Time (s)	3.465	0.603	0.952	1.405	3.1402
	Vol (%)	66.85	≈ 0	0	0	0
Maximal RCIS	Time (s)	0.734	11.114	>3600	>3600	>3600

Table 5.3: Computation Time and Volume Percentage of Computed RCIS to the maximal RCIS. (Truck with N trailers)

System dimension		$n=3$	$n=5$	$n=7$	$n=9$
Our method	Time (s.)	0.109	0.781	8.669	163.1
	Vol (%)	100	100	>0	0
Method in Chapter 4	Time (s.)	0.547	0.814	1.352	6.577
	Vol (%)	100	98.90	0	0
Maximal RCIS	Time (s.)	0.746	13.76	>3600	>3600

Hence, N trailers correspond to dimension $n = 2N+1$. The input is the velocity of the truck. We discretize the model with a sampling time of T_s seconds assuming piecewise constant inputs.

Table 5.3 shows the results of this case study for our method and the method in Chapter 4 ($\mathscr{C}_{\tau,\lambda}$ with $\tau = 0$ and $\lambda = 14$). For $n \geq 7$ the method computing the maximal RCIS does not terminate after 1 hour, and, hence, we only check non-emptiness of sets instead of volume percentage. When the maximal RCIS is computed, we see that our approach covers it, but due to the implicit representation the running times are much faster. However, we see that in this example, after some point, as the dimension becomes large, the set our algorithm returns is empty. This can be understood as by adding more trailers the noise from each spring compounds towards the ones behind it, resulting in the shrinking the RCIS.

5.6 Conclusion

In this chapter, we derive closed-form expressions for implicit RCISs for discrete-time linear systems with measurable disturbances. In particular, a disturbance-reactive (or disturbance feedback) controller in the form of a parametrized finite automaton is considered. We show that, for a class of automata, the RPISs of the corresponding closed-loop systems can be expressed by a set of linear inequality constraints in the joint space of system states and controller parameters. This leads to an implicit representation of the invariant set in a lifted space. We further show how the same parameterization can be used to compute invariant sets when the disturbance is not measurable.

Also, we show that the method proposed in this chapter generalizes the method in Chapter 4. The idea of using disturbance-feedback controllers for constructing implicit RCISs allows us

to significantly reduce the conservativeness inherent in the method in Chapter 4. In particular, numerical examples show that the proposed method can yield nonempty implicit RCIS for non-measurable disturbance while the method in Chapter 4 fails. This underscores the considerable improvement and practical applicability of the method introduced in this chapter.

CHAPTER 6

Scalable Computation of RCISs for Discrete-time Linear Systems with Input Delays

One critical application of RCISs is to synthesize safety controllers and/or safety supervisors for autonomous vehicles [91, 112]. While there exist simple linear dynamical models for vehicle control, when deploying such controllers on actual vehicles [92], we have realized that there is a non-negligible time delay on the input signal. This delay may arise from the inherent dynamics of low-level actuators or communication bottlenecks within the vehicle's system. To address this input delay, we must extend the original system by incorporating additional states corresponding to these delayed inputs, resulting in a high-dimensional augmented system. Computing invariant sets for such high-dimensional systems is challenging. Intriguingly, this augmented system is very structured and our goal in this chapter is to exploit this structure to compute invariant sets for systems with input delays in a scalable manner.

Another concern for systems with input delays is that when the system is subject to additional disturbances, it becomes harder to guarantee invariance with a delayed input. To mitigate this challenge, one effective approach is to incorporate preview information on external disturbances into the control framework, a concept known as preview control [97]. In the latter part of this chapter, we explore the integration of such preview information into our invariant set computation framework while preserving scalability.

The main contributions of this chapter are summarized below:

- We construct a delay-free auxiliary system by predicting the future states in τ steps (where τ is the input delay), and then show that the computation of the maximal RCIS for the high-dimensional equivalent of the delayed system can be reduced to the computation of the maximal RCIS of the low-dimensional auxiliary system and $(\tau + 2)$ set intersection operations.

- We extend the proposed method for systems with both input delay and disturbance preview, where the preview on disturbance can mitigate the difficulty in controlling systems with large

input delays.

- We provide two examples to show the efficiency and utility of the proposed method.

Chapter Overview. We introduce the problem statement in Section 6.1. Then, we present our approach for systems with input delays in Section 6.2 and its extension to systems with both input delay and disturbance preview in Section 6.3. Section 6.4 illustrates the effectiveness of our approach with numerical examples. We conclude this chapter in Section 6.5.

Notation. The Minkowski sum of a collection of sets $\{S_i\}_{i\in I}$ is denoted by $\sum_{i\in I} S_i$. Note that we apply the convention that if $b < a$, the Minkowski sum $\sum_{i=a}^{b} S_i = \emptyset$.

6.1 Problem Setup

Consider a linear system with τ-step pure delay in control input, that is

$$\Sigma_{delay} : x(t+1) = Ax(t) + Bu(t-\tau) + Fd(t) \tag{6.1}$$

with state $x(t) \in \mathbb{R}^n$, input $u(t) \in \mathbb{R}^m$, and disturbance $d(t)$ in a polytopic set $D \subset \mathbb{R}^l$. Let the safe set of the system be $S_{xu} = X \times U$, with $X \subseteq \mathbb{R}^n$ and $U \subseteq \mathbb{R}^m$ both polytopes. We can rewrite the time-delayed system Σ_{delay} as a linear system without input delay, by appending the past τ-step inputs to the state space. The resulting $(n+m\tau)$-dimensional augmented system takes the form:

$$\Sigma_{aug} : \begin{cases} x(t+1) &= Ax(t) + Bu_1(t) + Fd(t) \\ u_1(t+1) &= u_2(t) \\ u_2(t+1) &= u_3(t) \\ &\vdots \\ u_\tau(t+1) &= u(t). \end{cases} \tag{6.2}$$

If we want the state trajectories of the system Σ_{delay} to remain within the safe set $X \times U$, this is the same as asking the trajectories of the augmented system Σ_{aug} to remain in the safe set $S \times U$, with $S := X \times U^\tau$. Therefore, one can state the invariance problem for a system with input delay in terms of the system Σ_{aug} as follows.

Problem 6.1. *Find the maximal RCIS of the system in Σ_{aug} subject to the safe set $S \times U$.*

According to Section 2.5, Problem 6.1 can, in principle, be solved by the outside-in algorithm (Alg. 1). However, it is well-known that the outside-in algorithm suffers from the curse of

dimensionality. Since the dimension of the augmented system increases linearly with the delay steps τ, the outside-in algorithm becomes computationally intractable very soon as τ increases. In what follows, we propose a method to solve Problem 6.1 whose complexity is independent of τ. For the remainder of this chapter, we make the following technical assumption:

Assumption 6.1. *The k-step BRS of S subject to the system Σ_{aug} and safe set $S \times U$ converges to the maximal RCIS of Σ_{aug} subject to $S \times U$ as k goes to infinity.*

Assumption 6.1 is guaranteed if the sets X, U, and D are compact [20]. Under Assumption 6.1, the following proposition presents a key property of the maximal RCIS, later used to prove our main results.

Proposition 6.1. Consider a system Σ as in (2.1) with safe set S_{xu}. Let C_k be the k-step BRS of X, with X the projection of S_{xu} onto the state space. Suppose $C_{max} = \bigcap_{k\geq 0} C_k$. Then for all $x \notin C_{max}$, there exists a non-negative integer $N < \infty$ such that if the system starts at x, the system can be forced to violate the state-input constraint $(x,u) \in S_{xu}$ by disturbances in at most N steps.

Proof. Define $W_k = \mathbb{R}^n \backslash V_k$ for $k \geq 0$ and $W_\infty = \cup_{i=1}^{\infty} W_k$. Then $W_0 = \mathbb{R}^n \backslash X$ is the unsafe set, and as the complement of V_k, W_k is the maximal set of states that if the system starts from W_k, there exists disturbances that can force the system state to violate the state-input constraint $(x,u) \in S_{xu}$ in at most k steps. By De Morgan's law, $W_\infty = X \backslash V_\infty$. Now pick $x \notin C_{max} = V_\infty$. Then $x \in W_\infty = \cup_{i=1}^{\infty} W_k$, which implies that there exists a $N < \infty$ such that $x \in W_N$. As discussed above, if the system starts from state $x \in W_N$, disturbance can force the system state to reach W_0 in at most N steps. □

6.2 State Prediction and Prediction Dynamics

Our solution approach relies on the construction of a reduced-order delay-free auxiliary dynamics with state space dimension the same as $x(t)$ in (6.1). We then show that the maximal RCIS of the system Σ_{aug} subject to S can be reconstructed from the maximal RCIS of the delay-free dynamics within a modified safe set. Since the dimension of new dynamics does not explode as the delay step τ increases, the proposed method is computationally more efficient than directly computing the maximal RCIS of (6.2).

Our method is inspired by the following observation. We denote the maximal RCIS of the system without delay (that is, $\tau = 0$) by C. Also, suppose that there is a sensor that can measure the future τ-step values of the disturbance d. Then at each time t, the exact state evolution in τ steps, namely $x(t+1:t+\tau)$, can be calculated based on the measurement of $x(t)$, future disturbance $d(t:t+\tau-1)$ and the extended states $u_{1:\tau}(t)$, which essentially correspond to past inputs. Then for

the dynamics (6.2), as long as $x(t+\tau) \in C$, we can pick $u(t)$ such that $x(t+\tau+1) \in C$. It can be shown that

$$\exists u(t) \text{ s.t. } x(t+\tau) \in X, \forall t \geq 0 \Longleftrightarrow x(\tau) \in C, \tag{6.3}$$

which is very close to our goal (that is, $\exists u(t)$ s.t. $x(t) \in X$ for $t \geq 0$) except that $x(t)$ with $t \in [0, \tau-1]$ can be out of X. Moreover, by definition, $d(t : t+\tau-1)$ is not accessible at time t. Nevertheless, we can predict the state evolution by assuming the future disturbances to be zero. The question arises as to whether a result akin to (6.3) exists, and the affirmative answer is indeed the case. We are going to show this result in the rest of this section.

First, we expand Σ_{aug} in τ steps to obtain the exact expression of $x(t+\tau)$ as

$$x(t+\tau) = A^{\tau}x(t) + \sum_{i=1}^{\tau} A^{i-1}Bu_{\tau-i+1}(t) + \sum_{i=1}^{\tau} A^{i-1}Fd(t+\tau-i). \tag{6.4}$$

Since $x(t)$ and $u_{1:\tau}(t)$ are known at time t, we define

$$\widehat{x}_{\tau}(t) = A^{\tau}x(t) + \sum_{i=1}^{\tau} A^{i-1}Bu_{\tau-i+1}(t), \tag{6.5}$$

as a prediction of $x(t+\tau)$ based on the state measurements of dynamics Σ_{aug} at time t. Define the polytope $D_{\tau} = \sum_{i=1}^{\tau} A^{i-1}FD$, which is the exact bound of the prediction error $(x(t+\tau) - \widehat{x}_{\tau}(t))$. Note that D_{τ} is time invariant and can be computed offline. Then, for all $t \geq 0$, we have the following inclusion relation:

$$x(t+\tau) \in \widehat{x}_{\tau}(t) + D_{\tau} \doteq \{\widehat{x}_{\tau}(t) + d \mid d \in D_{\tau}\}, \tag{6.6}$$

which implies the following statement:

$$\widehat{x}_{\tau}(t) + D_{\tau} \subseteq X, \forall t \geq 0 \Rightarrow x(t+\tau) \in X, \forall t \geq 0. \tag{6.7}$$

Therefore, if there exists a controller $u(t)$ such that $\widehat{x}_{\tau}(t) \in X - D_{\tau}$ for all $t \geq 0$, such a controller guarantees that $x(t+\tau) \in X$ for all $t \geq 0$. According to the analysis above, it is important to understand the relation between $\widehat{x}_{\tau}(t)$ and $\widehat{x}_{\tau}(t+1)$. By definition and after some simple algebra,

we have Σ_{aux} defined by

$$\widehat{x}_\tau(t+1) = A^\tau x(t+1) + \sum_{i=1}^{\tau} A^{i-1} B u_{\tau-i+1}(t+1) \tag{6.8}$$

$$= A\widehat{x}_\tau(t) + Bu(t) + A^\tau F d(t), \tag{6.9}$$

with disturbance $d(t) \in D$. Thus, the problem becomes: Given $\widehat{x}_\tau(t) \in X - D_\tau$, find a $u(t) \in U$ so that for all $d(t) \in D$, $\widehat{x}_\tau(t+1) \in X - D_\tau$. All that we need is to compute the maximal RCIS $\widehat{C}$ of Σ_{aux} subject to the safe set $(X - D_\tau) \times U$. This computation can be done using the outside-in algorithm. Since the dimension of Σ_{aux} is equal to the dimension of x in (6.1), the complexity is not directly affected by the delay time τ.

Once $\widehat{C}$ is obtained, to guarantee $x(t+\tau) \in X$ for all $t \geq 0$, we need the initial state $x(0)$ and $u_{1:\tau}(0)$ of dynamics Σ_{aug} to be in the set

$$C_\tau = \{(x(0), u_{1:\tau}(0)) \mid \widehat{x}_\tau(0) \in \widehat{C}\}, \tag{6.10}$$

where $\widehat{x}_\tau(0)$ is a function of $x(0)$ and $u_{1:\tau}(0)$ defined in (6.5). Furthermore, we want $x(0:\tau-1)$ to stay within X. Note that $x(0:\tau-1)$ is determined by $x(0)$ and $u_{1:\tau-1}(0)$, that is

$$x(k) = A^k x(0) + \sum_{i=1}^{k} A^{i-1} B u_{k-i+1}(0) + \sum_{i=1}^{k} A^{i-1} F d(k-i). \tag{6.11}$$

Therefore, for $k = 0, 1, ..., \tau - 1$, the condition under which $x(k)$ is in X for arbitrary $d(0:k-1)$ is

$$C_k = \left\{ (x(0), u_1(0), \ldots, u_\tau(0)) \,\middle|\, \left(A^k x(0) + \sum_{i=1}^{k} A^{i-1} B u_{k-i+1}(0)\right) \in X - \sum_{i=1}^{k} A^{i-1} F D \right\}. \tag{6.12}$$

Now we denote the set of states in S satisfying the constraints in (6.10) and (6.12) by

$$C_{ext} = \left(\bigcap_{i=0}^{\tau} C_i \right) \cap S. \tag{6.13}$$

Now, we are ready to state our main result.

Theorem 6.2. *The set C_{ext} in* (6.13) *is the maximal RCIS of Σ_{aug} subject to the safe set $S \times U$.*

Proof. Denote the maximal RCIS of Σ_{aug} contained by S as C_{aug}. First, we want to show $C_{ext} \subseteq C_{aug}$. It is enough to show that C_{ext} is an RCIS subject to $S \times U$: Let $(x(t), u_{1:\tau}(t)) \in C_{ext}$. We want to find a $u(t) \in U$ such that for all $d(t) \in D$, $(x(t+1), u_{2:\tau}(t), u(t)) \in C_{ext}$.

Since $(x(t), u_{1:\tau}(t)) \in C_\tau \cap S$, we have $\widehat{x}_\tau(t) \in \widehat{C}$. Hence, there exists $u(t) \in U$ such that for all

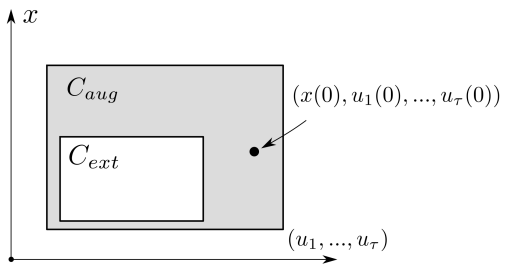


Figure 6.1: Pick a point from $C_{aug} \backslash C_{ext}$.

$d(t) \in D$, $\widehat{x}_\tau(t+1) \in \widehat{C}$. That is,

$$(x(t+1), u_{2:\tau}(t), u(t)) \in C_\tau. \tag{6.14}$$

Also, since $(x(t), u_{1:\tau}(t)) \in \bigcap_{i=0}^{\tau-1} C_i$, the set $\{x(t+1), ..., x(t+\tau-1)\}$ is contained by X. Because $x(t+1) \in X$ and $u(t) \in U$,

$$(x(t+1), u_{2:\tau}(t), u(t)) \in S. \tag{6.15}$$

Since $\widehat{x}_\tau(t) \in \widehat{C} \subseteq X - D_\tau$, $x(t+\tau) \in X$ by (6.7). Hence, for state $(x(t+1), u_{2:\tau}(t), u(t))$ and arbitrary $d(t+1:t+\tau-1) \in D^{\tau-1}$, it is verified that $\{x(t+1), ..., x(t+\tau)\}$ is contained by X, which implies

$$(x(t+1), u_{2:\tau}(t), u(t)) \in \bigcap_{i=0}^{\tau-1} C_i. \tag{6.16}$$

By (6.14), (6.15) and (6.16), there exists $u(t)$ such that for arbitrary $d(t) \in D$,

$$(x(t+1), u_{2:\tau}(t), u(t)) \in \left(\bigcap_{i=0}^{\tau} C_i \right) \cap S = C_{ext}.$$

Therefore, C_{ext} is an RCIS of dynamics Σ_{aug} subject to $S \times U$. Since C_{aug} is the maximal RCIS, $C_{ext} \subseteq C_{aug}$. Next, we want to show $C_{ext} = C_{aug}$. Suppose that there exists

$$(x(0), u_{1:\tau}(0)) \in C_{aug} \backslash C_{ext},$$

as demonstrated in Figure 6.1. It is easy to show that $C_{aug} \subseteq C_k \cap S$ for k from 0 to $\tau - 1$. Since $C_{ext} = \left(\bigcap_{i=0}^{\tau-1} (C_i \cap S) \right) \cap C_\tau \subseteq C_{aug}$, we have $(x(0), u_{1:\tau}(0)) \notin C_\tau$ and thus $\widehat{x}_\tau(0) \notin \widehat{C}$. Since $\widehat{C}$ is the maximal RCIS of Σ_{aux} in $X - D_\tau$, by Proposition 6.1, there exists $N \geq 0$ such that $\forall u(0) \in U$, $\exists d(0) \in D$, $\forall u(1) \in U$, $\exists d(1) \in D$, ..., $\forall u(N-1) \in U$, $\exists d(N-1) \in D$, the trajectory $\{\widehat{x}_\tau(k)\}_{k=0}^{N} \not\subseteq$

$X - D_\tau$. That is, there exists some $s \leq N$ such that $\widehat{x}_\tau(s) \notin X - D_\tau$. Denote $(x(s), u_{1:\tau}(s))$ as the state of Σ_{aug} corresponding to $\widehat{x}_\tau(s)$. Followed by $\widehat{x}_\tau(s) \notin X - D_\tau$, for any possible state $(x(s), u_{1:\tau}(s))$ of Σ_{aug} at time s, there exists $d(s : s+\tau-1) \in D^\tau$ such that $x(s+\tau) \notin X$, which holds for arbitrary control inputs $u(s : s+\tau-1)$. Contradiction to the assumption that C_{aug} is an RCIS of dynamics Σ_{aug}. Therefore, $C_{ext} = C_{aug}$. □

As an application of Theorem 6.2, we can compute the maximal RCIS C_{ext} of Σ_{aug} using the RHS of (6.13). In terms of computational complexity, given the maximal RCIS $\widehat{C}$ of the auxiliary system, the sets C_i for i from 0 to τ are polytopes explicitly defined in (6.10) and (6.12) and thus can be constructed in one shot. Thus, the main computational burden comes from computing $\widehat{C}$. Since $\widehat{C}$ is the maximal RCIS of a system in $\mathbb{R}^n$, compared with directly applying the outside-in algorithm to the augmented system Σ_{aug} in $\mathbb{R}^{n+\tau m}$, our approach can reduce the computation time significantly. This is verified later by numerical examples in Section 6.4.

Finally, the initial section of the proof for Theorem 6.2 yields a significant corollary, demonstrating the versatility of our approach in cases where $\widehat{C}$ may not necessarily be the maximal RCIS of Σ_{aux}. This can be highly valuable, as it is often easier to find an RCIS than the maximal RCIS.

Corollary 6.2.1. If $\widehat{C}$ is an RCIS of Σ_{aux} subject to $X - D_\tau \times U$ (but not necessarily the maximal one), then C_{ext} is an RCIS of Σ_{aug} subject to $S \times U$.

6.3 Extension to Systems with Preview

In the previous section, we make the assumption that the disturbance is entirely unknown at the runtime. However, in real-world systems, it is often possible to measure many external signals through sensors ahead of time. For instance, consider the lane-keeping example in Section 6.4.2, where the disturbance term corresponds to lane curvature and can be predicted.

In this section, we shift our focus to the investigation of the maximal RCIS for time-delayed systems with disturbance inputs that can be categorized into two types: non-measurable disturbances and disturbances with preview. Formally, we define a disturbance as having a *k-step preview* if the controller can measure and utilize the disturbance signal $d(t : t+k-1)$ at time t. A *non-measurable* disturbance is one with a 0-step preview.

Consider the following linear system[1]

$$\Sigma^{prev}_{delay} : x(t+1) = Ax(t) + Bu(t-\tau) + F_0 d_0(t) + F_p d_p(t), \tag{6.17}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, d_0 is a non-measurable disturbance in D_0 and d_p is a disturbance with

[1]The results in this section can be generalized for systems with multiple preview horizons (less than or equal to τ).

p-step preview in $D_p \subseteq \mathbb{R}^l$. Let the safe set of the system be $S_{xu} = X \times U$. We assume that the sets X, U, D_0, and D_p are polytopes. In the analysis that follows, we assume $p \leq \tau$ and show that in this case one can still compute the invariant set by applying the invariance iterations in n-dimensional space. While the case when $p > \tau$ can be handled with some added complexity within the current framework, whether this case can also be reduced to iterations in an n-dimensional space is left for future research.

Similar to the delay, a system with preview can also be converted to a standard linear system by appending the state space with addition states corresponding to the preview of the disturbance. In particular, we have the following augmented system equivalent to Σ^{prev}_{delay}:

$$\Sigma^{prev}_{aug}: \begin{cases} x(t+1) &= Ax(t) + Bu_1(t) + F_0 d_0(t) + F_p d_{p,1}(t) \\ u_1(t+1) &= u_2(t) \\ &\vdots \\ u_\tau(t+1) &= u(t) \\ d_{p,1}(t+1) &= d_{p,2}(t) \\ d_{p,2}(t+1) &= d_{p,3}(t) \\ &\vdots \\ d_{p,p}(t+1) &= d_{p,f}(t), \end{cases} \tag{6.18}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $d_0(t)$ and $d_{p,f}(t)$ are non-measurable disturbances bounded by D_0 and D_p. Note that $d_{p,f}(t)$ is just an alias of $d_p(t+p)$. The safe set for the system Σ^{prev}_{aug} is $S_p \times U$, with $S_p = X \times U^\tau \times D_p^p$.

Problem 6.2. *Find the maximal RCIS of system Σ^{prev}_{aug} subject to the safe set $S_p \times U$.*

Proceeding as in the previous section, we can write the τ-step expansion of $x(t+\tau)$ as

$$\begin{aligned} x(t+\tau) =& A^\tau x(t) + \sum_{j=1}^{\tau} A^{j-1} B u_{\tau-j+1}(t) + \sum_{j=\tau-p+1}^{\tau} A^{j-1} F_p d_{p,\tau-j+1}(t) + \\ & \sum_{j=1}^{\tau} A^{j-1} F_0 d_0(t+\tau-j) + \sum_{j=1}^{\tau-p} A^{j-1} F_p d_{p,f}(t+\tau-p-j). \end{aligned} \tag{6.19}$$

Based on what can be measured at time t, we define the prediction variable

$$\widehat{x}_\tau(t) = A^\tau x(t) + \sum_{i=1}^{\tau} A^{i-1} B u_{\tau-i+1}(t) + \sum_{j=\tau-p+1}^{\tau} A^{j-1} F_p d_{p,\tau-j+1}(t) \tag{6.20}$$

assuming the non-measurable disturbance is zero. The dynamics of $\widehat{x}_\tau$ takes the form

$$\Sigma_{aux}^{prev} : \widehat{x}_\tau(t+1) = A\widehat{x}_\tau(t) + Bu(t) + A^\tau F_0 d_0(t) + A^{\tau-p} F_p d_{p,f}(t), \tag{6.21}$$

where $u(t) \in U$, $d_0 \in D_0$ and $d_{p,f}(t) \in D_p$ are non-measurable disturbances, which is again an n-dimensional system. From Eq. (6.20), it follows that to ensure $x(t+\tau) \in X$, we need

$$\widehat{x}_\tau(t) \in \widehat{X} := X - \sum_{i\in\{0,p\}} \sum_{j=1}^{\tau-i} A^{j-1} F_i D_i. \tag{6.22}$$

Let $\widehat{C}_p$ as the maximal RCIS of Σ_{aux}^{prev} within $\widehat{X}$. We next show how to construct an invariant set for the $(n+m\tau+pl)$-dimensional augmented system Σ_{aug}^{prev} using the set $\widehat{C}_p \subseteq \mathbb{R}^n$. The constraints on the initial states of the system Σ_{aug}^{prev}, ensuring the existence of a controller such that $x(t+\tau) \in X$ for all $t \geq 0$, can be written as

$$C_{p,\tau} = \{(x(0), u_1(0), ..., d_{\tau,\tau}(t)) \mid \widehat{x}_\tau(0) \in \widehat{C}_p\}, \tag{6.23}$$

where $\widehat{x}_\tau(0)$ is defined by (6.20).

To ensure $x(k) \in X$ for the time period $k = 0, \ldots, \tau-1$, we need the following constraints on initial states:

$$C_{p,k} = \Bigg\{ (x(0), u_1(0), \ldots, d_{\tau,\tau}(t)) \mid A^k x(0) + \sum_{j=1}^{k} A^{j-1} B u_{k-j+1}(0) + \sum_{j=\max\{1,k-p+1\}}^{k} A^{j-1} F_p d_{p,k-j+1}(0) \in X - \sum_{i\in\{0,p\}} \sum_{j=1}^{k-i} A^{j-1} F_i D_i \Bigg\} \tag{6.24}$$

Finally, define the intersection of these constraint sets:

$$C_{p,ext} = (\bigcap_{k=0}^{\tau} C_{p,k}) \cap S_p. \tag{6.25}$$

Theorem 6.3. *$C_{p,ext}$ is the maximal RCIS of system Σ_{aug}^{prev} in set S_p.*

Proof. The proof for Theorem 6.3 can be easily extended from the proof of Theorem 6.2, omitted for brevity. □

Similar to the preceding section, we have the following corollary.

Corollary 6.3.1. If the set $\widehat{C}_p$ is an RCIS of Σ_{aux}^{prev} subject to $(X - \sum_{i\in\{0,p\}} \sum_{j=1}^{\tau-i} A^{j-1} F_i D_i) \times U$, then the set $C_{p,ext}$ defined by (6.25) is an RCIS of Σ_{aug}^{prev} in S_p.

6.4 Numerical Examples

The algorithms are implemented in MATLAB 2018b on a computer equipped with Intel i7-8650U CPU and 16 GB memory. We use implementations from MPT3 toolbox [48] for the polytope operations in the algorithms.

6.4.1 One-dimensional Example

In this section, we use a toy example to show how much performance improvement is achieved by applying the proposed method.

Consider the following 1-dimensional system:

$$x(t+1) = 1.5x(t) + u(t-\tau) + d(t) \tag{6.26}$$

where $x(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}$ and $d(t) \in [-2,2]$ with p-step preview. The safe set on x and u is $[-32,32] \times [-20,20]$.

In Table 6.1, we compare the computation time of the maximal invariant sets for different τ and p using two methods: the proposed method (that uses outside-in algorithm on computing the maximal RCIS of the low-dimensional auxiliary system), and outside-in algorithm directly operating on the augmented system. We call the later the direct method for short. The outside-in algorithm terminates in finite number of steps for all of the test cases in the table. There are two important observations.

First, for each τ in Table 6.1, p is selected as the smallest preview length that makes the maximal invariant set nonempty. The increasing trend on p in Table 6.1 implies that if we do not have any preview on disturbance, the RCIS becomes empty very soon as τ increases. That reveals how preview on disturbance reduces conservativeness for input-delay systems, which is why we take preview into consideration in Section 6.3.

Second, according to the last two columns of Table 6.1, the computation time with the direct method increases drastically as τ increases, while the computation time for the proposed method just increases slightly. This is because the dimension of the reduced-order system does not change as τ and p increase. Our method is apparently more efficient than the direct method in this example.

6.4.2 Vehicle Lane Keeping Control

In this example, the proposed method is applied to synthesize a controller that guarantees the safety of a vehicle in a lane-keeping scenario. The goal of lane keeping is to control the vehicle to follow

Table 6.1: Time required to compute an invariant set with the proposed method and the direct method.

τ	p	proposed method (s)	direct method (s)
1	0	0.7705	0.5960
5	1	0.8779	7.7573
10	6	1.1548	98.3379
15	11	1.6999	525.7656
20	16	3.0460	1.6217×10^3

the center line of the road. The safety requirement is to make sure the lateral displacement, the lateral velocity, yaw angle and yaw rate of the vehicle with respect to the road center are within given bounds so that the vehicle does not leave the target road, spin, or rollover.

The vehicle dynamics considered is linearized from a bicycle model [112] and discretized by forward Euler method with time step $h = 0.1s$. The longitudinal velocity v_d is fixed and equal to $30m/s$. The state of the system consists of the lateral displacement y between the vehicle center and the road center, the lateral velocity v, the yaw angle $\Delta\Psi$ and the yaw rate r of the vehicle, denoted by $x = [y, v, \Delta\Psi, r]$. The dynamics Σ_{car} of x is

$$x(t+1) = (I + A \cdot h)x(t) + Bh\delta_f(t-\tau) + Fhr_d(t) \tag{6.27}$$

with I equal to the identity matrix and

$$A = \begin{bmatrix} 0 & 1 & u & 0 \\ 0 & -\frac{C_{\alpha f}+C_{\alpha r}}{mu} & 0 & \frac{bC_{\alpha r}-aC_{\alpha f}}{mu} - u \\ 0 & 0 & 0 & 1 \\ 0 & \frac{bC_{\alpha r}-aC_{\alpha f}}{I_z u} & 0 & -\frac{a^2C_{\alpha f}+b^2C_{\alpha r}}{I_z u} \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{C_{\alpha f}}{m} \\ 0 \\ a\frac{C_{\alpha f}}{I_z} \end{bmatrix}, F = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix},$$

where the steering angle $\delta_f \in \mathbb{R}$ is the control input with τ-step delay and the desired yaw rate r_d is a disturbance with p-step preview ($p \leq \tau$). The parameters in A, B matrices are taken from [112]. According to [1], the maximal range of r_d with respect to $v_d = 30m/s$ in Michigan is $D = [-0.05, 0.05]$. Desired yaw rate r_d is a function of the road curvature, which can be measured with a forward looking camera or acquired from a map ahead of time. Therefore it is reasonable to assume that r_d is a disturbance with preview.

The safe region X of states is given by bounds $|y| \leq 0.9$, $|v| \leq 1.2$, $|\Delta\Psi| \leq 0.05$, $|r| \leq 0.3$. The input constraint set U is given by input bound $|\delta_f| \leq \pi/2$. For $\tau = 10, p = 8$, our method takes $249s$ to compute the maximal RCIS of the 22-dimensional augmented system subject to the safe set $X \times U^{10} \times D^8 \times U$. By fixing r, u_1, …, u_{10}, d_1, …, d_8 to be zero, we make a 3-dimensional slice of

the 22-dimensional polytope, shown in Figure 6.2. The red region in Figure 6.2 contains all the feasible initial values of the first three coordinates $(y, v, \Delta\Psi)$ from which it is possible to guarantee safety, when the other coordinates have initial value equal to 0.

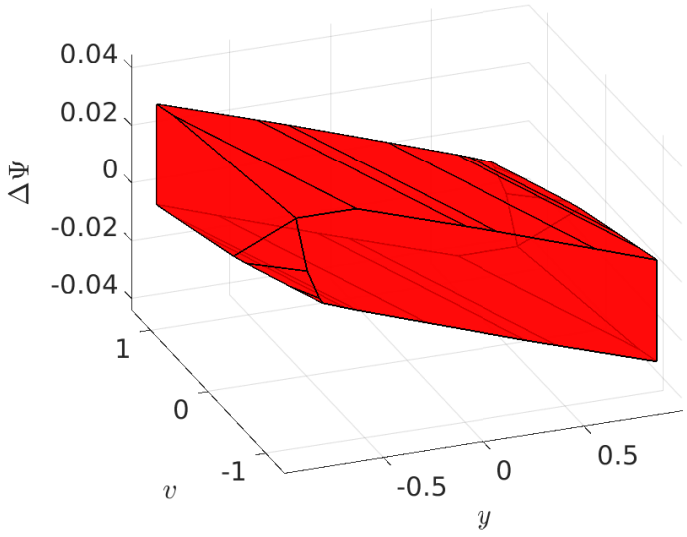


Figure 6.2: A slice of the maximal RCIS.

Once the maximal RCIS C is obtained, the admissible input set with respect to a state in C is the set of inputs that make the next state within C robust to any disturbances in D. A safety supervisor for a legacy vehicle controller or human-driver can be implemented by checking if the controller's output is within the admissible input set at each time and making appropriate adjustment [91].

We run a simulation under the supervisory control framework using the maximal RCIS of Σ_{car}. In the simulation, r_d is given by a sine function over time. A legacy controller u_n of the vehicle is obtained by solving a LQR problem for the augmented system of Σ_{car}. The supervisor is implemented by projecting the output of u_n to the admissible input set given by the maximal acRCIS of Σ_{car} at each time step. As a baseline, we first assume that the invariant set designer is either unaware of the existence of the delay and preview or simply ignores them and implements the supervisor using the maximal controlled invariant of Σ_{car} with zero delay and no preview. Then, another supervisor is implemented based on the maximal RCIS of Σ_{car} with the actual delay steps and preview steps. A sample trajectory of the closed-loop systems equipped with the first and second supervisors are compared in Figure 6.3, indicated by red and blue curves. The red trajectory terminates at 4.9s because at that time the system equipped with the first supervisor reaches the unsafe region. In contrast, the system equipped with the second supervisor stays within the safety bounds all the time. Comparing the two different simulation results, it can be seen that simply ignoring the delay can lead to unsafe situations. It is also worth noting that the invariant set becomes

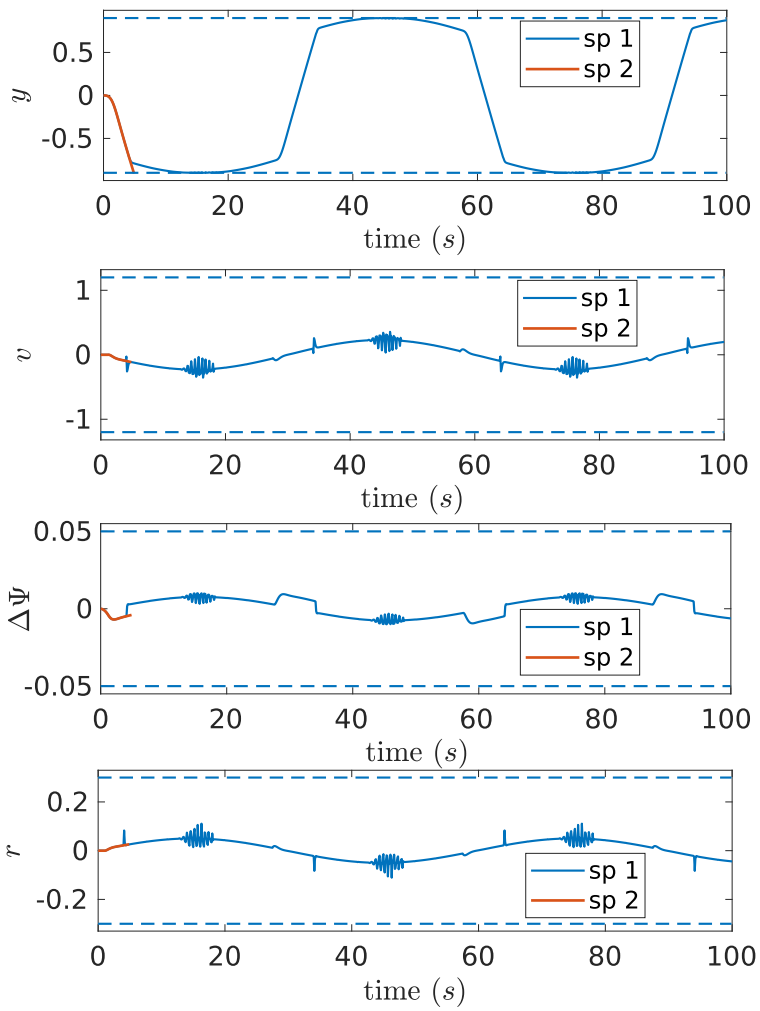


Figure 6.3: Trajectories of the supervisory control simulation. The safety bound on each coordinates are indicated by the dash lines. The red and blue trajectories correspond to supervisors designed with different knowledge on the delay time.

empty in this example when taking the preview time p to be zero while keeping the delay time as is. In fact, for any value of $p < 8$, the invariant set is empty. This indicates the value of preview in coping with uncertainty for systems with input delays.

6.5 Conclusion

In this chapter, we propose a scalable method for computing RCISs for linear systems subject to input delays. This method is extended to incorporate preview information while preserving the scalability properties. Both of the problems studied are motivated by safety control problems in automotive domain, yet we believe the proposed methods are broadly applicable. Our current work

focuses on understanding the robustness of the approach to uncertainties in the delay time. We are also interested in time-varying delays where the correctness and maximality guarantees will depend on the protocol that resolves missing or clashing input packets.

Part II

Safety Control Beyond the Maximal RCIS

CHAPTER 7

Quantifying the Value of Preview Information for Safety Control

In this chapter, we are interested in safety control for systems with preview information. The goal of safety control is to synthesize safe controllers that can guarantee the closed-loop system satisfies given safety requirements indefinitely, robust to certain amount of uncertainties in the dynamics [20, 77, 78, 83]. For systems with preview, the safe controllers are allowed to use feedback on the states and feedforward on the preview information. An important question is if the safe controller should utilize all the available preview information. In theory, the more preview information a controller utilizes, the more safety the system may gain. This is verified by numerical examples in our previous works [77, 78], where incorporating more preview information enlarges (i) the region of states where the safety requirements can be enforced or (ii) the range of disturbances the system can tolerate under the safety constraints. However, in practice, it is often computationally intractable to incorporate all the available preview information since many safety control algorithms (even the most scalable ones) suffer from the curse of dimensionality [119, 47, 25, 45]. In those cases, one needs to carefully select the amount of preview information fed into the safe controllers, for a good balance between the computational cost and the safety loss due to omitting part of the preview information.

Motivated by this need, in this chapter, we want to reveal how the safety of a system is impacted by different amount of preview. Since we focus on the preview of disturbances, in our analysis, the amount of preview information is indicated by the number of the time steps that the future disturbances can be previewed, which we call the *preview time*. We measure the safety of the same system with different preview time by a notion called *safety regret*, defined based on the maximal RCIS of an auxiliary system that augments the original states with the preview information. Given a preview time p, the corresponding safety regret reflects the room to improve the system safety as we increase the preview time from p to infinity. The main contributions of this chapter include:

- We provide novel outer approximations of the maximal RCIS for nonlinear systems augmented with preview information, by exploring a duality between preview and input delay;

- For linear systems, we prove that the safety regret of a finite-step preview decays exponentially fast with the preview time. For polytopic state-input constraints, we further develop algorithms that computes upper bounds of the safety regret;
- We extend our analysis to show the projection of the feasible domain of a preview-based robust MPC onto the space of initial states converges exponentially with the prediction horizon;
- We demonstrate the usage of our theoretical results and the proposed algorithms with both analytical and numerical examples.

Chapter Overview. We state the problem in Section 7.1, and present several structural properties of the maximal RCIS for systems with preview in Section 7.2. The main results of this chapter on the convergence of safety regret are presented in Section 7.3. Using similar techniques from the previous section, Section 7.4 analyzes the convergence of the feasible domain of a preview-based MPC. In Section 7.5, we demonstrate our theoretical and algorithmic results with examples from multiple fields. All the proofs in this chapter can be found in Appendix D.

Notations. For a polytope $\mathscr{P} = \{x \mid Hx \leq h\}$, the H-representation of $\mathscr{P}$ is the matrix $[H\ h]$. For a positive integer q, the set $\{1,2,\cdots,q\}$ is denoted by $[q]$. The unit hypercube in $\mathbb{R}^n$ is denoted by $\mathbf{B}(n) = [-1,1]^n$. The Hausdorff distance between two sets X and Y in $\mathbb{R}^n$ is induced from 2-norm in $\mathbb{R}^n$ and is denoted by $d(X,Y)$. Given a compact set X in $\mathbb{R}^n$, the radius of X with respect to a point x_0 is defined by $\psi(X) = \sup_{x\in X} \|x - x_0\|_2$.

7.1 Problem Setup

A general discrete-time dynamical system Σ is in form of

$$\Sigma : x(t+1) = f(x(t), u(t), d(t)), \tag{7.1}$$

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$ and disturbance $d \in D \subseteq \mathbb{R}^l$. The disturbance set D is assumed to be compact. A system Σ in form of (7.1) is said to have *p-step preview* if at each time step t, the controller has access to the measurements of

- the current state $x(t)$, and
- the current and incoming disturbances $\{d(t+k)\}_{k=0}^{p-1}$ in p steps, denoted by $d_{1:p}(t)$ for short.

For a system Σ with p-step preview, we define an augmented system whose states stack the states $x(t)$ and the previewed disturbances $d_{1:p}(t)$ of Σ together, called the p-augmented system Σ_p of Σ.

The p-augmented system Σ_p is in form of

$$\Sigma_p : \begin{bmatrix} x(t+1) \\ d_1(t+1) \\ \vdots \\ d_{p-1}(t+1) \\ d_p(t+1) \end{bmatrix} = \begin{bmatrix} f(x(t),u(t),d_1(t)) \\ d_2(t) \\ \vdots \\ d_p(t) \\ d(t) \end{bmatrix} \tag{7.2}$$

with state $(x,d_1,\cdots,d_p) \in \mathbb{R}^{n+pl}$, input $u \in \mathbb{R}^m$ and disturbance $d \in D \subseteq \mathbb{R}^l$. Any preview-based controller of Σ with feedback on the state $x(t)$ and feedforward on the previewed disturbances $d_{1:p}(t)$ is equivalent to a state-feedback controller of the p-augmented system Σ_p. Due to this equivalence relation, we study the safety control for the system Σ with p-step preview by simply studying the state-feedback safety control for its p-augmented system Σ_p.

In this chapter, we want to study the impact of preview to the safety of the system Σ by revealing how the maximal RCIS varies with different preview time. More specifically, given the safe set S_{xu} of Σ, we define safe set $S_{xu,p}$ of the p-augmented system Σ_p of Σ by

$$S_{xu,p} = \{(x,d_{1:p},u) \mid (x,u) \in S_{xu}, d_{1:p} \in D^p\}. \tag{7.3}$$

Intuitively, the safe set $S_{xu,p}$ imposes the same constraints on (x,u) as in S_{xu} and does not constrain the previewed disturbances $d_{1:p}$ since we cannot control the preview information.

We denote the maximal RCIS of the system Σ_p in the safe set $S_{xu,p}$ by $C_{max,p}$. The shape and dimensions of $C_{max,p}$ vary with the preview time p. Since $C_{max,p}$ characterizes all the safe controllers of Σ_p, the variation of $C_{max,p}$ with the preview time p reflects the impact of different preview time to the safety of Σ. In this chapter, we want to reveal how $C_{max,p}$ varies with different preview time p, and its implication to the safety of the system. Note that it is intractable to compute $C_{max,p}$ for all p and then compare their difference. Later we introduce a quantity called *safety regret* to evaluate the variation of $C_{max,p}$ with the preview time p. We show that it is possible to estimate the safety regret without computing $C_{max,p}$, by exploiting the structures in the augmented system Σ_p and its safe set $S_{xu,p}$.

Before moving to the next section, we introduce one technical definition used later.

Definition 7.1. A set X is an *N-step λ-contractive set* of the system Σ in the safe set S_{xu} if X satisfies that

$$X \subseteq Pre_{\Sigma}^{N}(\lambda X, S_{xu}). \tag{7.4}$$

Intuitively, X is N-step λ-contractive if we can steer the system from any state in X to a state

in λX in N steps while keeping the state-input trajectory in the safe set. Note that an N-step λ-contractive set is not necessarily an RCIS. When $N = 1$, we call X a λ-contractive set for short.

7.2 Structural Properties of the Maximal RCIS for Systems with Preview

In this section, we present several structural properties of the maximal RCIS $C_{max,p}$ of the p-augmented system, which allow us to approximate $C_{max,p}$ without computing it, and pave the way to our analysis in the following sections.

7.2.1 Inner and outer bounds of the maximal RCIS $C_{max,p}$

We briefly summarize the inner and outer bounds of the maximal RCIS $C_{max,p}$ derived in the previous work [78]. Those bounds are useful when the actual representation of $C_{max,p}$ is difficult to obtain, and play important roles later in our analysis to the variation of $C_{max,p}$ with different p.

To have an outer bound of the maximal RCIS $C_{max,p}$, we introduce an auxiliary system: Given a system Σ and a safe set S_{xu}, we define the *disturbance collaborative system* $\mathscr{D}(\Sigma)$ of Σ by

$$\mathscr{D}(\Sigma) : x(t+1) = f(x(t), u(t), u_d(t)), \tag{7.5}$$

with state $x \in \mathbb{R}^n$ and two inputs $u \in \mathbb{R}^m$, $u_d \in \mathbb{R}^l$. The safe set $S_{xu,co}$ of $\mathscr{D}(\Sigma)$ is $S_{xu} \times D$, that is

$$S_{xu,co} = \{(x, u, u_d) | (x, u) \in S_{xu}, u_d \in D\}. \tag{7.6}$$

We denote the maximal CIS[1] of $\mathscr{D}(\Sigma)$ in $S_{xu,co}$ by $C_{max,co}$. The only difference between the original system Σ and the corresponding $\mathscr{D}(\Sigma)$ is that we turn the disturbance term d in Σ into the second input u_d in $\mathscr{D}(\Sigma)$. Due to the extra control power introduced by u_d, the maximal CIS $C_{max,co}$ of $\mathscr{D}(\Sigma)$ provides an outer bound of the maximal RCIS $C_{max,p}$ for all p. This outer bound along with an inner bound of $C_{max,p}$ are stated in the following theorem.

Theorem 7.1 ([78], Theorems 1 and 2)**.** *For a system Σ with p-step preview, the maximal RCIS $C_{max,p}$ of Σ_p in S is inner approximated by $C_{max,p'} \times D^{p-p'}$ for any $p' < p$, and is outer approximated by the Cartesian product $C_{max,co} \times D^p$. That is,*

$$C_{max,p'} \times D^{p-p'} \subseteq C_{max,p} \subseteq C_{max,co} \times D^p. \tag{7.7}$$

[1]The set $C_{max,co}$ is a CIS instead of an RCIS since $\mathscr{D}(\Sigma)$ has no disturbance.

Furthermore, the inner bound $C_{max,p'} \times D^{p-p'}$ is an RCIS of Σ_p in $S_{xu,p}$.

The proof of Theorem 7.1 can be found in [78]. We have discussed the intuition behind the outer bound in Theorem 7.1. The inner bound is based on the intuition that a longer preview time p should make the maximal RCIS $C_{max,p}$ larger. However, this intuition is not fully correct. Since the dimensionality of Σ_p grows with p, the maximal RCIS $C_{max,p}$ lies in a different space for each different p and cannot be compared with each other directly. It turns out that for a shorter preview time $p' < p$, we can always lift an RCIS $C_{p'}$ of $\Sigma_{p'}$ to the RCIS $C_{p'} \times D^{p-p'}$ of Σ_p and then compare the lifted set with $C_{max,p}$. The difference between between the lifted RCIS $C_{max,p'} \times D^{p-p'}$ and the maximal RCIS $C_{max,p}$ tells how much we gain in safety by increasing the preview time from p' to p.

7.2.2 Improved outer bounds for the maximal RCIS $C_{max,p}$

The outer bound of $C_{max,p}$ in Theorem 7.1 is based on the maximal CIS $C_{max,co}$ of $\mathscr{D}(\Sigma)$ of Σ. Since $C_{max,co}$ is independent of the preview time p, this outer bound is not very tight. In this section, we derive tighter outer bounds of $C_{max,p}$, by exploring a duality between delay and preview.

In Section 7.2.1, we introduce the disturbance collaborative system $\mathscr{D}(\Sigma)$. However, except for $\mathscr{D}(\Sigma)$, we can also define the disturbance collaborative system $\mathscr{D}(\Sigma_p)$ of the p-augmented system Σ_p. Formally, the system $\mathscr{D}(\Sigma_p)$ is in form of

$$\mathscr{D}(\Sigma_p): \begin{bmatrix} x(t+1) \\ d_1(t+1) \\ \vdots \\ d_{p-1}(t+1) \\ d_p(t+1) \end{bmatrix} = \begin{bmatrix} f(x(t),u(t),d_1(t)) \\ d_2(t) \\ \vdots \\ d_p(t) \\ u_d(t) \end{bmatrix} \tag{7.8}$$

with control inputs $u \in \mathbb{R}^n$ and $u_d \in \mathbb{R}^l$. Let the safe set of the system $\mathscr{D}(\Sigma_p)$ be $S_{xu,p} \times D$. We denote the maximal CIS of the system $\mathscr{D}(\Sigma_p)$ in safe set $S_{xu} \times D$ by $C_{max,p,co}$. When $p = 0$, $C_{max,0,co} = C_{max,co}$.

Then, for any $p' \leq p$, the system Σ_p can be viewed as the $(p-p')$-augmented system of the system $\Sigma_{p'}$. By applying Theorem 7.1 to $\Sigma_{p'}$, we have the following corollary.

Corollary 7.1.1. For a system Σ with p-step preview and any non-negative integer $p' \leq p$, the maximal RCIS $C_{max,p}$ is outer approximated by the Cartesian product $C_{max,p',co} \times D^{p-p'}$. That is,

$$C_{max,p} \subseteq C_{max,p',co} \times D^{p-p'}. \tag{7.9}$$

According to Corollary 7.1.1, we have $(p+1)$ outer bounds for the maximal RCIS $C_{max,p}$, that is $\{C_{max,k,co} \times D^{p-k}\}_{k=0}^{p}$, including the one in Theorem 7.1.

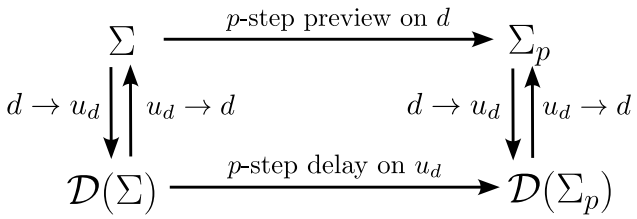


Figure 7.1: The relation diagram of the four systems Σ, $\mathscr{D}(\Sigma)$, Σ_p, and $\mathscr{D}(\Sigma_p)$.

Next, we want to study the relation between those outer bounds, and figure out which one is the tightest bound. The key is to realize that $\mathscr{D}(\Sigma_p)$ is actually the state-space representation of $\mathscr{D}(\Sigma)$ with p-step input delay in u_d:

$$x(t+1) = f(x(t), u(t), u_d(t-p)). \tag{7.10}$$

In [83], $\mathscr{D}(\Sigma_p)$ is called the augmented system of the input delay system in (7.10). Intuitively, since we turn the disturbance in Σ_p to the input u_d in $\mathscr{D}(\Sigma_p)$, the p-step preview on d becomes a p-step delay on the input u_d. This is what we call the duality between delay and preview. The relations of the four systems Σ, Σ_p, $\mathscr{D}(\Sigma)$ and $\mathscr{D}(\Sigma_p)$ are shown by the diagram in Fig. 7.1. Since $\mathscr{D}(\Sigma_p)$ is the "delayed" version of $\mathscr{D}(\Sigma)$, the maximal CIS of $\mathscr{D}(\Sigma_p)$ is embedded in the maximal CIS of $\mathscr{D}(\Sigma)$, shown by the next theorem.

Theorem 7.2. *For any $p \geq 0$, the maximal CIS $C_{max,p,co}$ of the system $\mathscr{D}(\Sigma_p)$ in the safe set $S_{xu,p} \times D$ can be obtained from the maximal CIS $C_{max,co}$ of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$, via the formula*

$$\begin{aligned} C_{max,p,co} = \{&(x(0), d_1(0), \cdots, d_p(0)) \mid \\ &\exists u(0), \cdots, u(p-1), \\ &(x(t), u(t)) \in S_{xu}, 0 \leq t \leq p-1, \\ &x(p) \in C_{max,co}, \\ &d_i(0) \in D, 1 \leq i \leq p\}, \end{aligned} \tag{7.11}$$

where $x(t) = f(x(t-1), u(t-1), d_t(0))$ for $0 \leq t \leq p$. Furthermore, we have

$$C_{max,p,co} \subseteq C_{max,co} \times D^p. \tag{7.12}$$

Theorem 7.2 extends the results in [83] and actually holds for any deterministic system with input delays. According to (7.12), it can be shown that for any $p' \leq p$,

$$C_{max,p,co} \subseteq C_{max,p',co} \times D^{p-p'} \subseteq C_{max,co} \times D^p. \tag{7.13}$$

Thus, $C_{max,p,co}$ is the tightest outer bound of $C_{max,p}$ in the set $\{C_{max,k,co} \times D^{p-k}\}_{k=0}^{p}$. Furthermore, this outer bound can be computed by (7.11), once the maximal RCIS $C_{max,co}$ is known. Compared with the computation cost of $C_{max,co}$, the computation cost of (7.11) is typically negligible. Thus, we improve the outer bound of $C_{max,p}$ with little extra cost.

Due to the similarity between the system pairs (Σ, Σ_p) and $(\mathscr{D}(\Sigma), \mathscr{D}(\Sigma_p))$, one may wonder if the maximal RCIS $C_{max,p}$ of the p-augmented system can also be obtained from C_{max} by a formula similar to (7.11). Unfortunately, we cannot find such a formula. But the duality between the preview and delay, as shown in Fig. 7.1, does allow us to take the advantage of (7.11) to obtain the tighter outer approximations of $C_{max,p}$ easily.

7.3 Quantifying the Value of Preview

In this section, we want to study how the value of preview information in safety control varies with the preview time p. First, we need to find a way to quantify the value of different preview time. Ideally, we want to quantify the value of preview by the size of the maximal RCIS $C_{max,p}$, since this set characterizes all the safe controllers. However, since the dimensionality of $C_{max,p}$ depends linearly on p, it is unfair to compare the size of $C_{max,p}$ over different preview time p. To resolve this issue, we project $C_{max,p}$ onto the first n coordinates, and study how the size of the projections $\pi_{[1,n]}(C_{max,p})$ varies with the preview time p.

We argue that the size of the projection $\pi_{[1,n]}(C_{max,p})$ indeed reflects the value of p-step preview. First, the projection $\pi_{[1,n]}(C_{max,p})$ contains all the initial states x_0 of the system Σ where a safe controller exists for some initial p-step preview. Second, by Theorem 7.1, for any $p' \leq p$,

$$\pi_{[1,n]}(C_{max,p'}) \subseteq \pi_{[1,n]}(C_{max,p}). \tag{7.14}$$

That is, the projection $\pi_{[1,n]}(C_{max,p})$ expands with p, which matches our intuition that a longer preview time has higher value in safety control.

In the remainder of this section, we show what the limit of the projection $\pi_{[1,n]}(C_{max,p})$ is and how fast this projection converges to the limit in Hausdorff distance. For short, the Hausdorff distance between the projection $\pi_{[1,n]}(C_{max,p})$ and its limit is denoted by d_p, that is

$$d_p = d(\pi_{[1,n]}(C_{max,p}), \lim_{p\to\infty} \pi_{[1,n]}(C_{max,p})). \tag{7.15}$$

Intuitively, the value of d_p reflects the room to improve the safety of the system if we are allowed to further increase the preview time. In a sense, d_p measures the safety gap between the p-step preview and the infinite-step preview. Due to this reason, we also call d_p the *safety regret* of the

p-step preview (similar to the notion of regret in [122, 70]).

7.3.1 Assumptions

We restrict our analysis to the class of discrete-time linear systems. A system Σ is *linear* if its transition function f in (7.1) is in form of

$$f(x,u,d) = Ax + Bu + Ed, \tag{7.16}$$

with matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $E \in \mathbb{R}^{n\times l}$. The results later in this section are based on the following assumptions.

Assumption 7.1. *The disturbance collaborative system $\mathscr{D}(\Sigma)$ of the linear system Σ is stabilizable.*

Note that the disturbance-collaborative system $\mathscr{D}(\Sigma)$ being stabilizable is a weaker condition than the system Σ being stabilizable, since the system $\mathscr{D}(\Sigma)$ has one more control input u_d than the system Σ.

Assumption 7.2. *The safe set S_{xu} and the disturbance set D are convex and compact.*

Lemma 7.3. Suppose that the system Σ is linear. Under Assumption 7.2, if the set $\pi_{[1,n]}(C_{max,p})$ is nonempty, then

(i) $\pi_{[1,n]}(C_{max,p})$ is a convex compact CIS of the system $\mathscr{D}(\Sigma)$ within $S_{xu} \times D$;

(ii) there exists a forced equilibrium $(x_e, u_e, d_e) \in S_{xu} \times D$ of the system $\mathscr{D}(\Sigma)$ such that x_e is in $\pi_{[1,n]}(C_{max,p})$.

Assumption 7.3. *For some $p_0 \geq 0$, there exists a forced equilibrium (x_e, u_e, d_e) of $\mathscr{D}(\Sigma)$ in the interior of $S_{xu} \times D$ with $x_e \in \pi_{[1,n]}(C_{max,p_0})$.*

According to Lemma 7.3, Assumption 7.3 is almost an implication of Assumption 7.2, except that we require the forced equilibrium is not only in the safe set $S_{xu} \times D$, but in its interior.

Remark 7.1. For linear systems, we can shift the origin of the state space to any forced equilibrium without changing the system equations. Hence without loss of generality, for the remainder of this section, we simply assume that the forced equilibrium (x_e, u_e, d_e) in Assumption 7.3 is the origin of the state-input-disturbance space $\mathbb{R}^{n+m+l}$.

7.3.2 Convergence of $\pi_{[1,n]}(C_{max,p})$

By Remark 7.1, the safe set $S_{xu} \times D$, the maximal RCIS $C_{max,co}$ and the projection $\pi_{[1,n]}(C_{max,p})$ all contain the origin for any $p \geq p_0$. Thus, there exists a scalar $\lambda \in (0,1]$ such that

$$0 \in \lambda C_{max,co} \subseteq \pi_{[1,n]}(C_{max,p_0}) \subseteq C_{max,co}. \tag{7.17}$$

We call the maximal λ such that (7.17) holds the *initial factor* λ_0, which reflects the percentage of the set $C_{max,co}$ contained in the projection of the set C_{max,p_0}. By definition, the initial factor $\lambda_0 \geq 0$. To prove the convergence of $\pi_{[1,n]}(C_{max,p})$, we need the initial factor $\lambda_0 > 0$, which is ensured by the following assumption.

Assumption 7.4. *The forced equilibrium (x_e, u_e, d_e) in Assumption 7.3 satisfies that x_e is in the interior of $\pi_{[1,n]}(C_{max,p_0})$.*

The reason to have the initial factor $\lambda_0 > 0$ becomes clear later in this section.

Lemma 7.4. For any system Σ in (7.1) with safe set S_{xu} and a preview time p, the projection $\pi_{[1,n]}(C_{max,p})$ satisfies that for any $k > 0$,

$$\begin{aligned} &Pre^k_{\mathscr{D}(\Sigma)}(\pi_{[1,n]}(C_{max,p}), S_{xu} \times D) \\ &\subseteq \pi_{[1,n]}(C_{max,p+k}) \subseteq C_{max,co}. \end{aligned} \tag{7.18}$$

By (7.17) and Lemma 7.4, we obtain the following inner bound of $\pi_{[1,n]}(C_{max,p_0+k})$:

$$\begin{aligned} &Pre^k_{\mathscr{D}(\Sigma)}(\lambda_0 C_{max,co}, S_{xu} \times D) \\ &\subseteq \pi_{[1,n]}(C_{max,p_0+k}) \subseteq C_{max,co}. \end{aligned} \tag{7.19}$$

Since $\lambda_0 C_{max,co}$ is a CIS of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$, the k-step backward reachable set of $\lambda_0 C_{max,co}$ in $S_{xu} \times D$ is expanding with k. If we can show that this k-step backward reachable set converges to the maximal CIS $C_{max,co}$ of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$ as k goes infinity, then by (7.19), the projection $\pi_{[1,n]}(C_{max,p})$ converges to the maximal CIS $C_{max,co}$. Furthermore, if we know how fast $Pre^k_{\mathscr{D}(\Sigma)}(\lambda_0 C_{max,co}, S_{xu} \times D)$ converges, then we have a lower bound on the convergence rate of $\pi_{[1,n]}(C_{max,p})$. The following theorem gives such a lower bound.

Theorem 7.5. *For a linear system Σ with a safe set S_{xu} satisfying Assumption 7.2, suppose that there exists a scalar $\gamma \in (0,1]$, a positive integer N and a scalar $\lambda \in [0,1]$ such that $\gamma C_{max,co}$ is an N-step λ-contractive CIS of the system $\mathscr{D}(\Sigma)$ within the safe set S_{xu}. Then, the kN-step backward reachable set of $\lambda_0 C_{max,co}$ satisfies that for $k \leq k_0$,*

$$\frac{\lambda_0}{\lambda^k} C_{max,co} \subseteq Pre^{kN}_{\mathscr{D}(\Sigma)}(\lambda_0 C_{max,co}, S_{xu} \times D); \tag{7.20}$$

for $k \geq k_0$,

$$\left(1-(1-\lambda_0/\lambda^{k_0})\left(\frac{1-\gamma}{1-\gamma\lambda}\right)^{k-k_0}\right)C_{max,co}$$
$$\subseteq Pre^{kN}_{\mathscr{D}(\Sigma)}(\lambda_0 C_{max,co}, S_{xu}\times D), \tag{7.21}$$

where

$$k_0 = \max\left(0, \left\lceil \frac{\log\lambda_0 - \log\gamma}{\log\lambda} - 1 \right\rceil\right). \tag{7.22}$$

It turns out that for stabilizable systems satisfying assumptions in Section 7.3.1, the N-step λ-contractive CIS stated in Theorem 7.5 always exists, shown by the following lemma.

Lemma 7.6. Under Assumptions 7.1, 7.2 and 7.3, there exist a scalar $\gamma \in (0,1]$, a positive integer N and a scalar $\lambda \in [0,1)$ such that $\gamma C_{max,co}$ is an N-step λ-contractive CIS of the system $\mathscr{D}(\Sigma)$ within the safe set S_{xu}, that is

$$\gamma C_{max,co} \subseteq Pre^{N}_{\mathscr{D}(\Sigma)}(\lambda\gamma C_{max,co}, S_{xu}). \tag{7.23}$$

By combining (7.19) with Lemma 7.6 and Theorem 7.5, the following theorem bounds the convergence of the projection $\pi_{[1,n]}(C_{max,p_0+kN})$.

Theorem 7.7. *Suppose that a system* Σ*, a safe set* S_{xu} *and a preview time* p_0 *satisfy Assumptions 7.1, 7.2 and 7.3. Then, there exists a scalar* $\gamma \in (0,1]$*, a positive integer* N *and a scalar* $\lambda \in [0,1)$ *such that the projection* $\pi_{[1,n]}(C_{max,p_0+kN})$ *satisfies that for* $k \leq k_0$*,*

$$\frac{\lambda_0}{\lambda^k} C_{max,co} \subseteq \pi_{[1,n]}(C_{max,p_0+kN}) \subseteq C_{max,co}; \tag{7.24}$$

for $k \geq k_0$*,*

$$\left(1-ca^k\right)C_{max,co} \subseteq \pi_{[1,n]}(C_{max,p_0+kN}) \subseteq C_{max,co}, \tag{7.25}$$

where k_0 *is defined by* (7.22)*,* $a = (1-\gamma)/(1-\gamma\lambda)$ *and* $c = (1-\lambda_0/\lambda^{k_0})a^{k_0}$.

In the case of $\lambda_0 = 0$, Theorem 7.7 is trivial as $k_0 = +\infty$ and the right most term in (7.24) becomes $\{0\}$. That is why we need Assumption 7.4 to enforce $\lambda_0 > 0$ and exclude this trivial case. The results in this section is summarized by the following corollary.

Corollary 7.7.1. Under Assumptions 7.1, 7.2, 7.3 and 7.4, the projection of $C_{max,p}$ onto the first n-coordinates converges to the maximal RCIS $C_{max,co}$ of the disturbance collaborative system $\mathscr{D}(\Sigma)$

in Hausdorff distance, that is

$$d_p = d(\pi_{[1,n]}(C_{max,p}), C_{max,co}) \xrightarrow{p\to\infty} 0.$$

Furthermore, the Hausdorff distance d_p satisfies the following inequality:
For $p_0 \le p < p_0 + N(k_0+1)$,

$$d_p \le (1 - \lambda_0 \lambda^{-\lfloor (p-p_0)/N \rfloor}) r_{co}; \tag{7.26}$$

for $p \ge p_0 + N(k_0+1)$,

$$d_p \le c a^{\lfloor (p-p_0)/N \rfloor} r_{co}, \tag{7.27}$$

with r_{co} the radius of $C_{max,co}$ with respect to 0. The other constants k_0, N, λ_0, λ, a and c are the same as in Theorem 7.7.

According to (7.27), as we increase the preview time, the safety regret decays exponentially fast. We further define the marginal value Δd_p of the preview at preview time p as the Hausdorff distance between $\pi_{[1,n]}(C_{max,p})$ and $\pi_{[1,n]}(C_{max,p+N})$. Then, by (7.27), for $p \ge p_0 + N(k_0+1)$,

$$\Delta d_p = d(\pi_{[1,n]}(C_{max,p}), \pi_{[1,n]}(C_{max,p+N})) \tag{7.28}$$

$$\le d_p + d_{p+N} \le c(1+a) a^{\lfloor (p-p_0)/N \rfloor} r_{co}. \tag{7.29}$$

Thus, we show that the marginal value of preview decays exponentially fast as p increases.

7.3.3 Estimating the parameters in Theorem 7.7

In this section, we show how to numerically obtain an exponentially decaying upper bound of d_p predicted by Theorem 7.7 for a given system. Specifically, we propose a sequence of optimization programs to estimate the parameters λ_0, γ, N and λ in Theorem 7.7. To make the computation tractable, we assume that the safe set S_{xu} and the disturbance set D are represented by polytopes. In case where the maximal CIS $C_{max,co}$ and the maximal RCIS C_{max,p_0} are not polytopes, we use a polytopic outer approximation of $C_{max,co}$ and a polytopic inner approximation of C_{max,p_0} instead, which results in a lower estimate of λ_0. Clearly, results in Section 7.3.2 still hold when λ_0 is replaced by its lower estimate. There is a rich literature of computing polytopic inner or outer approximations of the maximal RCIS for discrete-time linear systems (see [103] for example).

Step 1: We check if the origin is a forced equilibrium of $\mathscr{D}(\Sigma)$ that satisfies Assumptions 7.3 and 7.4. If not, we find a forced equilibrium (x_e, u_e, d_e) satisfying Assumptions 7.3 and 7.4 by the

following linear program:

$$\begin{aligned} \varepsilon^* = \max_{\varepsilon \geq 0, x_e, u_e, d_e} \ & \varepsilon \\ \text{s.t. } & Ax_e + Bu_e + Ed_e = x_e, \\ & x_e + \varepsilon \mathbf{B}(n) \subseteq \pi_{[1,n]}(C_{max,p_0}), \\ & (x_e, u_e, d_e) + \varepsilon \mathbf{B}(n+m+l) \subseteq S_{xu} \times D. \end{aligned} \tag{7.30}$$

Given the H-representations of C_{max,p_0}, S_{xu} and D, the second and third constraints of (7.30) can be easily encoded as linear inequality constraints by iterating vertices of the hypercubes $\mathbf{B}(n)$ and $\mathbf{B}(n+m+l)$.

When $\varepsilon^* > 0$, the solution (x_e, u_e, d_e) of (7.30) is a feasible forced equilibrium satisfying Assumptions 7.3 and 7.4. Then, as highlighted in Remark 7.1, we shift the origin of the state-input-disturbance space to the forced equilibrium computed in (7.30), that is to shift the sets S_{xu}, D, $C_{max,co}$ and C_{max,p_0} accordingly.

Step 2: We want to compute (lower-estimates of) the initial factor λ_0. We first introduce a baseline method, with two steps: The first step is to find the maximal λ such that $\lambda C_{max,co} \subseteq \mathbf{B}(n)$. This scalar λ can be computed by a linear program. By the construction of (7.30), we know $\varepsilon^* \mathbf{B}(n) \subseteq \pi_{[1,n]}(C_{max,p_0})$, where ε^* is the optimal cost of (7.30). Thus, $\varepsilon^* \lambda C_{max,co} \subseteq \varepsilon^* \mathbf{B}(n) \subseteq \pi_{[1,n]}(C_{max,p_0})$. That is, $\varepsilon^* \lambda$ provides a lower estimate of λ_0. The benefits of this method include (i) whenever (7.30) returns a positive ε^* [2], the estimated λ_0 is guaranteed to be positive, and (ii) the computation is easy. The main drawback is that the estimated λ_0 can be very conservative.

Alternatively, according to (7.17), the estimation of λ_0 can be formulated as a polytope containment problem [104]: Let P_1 and P_2 be two polytopes with H-representation $[H_1 \ h_1]$ and $[H_2 \ h_2]$, where $h_i \in \mathbb{R}^{q_i}$ for $i = 1, 2$. Our goal is to find the maximal λ such that $\lambda P_1 \subseteq P_2$, which is equivalent to find the minimal r such that $P_1 \subseteq r P_2$. Then, according to Farka's Lemma, the minimal r can be obtained by the following linear program:

$$\begin{aligned} r^* = \min_{r, \Lambda \in R_+^{q_2 \times q_1}} \ & r \\ \text{s.t. } & \Lambda H_1 = H_2 \\ & \Lambda h_1 \leq r h_2. \end{aligned} \tag{7.31}$$

Thus, by replacing the polytopes P_1 and P_2 in (7.31) with $C_{max,co}$ and $\pi_{[1,n]}(C_{max,p_0})$ (or their polytopic approximations), we obtain an estimate of λ_0 as the reciprocal of the optimal solution r^* of (7.31).

[2] Note that if (7.30) returns $\varepsilon^* = 0$, there is no need to compute λ_0 anymore, as Assumption 7.4 cannot be verified.

This alternative method returns more accurate λ_0 than the baseline. When the H-repesentations of $C_{max,co}$ and $\pi_{[1,n]}(C_{max,p_0})$ are exact (instead of approximated), λ_0 estimated by (7.31) matches the true λ_0. But, this method is more time consuming than the baseline. Recall that we only have the H-representation of C_{max,p_0}, but (7.31) needs the H-representation of the projection $\pi_{[1,n]}(C_{max,p_0})$ of C_{max,p_0}. The projection operation of polytopes in H-representation is computationally expensive.

It is also possible to encode the polytope containment constraint in (7.17) directly based on the H-representations of $C_{max,co}$ and C_{max,p_0}, which enables us to estimate λ_0 without the projection step [104]: Suppose that the H-representations of $C_{max,co}$ and C_{max,p_0} are $[H_1\ h_1]$ and $[H_2\ h_2]$ with $h_i \in \mathbb{R}^{q_i}$, $i = 1, 2$. Then, λ_0 can be estimated by the reciprocal of the optimal solution of the following linear program:

$$\begin{aligned} \lambda_0^{-1} = \min_{r,\Gamma\in\mathbb{R}^{n_p\times n},\beta\in\mathbb{R}^{n_p},\Lambda\in R_+^{q_2\times q_1}} \ & r \\ \text{s.t. } & \Lambda H_1 = H_2\Gamma \\ & \Lambda h_1 \le r h_2 + H_2\beta \\ & \mathscr{P}\Gamma = I, \\ & \mathscr{P}\beta = 0, \end{aligned} \tag{7.32}$$

where $n_p = n + p_0 l$, I is the $n \times n$ identity matrix, and $\mathscr{P}$ is the projection matrix that maps points in $\mathbb{R}^{n_p}$ onto the first n coordinates. The linear program in (7.32) is formulated based on a sufficient condition of polytope containment in [104]. Thus, λ_0 estimated by (7.32) is more conservative than λ_0 estimated by (7.31).

To summarize, we propose three methods to estimate λ_0, with different conservativeness and computation cost. As discussed in Section 7.3.2, the estimate of λ_0 is meaningful only when it is positive, which can only be guaranteed by the first two methods. Thus, in practice, we suggest users to first obtain a positive baseline estimate of λ_0 via the first method, and then select the second or third method, based on the available computation power, for a potentially better estimate of λ_0.

Step 3: We propose a method to find feasible γ, N and λ, inspired by the proof of Lemma 7.6. First, compute a λ_a-contractive ellipsoidal CIS of $\mathscr{D}(\Sigma)$ for some λ_a in $[0, 1)$. Here we formulate a

bilinear program[3] to calculate the minimal possible λ_a:

$$\begin{aligned}
&\lambda_a^2 = \min_{r\geq 0, Q\succ 0, R_1, R_2} \; r \\
&\text{subject to} \\
&\begin{bmatrix} Q & AQ+BR_1+ER_2 \\ (AQ+BR_1+ER_2)^T & rQ \end{bmatrix} \succ 0.
\end{aligned} \tag{7.33}$$

The bilinear program in (7.33) can be solved by line search over r. For any stabilizable system, the optimal value λ_a^2 of (7.33) must be smaller than one[4]. Given the optimal solution $r = \lambda_a^2$, Q, R_1 and R_2 of (7.33), the ellipsoid $\mathscr{E}(c) = \{x \mid x^T Q^{-1} x \leq c^2\}$ is a λ_a-contractive CIS of $\mathscr{D}(\Sigma)$ for any $c > 0$, and the controller $u = R_1 Q^{-1} x$, $u_d = R_2 Q^{-1} x$ is a safe controller for any state x in $\mathscr{E}(c)$.

Let c_0 be a scalar such that $(x, R_1 Q^{-1} x)$ is in S_{xu} and $R_2 Q^{-1} x$ is in D for all $x \in \mathscr{E}(c_0)$. Let c_{out} be a scalar such that $C_{max,co} \subseteq \mathscr{E}(c_{out})$. The second step of finding γ, N and λ is to estimate the maximal c_0 and the minimal c_{out}.

Suppose that the Cholesky decomposition of the positive definite matrix Q is $Q = LL^T$ for some invertible matrix $L \in \mathbb{R}^{n\times n}$. Then, the ellipsoid $\mathscr{E}(c)$ can be represented equivalently by $\mathscr{E}(c) = \{cLs \mid s^T s \leq 1\}$. By Section 8.4.2 of [24], the maximal c_0 is given by the optimal value of the following linear program:

$$\begin{aligned}
&c_0 = \max_{c\geq 0} \; c \\
&\text{subject to} \\
&c \| [L^T \; L^{-1} R_1^T] H_{xu,i}^T \|_2 \leq h_{xu,i}, \quad i \in [q_{xu}] \\
&c \| L^{-1} R_2^T H_{D,j}^T \|_2 \leq h_{D,j}, \quad j \in [q_d],
\end{aligned} \tag{7.34}$$

where $[H_{xu,i} \; h_{xu,i}]$ is the i th row of the H-representation $[H_{xu} \; h_{xu}]$ of S_{xu}, and $[H_{D,j} \; h_{D,j}]$ is the j th row of the H-repsentation $[H_D \; h_D]$ of D, and q_{xu} and q_d are the numbers of rows of H_{xu} and H_D.

Suppose that the set of vertices of $C_{max,co}$ is $\mathscr{V}$. Then, the square of the minimal c_{out} is equal to $c_{out} = \max_{v\in\mathscr{V}} \sqrt{v^T Q^{-1} v}$. However, it is usually time consuming to compute the vertices of $C_{max,co}$ given its H-representation. An alternative way is to first find the minimal bounding rectangle of $C_{max,co}$ and then use the minimal c such that $\mathscr{E}(c)$ contains the minimal bounding rectangle as a conservative estimate of c_{out}.

Once we have the maximal c_0 and a feasible c_{out}, a set of feasible γ, N and λ is given by the following theorem.

[3]The constraints in (7.33) that encode $\sqrt{r}$-contractive ellisoidal CISs can be found in Remark 4.1 of [22] and Section 4.4.2 of [23]

[4]See Proposition 23 of [32].

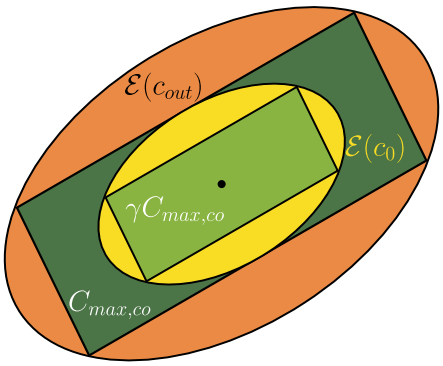


Figure 7.2: The inclusion relations among $\mathscr{E}(c_0)$ (yellow), $\mathscr{E}(c_{out})$ (orange), $C_{max,co}$ (dark green) and $\lambda C_{max,co}$ (light green).

Theorem 7.8. *Given the λ_a-contractive ellipsoidal CIS $\mathscr{E}(c)$ of $\mathscr{D}(\Sigma)$, the maximal c_0 and a feasible c_{out}, let*

$$\gamma = c_0/c_{out}, \tag{7.35}$$

$$N = \left\lfloor \frac{\log(\gamma)}{\log(\lambda_a)} \right\rfloor + 1, \tag{7.36}$$

$$\lambda = \lambda_a^N/\gamma. \tag{7.37}$$

Then $\gamma C_{max,co}$ is an N-step λ-contractive CIS of $\mathscr{D}(\Sigma)$, with $\gamma \leq 1$ and $\lambda < 1$.

The intuition behind Theorem 7.8 is: By the definition of c_0, the ellipsoid $\mathscr{E}(c_0)$ is a λ_a-contractive CIS of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$ and thus is contained in the maximal CIS $C_{max,co}$, as shown in Fig. 7.2. Then, for $\gamma = c_0/c_{out}$, we have $\gamma C_{max,co}$ contained in the λ_a-contractive CIS $\mathscr{E}(c_0)$. As $\mathscr{E}(c_0)$ can be controlled to reach an arbitrary small neighborhood of the origin over time (by being λ_a-contractive), for an arbitrary small λ, $\gamma C_{max,co}$ is N-step λ-contractive for a large enough N.

Step 4 (Refinement): If the H-representation of $C_{max,co}$ is accurate instead of an outer approximation, given the set of feasible parameters γ and N and λ in Theorem 7.8, we can find another set of feasible parameters which potentially lead to a tighter upper bound of d_p: Recall that $\gamma C_{max,co}$ is N-step λ-contractive. Let γ^* be the maximal scalar such that $Pre^N_{\mathscr{D}(\Sigma)}(\lambda\gamma C_{max,co}, S_{xu} \times D)$ contains $\gamma^* C_{max,co}$. Here γ^* can be solved by a linear program in form of (7.31). It can be shown that $\gamma^* C_{max,co}$ is N-step $\lambda\gamma/\gamma^*$-contractive. Then, we replace the estimated γ and λ by γ^* and $\lambda\gamma/\gamma^*$.

The entire procedure (Steps 1-4) of estimating λ_0, γ, N and λ is summarized into Alg. 5. One can disable the refinement step by commenting out the last *if* statement in Alg. 5. In Section 7.5.2, we show numerically that the estimated upper bound is much tighter when the refinement is applied.

Algorithm 5 Estimating parameters in Theorem 7.7

input: (A,B,E), S_{xu}, D, $C_{max,co}$, C_{max,p_0}
$(x_e,u_e,d_e) \leftarrow 0$
$\varepsilon^* \leftarrow$ solve the LP in (7.30) with (x_e,u_e,d_e) fixed to 0
if $\varepsilon^* = 0$ ▷ that is, (x_e,u_e,d_e) is not a forced equilibrium of $\mathscr{D}(\Sigma)$ satisfying Assumptions 7.3 and 7.4 **then**
    $(x_e,u_e,d_e), \varepsilon^* \leftarrow$ solve the LP in (7.30)
    $S_{xu} \leftarrow S_{xu} - (x_e,u_e)$
    $D \leftarrow D - d_e$ ▷ shift the origin to (x_e,u_e,d_e)
end if
$r^* \leftarrow$ solve the LP in (7.31) with $P_1 = C_{max,co}$, $P_2 = \mathbf{B}(n)$
$\lambda_0 \leftarrow \varepsilon^*/r^*$ (baseline estimate of λ_0)
if H-representation of $\pi_{[1,n]}(C_{max,p_0})$ available **then**
    $r^* \leftarrow$ solve the LP in (7.31) with $P_1 = C_{max,co}$, $P_2 = \pi_{[1,n]}(C_{max,p_0})$
else
    $r^* \leftarrow$ solve the LP in (7.32)
end if
$\lambda_0 \leftarrow \max(\lambda_0, 1/r^*)$ ▷ Pick the best estimate of λ_0
$\lambda_a^2, Q, R_1, R_2 \leftarrow$ solve the convex program in (7.33)
$c_0 \leftarrow$ solve the LP in (7.34)
if the set $\mathscr{V}$ of vertices of $C_{max,co}$ available **then**
    $c_{out} \leftarrow \max_{v\in\mathscr{V}} \sqrt{v^T Q^{-1} v}$
else
    $\mathscr{V} \leftarrow$ vertices of minimal rectangle containing $C_{max,co}$
    $c_{out} \leftarrow \max_{v\in\mathscr{V}} \sqrt{v^T Q^{-1} v}$
end if
$\gamma, N, \lambda \leftarrow$ RHSs of (7.35), (7.36) and (7.37)
if H-representation of $C_{max,co}$ is exact **then**
    $C_N \leftarrow Pre^N_{\mathscr{D}(\Sigma)}(\lambda\gamma C_{max,co}, S_{xu} \times D)$
    $\gamma^* \leftarrow$ solving LP in (7.31) with $P_1 = C_{max,co}$, $P_2 = C_N$
    $\gamma \leftarrow \gamma^*$, $\lambda \leftarrow \lambda\gamma/\gamma^*$
end if
return λ_0, γ, N, λ

7.3.4 Special case: controllable $\mathscr{D}(\Sigma)$

According to Corollary 7.7.1, the convergence of $\pi_{[1,n]}(C_{max,p_0+Nk})$ may have two phases, depending on whether k is greater or smaller than k_0. In this section, we show a single-phased convergence $\pi_{[1,n]}(C_{max,p})$ when the disturbance collaborative system $\mathscr{D}(\Sigma)$ is controllable. Furthermore, we show that Assumption 7.4 is no longer required for the convergence of $\pi_{[1,n]}(C_{max,p})$.

Assumption 7.5. *The disturbance collaborative system $\mathscr{D}(\Sigma)$ of the linear system Σ is controllable.*

The key observation for controllable disturbance collaborative system is stated by the following lemma.

Lemma 7.9. Under Assumptions 7.2, 7.3 and 7.5, for any $N \geq n$, there exists a scalar $\gamma \in (0,1]$ such that $\gamma C_{max,co}$ is a N-step 0-contractive CIS of the system $\mathscr{D}(\Sigma)$ within the safe set S_{xu}, that is

$$\gamma C_{max,co} \subseteq Pre^{N}_{\mathscr{D}(\Sigma)}(0, S_{xu}). \tag{7.38}$$

Combining Lemma 7.9 with Theorem 7.7, we bound the convergence of $\pi_{[1,n]}(C_{max,p})$ by the following theorem.

Theorem 7.10. *Suppose that a system Σ, a safe set S_{xu} and a preview time p_0 satisfy Assumptions 7.2, 7.3 and 7.5. For any $N \geq n$, let γ_{max} be the maximal γ such that* (7.38) *holds. Then, (i) $\gamma_{max} > 0$ and (ii) the projection $\pi_{[1,n]}(C_{max,p_0+kN})$ satisfies that for $k \geq 0$,*

$$\begin{aligned} C_{max,co} &\supseteq \pi_{[1,n]}(C_{max,p_0+kN}) \\ &\supseteq \left(1 - (1-\lambda_0)(1-\gamma_{max})^k\right) C_{max,co}. \end{aligned} \tag{7.39}$$

Proof. (Sketch) For the N-step 0-contractive CIS $\gamma C_{max,co}$ in Lemma 7.9, k_0 in (7.22) is 0 due to $\lambda = 0$. Then Theorem 7.7 with $k_0 = 0$ implies Theorem 7.10. □

It is obvious that the right hand of (7.39) converges to $C_{max,co}$ even if $\lambda_0 = 0$. Thus, here we do not need Assumption 7.4 to make $\lambda_0 > 0$.

Corollary 7.10.1. Under Assumptions 7.2, 7.3 and 7.5, the projection of $C_{max,p}$ onto the first n-coordinates converges to the maximal RCIS $C_{max,co}$ of the disturbance collaborative system $\mathscr{D}(\Sigma)$ in Hausdorff distance, that is

$$d_p = d(\pi_{[1,n]}(C_{max,p}), C_{max,co}) \xrightarrow{p\to\infty} 0.$$

Furthermore, the Hausdorff distance d_p satisfies the following inequality: For $p \geq p_0$,

$$d_p \leq (1-\lambda_0)(1-\gamma_{max})^{\lfloor (p-p_0)/N \rfloor} r_{co}, \tag{7.40}$$

with N and γ_{max} in Theorem 7.10, and r_{co} the radius of $C_{max,co}$ with respect to 0.

Compared with the upper bound on d_p in Corollary 7.7.1, the upper bound in (7.40) does not have the burn-in time k_0 and is easier to compute, which is shown next.

We propose an algorithm that numerically calculates the parameters λ_0 and γ_{max} in Theorem 7.10 for controllable $\mathscr{D}(\Sigma)$. The same as in Section 7.3.3, we assume that the safe set S_{xu} and the disturbance set D are polytopes, and the sets $C_{max,co}$ and C_{max,p_0} are replaced by their polytopic outer/inner approximations when they are not polytopes.

Step 1: We first check if the origin as a forced equilibrium satisfies Assumption 7.3. If not, we find a feasible forced equilibrium via a linear program modified from (7.30), where we replace the constraint $x_e + \varepsilon \mathbf{B}(n) \subseteq \pi_{[1,n]}(C_{max,p_0})$ in (7.30) by $x_e \subseteq \pi_{[1,n]}(C_{max,p_0})$.

When the optimal value $\varepsilon^* > 0$, the optimal solution (x_e, u_e, d_e) of the modified linear program is a feasible forced equilibrium satisfying Assumptions 7.3. Then, by Remark 7.1, we shift the origin of the state-input-disturbance space to this forced equilibrium.

Step 2: We compute λ_0 by the second or the third methods of estimating λ_0 proposed in Section 7.3.3. Since $\lambda_0 > 0$ is no longer necessary, one may prefer the third method for less computation cost.

Step 3: We select a step size N and estimate the corresponding γ_{max}. Since S_{xu} and D are polytopes, the N-step backward set $Pre^N_{\mathscr{D}(\Sigma)}(\{0\}, S_{xu} \times D)$ is a polytope, whose H-representation can be easily computed. Then, solving the maximal γ satisfying (7.38), that is γ_{max}, is again a polytope containment problem. Thus, γ_{max} is equal to the reciprocal of the optimal value r^* of (7.31) with $P_1 = C_{max,co}$ and $P_2 = Pre^N_{\mathscr{D}(\Sigma)}(0, S_{xu} \times D)$.

The procedure (Steps 1-3) of estimating λ_0 and γ_{max} is summarized into Alg. 6. Note that different from Alg. 5, the step size N here can be freely selected by users. By Lemma 7.9, for any $N \geq n$, Alg. 6 is guaranteed to find a nonzero γ_{max}. In Section 7.5.2, we show numerically how different N affects the estimated upper bound of d_p.

7.3.5 On the finite-time convergence of $\pi_{[1,n]}(C_{max,p})$

In practice, $\pi_{[1,n]}(C_{max,p})$ may converge to $C_{max,co}$ in finite preview time p, which is not reflected by the exponentially decaying upper bounds in Theorems 7.7 and 7.10. In this section, we propose a simple algorithm to detect the potential finite-time convergence of $\pi_{[1,n]}(C_{max,p})$.

Suppose that the projection $\pi_{[1,n]}(C_{max,p_0})$ is known for some p_0. According to Lemma 7.4, $\pi_{[1,n]}(C_{max,p})$ is equal to $C_{max,co}$ for $p = p_0 + k$ if

$$Pre^k_{\mathscr{D}(\Sigma)}(\pi_{[1,n]}(C_{max,p_0}), S_{xu} \times D) = C_{max,co}. \tag{7.41}$$

Based on the above sufficient condition, we propose Alg. 7 to detect the finite-time convergence of

Algorithm 6 Estimating parameters in Theorem 7.10

input: N, (A,B,E), S_{xu}, D, $C_{max,co}$, C_{max,p_0}
$(x_e,u_e,d_e) \leftarrow 0$
$\varepsilon^* \leftarrow$ solve the LP modified from (7.30) with (x_e,u_e,d_e) fixed to 0
if $\varepsilon^* = 0$ ▷ that is, (x_e,u_e,d_e) is not a forced equilibrium of $\mathscr{D}(\Sigma)$ satisfying Assumption 7.3 **then**
  $(x_e,u_e,d_e), \varepsilon^* \leftarrow$ solve the LP modified from (7.30)
  $S_{xu} \leftarrow S_{xu} - (x_e,u_e)$
  $D \leftarrow D - d_e$ ▷ shift the origin to (x_e,u_e,d_e)
end if
if H-representation of $\pi_{[1,n]}(C_{max,p_0})$ available **then**
  $r^* \leftarrow$ solve the LP in (7.31) with $P_1 = C_{max,co}$, $P_2 = \pi_{[1,n]}(C_{max,p_0})$
else
  $r^* \leftarrow$ solve the LP in (7.32)
end if
$\lambda_0 \leftarrow 1/r^*$
$C_N \leftarrow Pre^N_{\mathscr{D}(\Sigma)}(\{0\}, S_{xu} \times D)$
$r^* \leftarrow$ solve LP in (7.31) with $P_1 = C_{max,co}$, $P_2 = C_N$.
$\gamma_{max} \leftarrow 1/r^*$
return λ_0, γ_{max}

$\pi_{[1,n]}(C_{max,p_0})$. Note that we set a maximal iteration number k_{max} to guarantee the termination of

Algorithm 7 Detecting finite-time convergence

input: $C_{max,co}$, $\pi_{[1,n]}(C_{max,p_0})$, k_{max}
$k \leftarrow 0$, $C_0 \leftarrow \pi_{[1,n]}(C_{max,p_0})$, $\overline{p} \leftarrow \infty$
if $C_0 = C_{max,co}$ **then** $\overline{p} \leftarrow p_0$
end if
while $k < k_{max}$ **do**
  $k \leftarrow k+1$, $C_k \leftarrow Pre_{\mathscr{D}(\Sigma)}(C_{k-1}, S_{xu} \times D)$,
  if $C_{max,co} = C_k$ **then** $\overline{p} \leftarrow p_0 + k$; break
  end if
end while
return $\overline{p}$, $\{C_k\}_{k=0}^{\min(\overline{p},k_{max})}$

Alg. 7. In case where Alg. 7 returns a finite number $\overline{p}$, we know thta $\pi_{[1,n]}(C_{max,\overline{p}})$ must be equal to $C_{max,co}$. In terms of the computational cost, given the H-representation of $\pi_{[1,n]}(C_{max,p_0})$, the k-step backward reachable set of $\pi_{[1,n]}(C_{max,p_0})$ with respect to $\mathscr{D}(\Sigma)$ is much cheaper to compute than $\pi_{[1,n]}(C_{max,p+k})$, since the dimensions of the system $\mathscr{D}(\Sigma)$ are independent of the preview time p. Thus, Alg. 7 is more tractable than directly checking if $\pi_{[1,n]}(C_{max,p}) = C_{max,co}$ for $p \geq 0$.

As a side product, the Hausdorff distance between the backward reachable set C_k computed in Alg. 7 and $C_{max,co}$ gives an upper bound of d_{p_0+k} for $k = 0, \cdots, \min(\overline{p} - p_0, k_{max})$, tighter than those

obtained by Algs. 5 and 6. However, Alg. 7 is more computationally expensive than Algs. 5 and 6 due to the iterative backward reachable set computation (and the Hausdorff distance computation). Another drawback is that Alg. 7 can only provide the upper bounds of d_p for finitely many p.

7.4 Model Predictive Control with Preview

In this section, we study the impact of preview to the feasible domain of constrained MPC. We consider the class of discrete-time linear systems Σ as in (7.16) with p-step preview. Let the MPC planning horizon be the preview time p and the constraints on state and input to be the safe set S_{xu} of Σ for simplicity. Given the current state x_0 and preview information $d_{0:p-1}$, a standard optimization problem to be solved by MPC at each time step is:

$$\begin{aligned} \min_{x_{1:p}, u_{0:p-1}} \; & l_F(x_p, u_{p-1}) + \sum_{t=1}^{p-1} l_t(x_t, u_{t-1}) \\ \text{s.t. } & x_t = Ax_{t-1} + Bu_{t-1} + Ed_{t-1}, \\ & (x_{t-1}, u_{t-1}) \in S_{xu}, \forall t \in [p], \\ & \text{RFC}, \end{aligned} \tag{7.42}$$

where l_t and l_F are the stage cost at time t and the final cost, and RFC is a placeholder for constraints that guarantee the recursive feasibility of (7.42).

Definition 7.2. The feasible domain $\mathscr{F}$ of the MPC in (7.42) is the set of initial conditions $(x_0, d_{0:p-1})$ that satisfy the constraints in (7.42).

To guarantee the recursive feasibility of the MPC, we need to impose certain recursive feasibility constraints RFC in (7.42) such that the feasible domain $\mathscr{F}$ is an RCIS of the p-augmented system Σ_p in $S_{xu,p}$. Depending on the choice of the RFC, the size of feasible domain can be very different. Here we study two RFCs.

First, let RFC in (7.42) be

$$(x_1, d_{2:p-1}, d_p) \in C_{max,p}, \forall d_p \in D. \tag{7.43}$$

The constraints in (7.43) is the least conservative RFC, since the corresponding feasible domain is the maximal RCIS $C_{max,p}$ of Σ_p in $S_{xu,p}$, the largest feasible domain any RFC can produce.

Since the maximal RCIS $C_{max,p}$ is hard to compute as p increases, a more common RFC in practice is to force the final state x_p to be in an RCIS C of the original system Σ in S_{xu}, that is

$$x_p \in C. \tag{7.44}$$

The following theorem characterizes the relation between the terminal constraint set C and the corresponding feasible domain.

Theorem 7.11. *The feasible domain, denoted by $\mathscr{F}_p(C)$, of the MPC corresponding to the RFC as in* (7.44) *is the p-step backward reachable set of $C \times D^p$ for Σ_p in $S_{xu,p}$. That is,*

$$\mathscr{F}_p(C) = Pre^p_{\Sigma_p}(C \times D^p, S_{xu,p}). \tag{7.45}$$

Proof. Combining constraints in (7.42) and (7.44), it is easy to check that $\mathscr{F}_p(C) \subseteq Pre^p_{\Sigma_p}(C \times D^p, S_{xu,p})$. It is remained to show the other inclusion direction.

Suppose $(x_0, d_{0:p-1}) \in Pre^p_{\Sigma_p}(C \times D^p, S_{xu,p})$. Then, for arbitrary $d_{p:2p-1} \in D^p$, there exists $u_{0:p-1}$ such that for all $t = 0, \cdots, p-1$,

$$(x_t, d_{t:t+p-1}, u_t) \in S_{xu,p}, \tag{7.46}$$

$$(x_p, d_{p:2p-1}) \in C \times D^p. \tag{7.47}$$

That is, $(x_t, u_t) \in S_{xu}$ for $t = 0, \cdots, p-1$ and $x_p \in C$, which implies $(x_0, d_{0:p-1}) \in \mathscr{F}_p(C)$. □

By (7.45), $\mathscr{F}_p(C)$ is an RCIS of Σ_p in $S_{xu,p}$ and thus is a subset of $C_{max,p}$. Intuitively, $\mathscr{F}_p(C)$ is the set of states where the system is guaranteed to stay in the safe set indefinitely even if no preview is available after the first p steps, and thus is more conservative than $C_{max,p}$.

We want to compare how the gap between $\mathscr{F}_p(C)$ and $C_{max,p}$ changes as p increases. Since the dimensions of the two feasible domains $C_{max,p}$ and $\mathscr{F}_p(C)$ increases with p, a direct comparison is intractable. Similar to Section 7.3, we instead compare the gap between the projections of $\mathscr{F}_p(C)$ and $C_{max,p}$ onto the first n dimensions.

Lemma 7.12. The projection of $\mathscr{F}_p(C)$ is equal to the p-step backward reachable set of C with respect to $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$. That is,

$$\pi_{[1,n]}(\mathscr{F}_p(C)) = Pre^p_{\mathscr{D}(\Sigma)}(C, S_{xu} \times D). \tag{7.48}$$

Proof. Let $x(0) \in \pi_{[1,n]}(\mathscr{F}_p(C))$. There exists $d_{1:p}(0) \in D^p$ such that $(x(0), d_{1:p}(0)) \in \mathscr{F}_p(C)$. Since by Theorem 7.11 $\mathscr{F}_p(C) = Pre^p_{\Sigma_p}(C \times D^p, S_{xu,p})$, for the system Σ_p as in (7.2), there exist inputs $\{u(t)\}_{t=0}^{p-1}$ such that $(x(t), u(t)) \in S_{xu}$ for $t = 0, \cdots, p-1$ and $x(p) \in C$ for all possible $\{d(t)\}_{t=0}^{p} \subseteq D$. That implies $x(0) \in Pre^p_{\mathscr{D}(\Sigma)}(C, S_{xu} \times D)$. Thus, $\pi_{[1,n]}(\mathscr{F}_p(C))$ is a subset of $Pre^p_{\mathscr{D}(\Sigma)}(C, S_{xu} \times D)$.

Next, we want to prove the other direction. Let $x(0) \in Pre^p_{\mathscr{D}(\Sigma)}(C, S_{xu} \times D)$. Then, for the system $\mathscr{D}(\Sigma)$ as in (7.5) , there exist inputs $\{u(t)\}_{t=0}^{p-1}$ and $\{u_d(t)\}_{t=0}^{p-1} \subseteq D$ such that $(x(t), u(t)) \in$

S_{xu} for $t = 0, \cdots, p-1$ and $x(p) \in C$. That implies $(x(0), u_d(0), \cdots, u_d(p-1))$ is contained by $Pre^p_{\Sigma_p}(C \times D, S_{xu,p}) = \mathscr{F}_p(C)$ by Theorem 7.11. Thus, $x(0) \in \pi_{[1,n]}(\mathscr{F}_p(C))$. □

By (7.48), it is clear that the projection $\pi_{[1,n]}(\mathscr{F}_p(C))$ is a CIS of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$. Recall that $C_{max,co}$ is the maximal CIS of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$. Thus, we have

$$\pi_{[1,n]}(\mathscr{F}_p(C)) \subseteq \pi_{[1,n]}(C_{max,p}) \subseteq C_{max,co}. \tag{7.49}$$

Following similar steps in Section 7.3, we can show the convergence rate of the projection $\pi_{[1,n]}(\mathscr{F}_p(C))$ over the planning horizon p: First, we modify Assumptions 7.3 and 7.4 by replacing $\pi_{[1,n]}(C_{max,p_0})$ with the set C, and call the modified assumptions Assumption 7.3' and 7.4 '. Let λ_0 be the maximal λ such that $\lambda C_{max,co} \subseteq C$, which is greater than 0 under Assumption 7.4'. Then, the following theorem shows that $\pi_{[1,n]}(\mathscr{F}_p(C))$ converges to $C_{max,co}$ exponentially fast.

Theorem 7.13. *Under Assumptions 7.1, 7.2, 7.3' and 7.4' (or Assumptions 7.2, 7.3' and 7.5), the projection $\pi_{[1,n]}(\mathscr{F}_p(C))$ converges to the maximal RCIS $C_{max,co}$ of $\mathscr{D}(\Sigma)$ in Hausdorff distance, that is*

$$d(\pi_{[1,n]}(\mathscr{F}_p(C)), C_{max,co}) \xrightarrow{p \to \infty} 0. \tag{7.50}$$

Furthermore, there exist some constants $c > 0$ and $a \in [0,1)$ such that for $p \geq 0$,

$$d(\pi_{[1,n]}(\mathscr{F}_p(C)), C_{max,co}) \leq ca^p. \tag{7.51}$$

Proof. By (7.48), we want to show that $Pre^p_{\mathscr{D}(\Sigma)}(C, S_{xu} \times D)$ converges to the maximal CIS $C_{max,co}$ exponentially fast under the assumptions. The arguments to show this convergence is very similar to those in the proofs of Theorems 7.7 and 7.10 (by replacing $C_{max,p}$ with $\mathscr{F}_p(C)$) and thus is omitted. □

Remark 7.2. Since the proof of Theorem 7.13 is similar to those of Theorems 7.7 and 7.10, one may expect that the upper bound on $d(\pi_{[1,n]}(\mathscr{F}_p(C)), C_{max,co})$ is in form of (7.26), (7.27) or (7.40). It is indeed the case, but we relax those tighter bounds to the RHS of (7.51) for simplicity. The tighter upper bounds can be retrieved from Section 7.3 by replacing $\pi_{[1,n]}(C_{max,p})$ there with C.

According to (7.49) and Theorem 7.13, we have

$$\begin{aligned} &d(\pi_{[1,n]}(\mathscr{F}_p(C)), \pi_{[1,n]}(C_{max,p})) \\ &\leq d(\pi_{[1,n]}(\mathscr{F}_p(C)), C_{max,co}) \leq ca^p. \end{aligned}$$

That is, the gap between the projections of $\mathscr{F}_p(C)$ and $C_{max,p}$ decays exponentially fast under the conditions in Theorem 7.13. Thus, even if $\mathscr{F}_p(C)$ is more conservative than $C_{max,p}$, as the planning horizon p is increased, this conservativeness decays fast. Thus, in practice, the RFC in (7.44) with a long enough planning horizon p is usually a good choice. Also, the constants c and a in Theorem 7.13 can be estimated by some modified Algs. 5 and 6 that replace $\pi_{[1,n]}(C_{max,p_0})$ in both algorithms by C. Once c and a are obtained, users can simply compute ca^p to quantitatively evaluate the conservativeness of the RFC in (7.44) with respect to the maximal feasible domain $C_{max,p}$ for any given planning horizon p. Besides, similar to Section 7.3.5, the potential finite-time convergence of the projections of $\mathscr{F}_p(C)$ can be detected by a modified Alg. 7 that replaces $\pi_{[1,n]}(C_{max,p_0})$ with C.

7.5 Illustrative Examples

7.5.1 One-dimensional systems

Consider the 1-dimensional system [78]

$$\Sigma : x(t+1) = ax(t) + u(t) + d(t), \tag{7.52}$$

with $x(t)$, $u(t) \in \mathbb{R}$ and $d(t) \in [-\overline{d}, \overline{d}]$. The safe set $S_{xu} = [-\overline{x}, \overline{x}] \times [-\overline{u}, \overline{u}]$.

Suppose that the parameters a, $\overline{x}$, $\overline{d}$, $\overline{u}$ and p satisfy $a > 1$, $\overline{x} \geq (\overline{u} + \overline{d})/(a-1)$ and $a^{p-1}\overline{u} \geq \overline{d}$. Then, it is shown in [78] that the maximal RCIS $C_{max,p}$ of the p-augmented system of Σ within the augmented safe set $[-\overline{x}, \overline{x}] \times [-\overline{d}, \overline{d}]^p \times [-\overline{u}, \overline{u}]$ is

$$C_{max,p} = \left\{ (x, d_{1:p}) \;\middle|\; |d_i| \leq \overline{d}, \forall i \in \{1, \cdots, p\}, \right. \\ \left. |x + \sum_{i=1}^{p} \frac{d_i}{a^i}| \leq \frac{\overline{u} - \overline{d}/a^p}{a-1} \right\}. \tag{7.53}$$

The projection of the maximal controlled invariant set onto the first coordinate is

$$PROJ_1(C_{max,p}) = \left[-\frac{\overline{u} + \overline{d} - 2\overline{d}/a^p}{a-1}, \frac{\overline{u} + \overline{d} - 2\overline{d}/a^p}{a-1} \right]. \tag{7.54}$$

The disturbance-collaborative system with respect to (7.52) is

$$\mathscr{D}(\Sigma) : x(t+1) = ax(t) + u, \tag{7.55}$$

with the safe set $S_{xu,co} = [-\overline{x}, \overline{x}] \times [-\overline{u} - \overline{d}, \overline{u} + \overline{d}]$. The maximal controlled invariant set $C_{max,co}$ of $\mathscr{D}(\Sigma)$ in the safe set is $[-(\overline{u} + \overline{d})/(a-1), (\overline{u} + \overline{d})/(a-1)]$. Thus, for any p such that $a^{p-1}\overline{u} \geq \overline{d}$,

the Hausdorff distance between $PROJ_1(C_{max,p})$ and $C_{max,co}$ satisfies

$$d(PROJ_1(C_{max,p}), C_{max,co}) = \frac{2\overline{d}}{(a-1)a^p}, \tag{7.56}$$

which decays to 0 with exponential rate $1/a$ as p goes to infinity.

Note that this one-dimensional system is simple enough for us to solve Alg. 6 by hands: Since $Pre_{\mathscr{D}(\Sigma)}(0, S_{xu,co}) = [-(\overline{u}+\overline{d})/a, (\overline{u}+\overline{d})/a]$, the value of γ_{max} in Theorem 7.10 is $(1-1/a)$. We pick any p_0 satisfying $a^{p_0-1}\overline{u} \geq \overline{d}$. Then, by (7.54), λ_0 is equal to

$$\lambda_0 = 1 - \frac{2\overline{d}/a^{p_0}}{(\overline{u}+\overline{d})}. \tag{7.57}$$

The radius ψ of $C_{max,co}$ with respect to 0 is $(\overline{u}+\overline{d})/(a-1)$. Plugging γ_{max}, λ_0 into (7.40), for all $p \geq p_0$, the Hausdorff distance between $PROJ_1(C_{max,p})$ and $C_{max,co}$ satisfies

$$d(PROJ_1(C_{max,p}), C_{max,co}) \leq \frac{2\overline{d}}{(a-1)a^p}. \tag{7.58}$$

Comparing the right hand sides of (7.56) and (7.58), the upper bound of the Hausdorff distance d_p obtained by Alg. 6 is equal to the actual value in this toy example.

7.5.2 Two-dimensional system with random safe set

We consider the following 2-dimensional system Σ:

$$\Sigma : x^+ = \begin{bmatrix} 1.5 & 1 \\ 0 & 1.1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 1 \\ 1 \end{bmatrix} d, \tag{7.59}$$

with $x \in \mathbb{R}^2$, $u \in \mathbb{R}$ and $d \in [-0.3, 0.3]$. The safe set S_{xu} is randomly generated, shown by the dark blue polytope in Fig. 7.3. Assume that the preview on d is available. We select $p_0 = 1$. The projection of C_{max,p_0} onto $\mathbb{R}^2$ is shown by the yellow polytope in Fig. 7.3. The projections of C_{max,p_0+k} for $k = 1, \cdots, 8$ are shown by the nested cyan polytopes in Fig. 7.3. The set $C_{max,co}$ is equal to the projection of $C_{max,p}$ with $p = 9$, shown by the largest cyan polytope in Fig. 7.3.

Since Σ is controllable, we apply both Algs. 5 and 6 to this example. The parameters estimated by Alg. 1 (with or without Step 4) and Alg. 2 ($N = 1$ or 8) are listed in Table 7.1. Plugging the parameters in Table 7.1 into (7.26), (7.27) and (7.40), we obtain four upper bounds of the Hausdorff distance d_p in (7.15), as depicted in Fig. 7.4a. Note that the red curve in Fig. 7.4a lies below the blue curve, showing that the refinement step in Alg. 5 indeed helps us find a tighter upper bound on

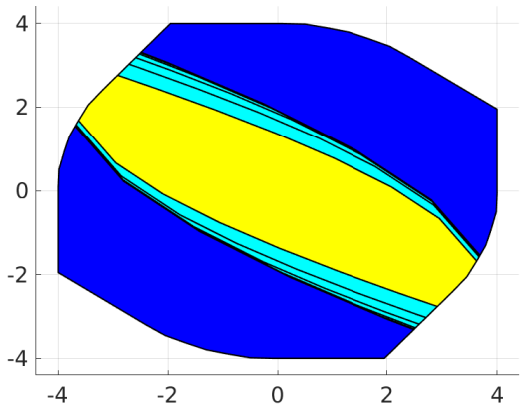


Figure 7.3: The projections of C_{max,p_0} (yellow polytope), C_{max,p_0+k} (cyan polytopes) with $k = 1,\cdots,8$ and the safe set (dark blue polytope).

Table 7.1: Parameters estimated by Algs. 5 and 6

	λ_0	γ or γ_{max}	N	λ
Alg. 5 w.o. Step 4	0.6550	0.0402	1	0.0011
Alg. 5 w. Step 4	0.6550	0.0752	1	5.737×10^{-4}
Alg. 6 ($N = 1$)	0.6550	0.0752	1	0
Alg. 6 ($N = 8$)	0.6550	0.9794	8	0

d_p. The purple curve obtained by Alg. 6 is coarser than the other curves due to a larger step size ($N = 8$), but is also decaying faster than the others as p increases. This observation implies that selecting a larger N in Alg. 6 may lead to a coarser but faster decaying upper bound. Among all the upper bounds in Fig. 7.4a, Alg. 5 with Step 4 and Alg. 6 ($N = 1$) find the tightest upper bounds (red and yellow curves) when p is small; Alg. 6 ($N = 8$) finds the tightest bound (purple curve in Fig. 7.4a) when p is large.

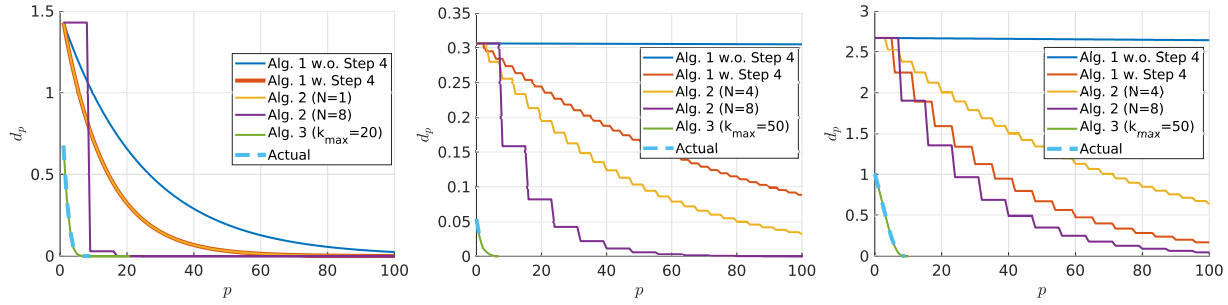
We also run Alg. 7 with $k_{max} = 50$, which terminates with $\overline{p} = \infty$. That indicates $\pi_{[1,n]}(C_{max,p})$ may not converge to $C_{max,co}$ for a finite p. The upper bound of d_p obtained as the side product of Alg. 7 is the tightest, coinciding with the actual d_p shown by the green curve in Fig. 7.4a. The computation time of the three algorithms is listed in the first column of Table 7.2.

7.5.3 Lane-keeping control

We consider the linearized bicycle model with respect to the longitudinal velocity $30m/s$ in [112] with the same parameters. The states $x = (y, v, \Delta\Psi, r)$ are the lateral displacement y, lateral velocity v, yaw angle $\Delta\Psi$ and yaw rate r; the input is the steering angle δ_f; the disturbance r_d is the road curvature, with $|r_d| \leq 0.05$. The safe set of the system is the set of state-input pairs satisfying

Table 7.2: Computation time of Algs. 5, 6 and 7

Time (s)	2D system	Lane-keeping	Biped
Alg. 5 w.o. Step 4	3.58	3.86	4.42
Alg. 5 w. Step 4	4.13	727.09	8.68
Alg. 6 ($N = n$)	0.58	13.22	0.84
Alg. 6 ($N = 8$)	3.66	227.37	2.41
Alg. 7 ($k_{max} = 50$)	14.29	573.33	188.55



(a) Two-dimensional example (b) Lane-keeping example (c) Biped example

Figure 7.4: The Hausdorff distance d_p (light blue dash curve) in (7.15) and its upper bounds estimated by Alg. 5 (blue and red curves), Alg. 6 (yellow and purple curves), and Alg. 7 (green curve) versus the preview time p.

$|y| \leq 0.9$, $|v| \leq 1.2$, $|\Delta\Psi| \leq 0.05$, $|r| \leq 0.3$ and $|\delta_f| \leq \pi/2$.

The preview on the road curvature r_d is assumed to be available. We select $p_0 = 0$ and run Algs. 5, 6 and 7. Alg. 7 obtains $\overline{p} = 7$, implying that $\pi_{[1,n]}(C_{max,p})$ converges to $C_{max,co}$ at $p = 7$. This observation suggests that the lane-keeping controller gains most from the first 7-step preview. The upper bounds on d_p from the three algorithms are shown in Fig. 7.4b. The upper bound obtained by Alg. 7 is the tightest. The computation time of the three algorithms is listed in the second column of Table 7.2.

7.5.4 Biped walking pattern generation

In [54, 118], preview of the zero-moment point (ZMP) reference is used to control the center of mass (CoM) of biped robots. We consider the discrete-time lateral dynamics of the biped robot in [118], with the same parameters. The states $(x, \dot{x}, \ddot{x})$ are the lateral position x of the CoM and its first and second derivatives; the input is the jerk $\dddot{x}$ of x. The target is to control the lateral position x such that the ZMP $z = x + (h_{CoM}/g)\ddot{x}$ tracks a given reference closely, where h_{CoM} and g are the robot altitude and the acceleration of gravity. The ZMP reference is typically a periodic square signal (see [54, 118]). Here we assume that the reference signal is a trajectory of the uncertain system

$$\overline{z}(t+1) = 0.15\overline{z}(t) + d(t), \tag{7.60}$$

with disturbance $|d| \leq 0.085$. This system can generate periodic signals with maximal magnitude up to 0.1. We couple the biped lateral dynamics and the reference dynamics in (7.60), which leads to a four-dimensional system with state-input pairs satisfying $(x, \dot{x}, \ddot{x}, \overline{z})$. Based on the control target, we select the safe set S_{xu} to be the set of state-input pairs satisfying $|x + (h_{CoM}/g)\ddot{x} - \overline{z}| \leq 0.1$, $|x| \leq 0.1$, $|\dot{x}| \leq 10$, $|\ddot{x}| \leq 10$, $|\overline{z}| \leq 0.1$ and $|\dddot{x}| \leq 100$.

The preview on the disturbance d in (7.60) is available, induced from the preview of ZMP reference [118]. We select $p_0 = 0$. Alg. 7 returns $\overline{p} = 10$, which implies that $\pi_{[1,n]}(C_{max,p})$ converges to $C_{max,co}$ at $p = 10$. This result suggests that the first 10-step preview on the ZMP reference is most useful for maintaining the lateral position of the robot in the safe set. We also depict the upper bounds of d_p obtained by Algs. 5, 6 and 7 in Fig. 7.4c. The conservativeness of those upper bounds are similar to the 2D example in Section 7.5.2. The computation time of the three algorithms is listed in the last column of Table 7.2.

7.5.5 Blade pitch control of wind turbine

Preview of incoming wind events can be measured by Lidar scanners, whose usage has been widely studied in the literature of wind turbine control [89, 111]. In this section, we demonstrate the use of our methods in analyzing the feasible domain of the preview-based constrained MPC in [111] for blade pitch control of a wind turbine. We consider the discrete-time linear wind turbine dynamics in [111] with states $x = (\delta\Omega, \int \delta\Omega, \delta\beta)$, input $\Delta\beta$ and disturbance δv. The states $\delta\Omega$, $\int \delta\Omega$ and $\delta\beta$ are the rotor speed (rad/s) relative to a nominal speed, the integral of $\delta\Omega$ and the pitch angle (deg) relative to a nominal angle; the input $\Delta\beta$ is the increment of $\delta\beta$ in one step; the disturbance δv is the wind speed relative to a nominal wind speed. We use the same parameters in [111]. The safe set S_{xu} is the set of state-input pairs satisfying the state-input constraints $Jx + E\Delta\beta \leq l$ of the MPC proposed in [111] (with J, E and l defined in [111]), and some large enough state bounds $|\delta\Omega| \leq 5$, $|\int \delta\Omega| \leq 100$, and $-4.53 \leq \delta\beta \leq 10.47$ to ensure the compactness of S_{xu}.

The disturbance δv on the wind speed can be previewed [111]. We select C to be the maximal RCIS of the system without preview. Alg. 7 terminates with $\overline{p} = 4$. That is, $\pi_{[1,n]}(\mathscr{F}_p(C))$ converges to $C_{max,co}$ at $p = 4$, which suggests that the MPC benefits most from the first 4-step preview in terms of feasible domain size. We also run Algs. 5 and 6 as in the previous examples. But since Alg. 7 terminates quickly and provides the tightest bound, the results from the other algorithms are omitted.

7.6 Conclusion

In this chapter, we study the impact of different preview time to the safety of discrete-time systems. For general nonlinear system, we derive a novel outer bound on the maximal RCIS $C_{max,p}$. For linear systems under mild conditions, we prove that the safety regret d_p decays to zero exponentially fast, which indicates that the marginal value of the preview information decays to zero exponentially fast with the preview time. We further develop algorithms that compute upper bounds on the safety regret d_p. We also adapt the established theoretical and algorithmic tools to analyze the convergence of the feasible domain of a preview-based robust MPC. The efficiency of the proposed algorithms are verified with four numerical examples.

In this chapter, we assume that we have accurate preview of future disturbances for a finite time horizon. Later in Section 8.3, we relax this assumption and study the safety control problem for systems with uncertain preview (that is, the measurements of future disturbances are noisy). Besides, our current work only deals with a single disturbance with a fixed preview time. As part of the future work, we want to extend our results to systems with multiple disturbances, where each disturbance may have a different preview time.

CHAPTER 8

Scalable Computation of RCISs for Systems with Preview: Three Cases

As shown in Chapter 7, safety control problem for systems with preview is equivalent to the safety control problem for the corresponding augmented systems whose states contain the measurements of the future disturbance (see Section 7.1 for details). Because of the high dimensionality of these augmented systems, synthesizing safety controllers for systems with preview is computationally demanding, and thus a challenging research problem. However, for certain classes of systems with special structures, exploiting structural properties in dynamics can help us reduce the computational cost in controller synthesis. In this chapter, we present three classes of systems where we can efficiently synthesize safety controllers with preview.

The first class is called systems in Brunovsky canonical form. For this class of systems with hyperbox safe sets, the maximal RCIS of the corresponding augmented system can be derived in closed form.

The second class of systems is switched linear systems, where we assume that the mode switching is uncontrolled but can by previewed. We further propose a novel modeling mechanism, called preview automaton, to encode switching signals that can be previewed at run time. The product of the switched system and the preview automaton yields a hybrid system, whose safety control problem is challenging in general. But for this specific hybrid system, we show that its maximal RCIS can be computed efficiently via iterative computations of BRSs of each linear subsystem.

The last class involves systems with uncertain preview, posing a special output-feedback safety control problem. While the general solution to an output-feedback safety control problem involves the maximal RCIS of a set dynamics, by leveraging a nilpotent structure inherent in systems with uncertain preview, we demonstrate that the set dynamics can be reduced to a finite-dimensional system, enabling efficient safety controller synthesis.

Chapter Overview. We first present our results for systems with Brunovsky canonical form in Section 8.1, followed by the results for switched systems equipped with preview automaton in

Section 8.2. In Section 8.3, we present the results for systems with uncertain preview.

8.1 Safety Control for Systems in Brunovsky Canonical Form with Hyperbox Safe Sets

In this section, we study systems in Brunovsky canonical form with a single input[1]. Due to the simple structure of the systems in Brunovsky canonical form, we can derive a closed-form expression of the maximal RCIS within hyperbox safe sets and show some interesting properties of the maximal RCIS based on the closed-form expression. In terms of generality, any controllable system can be converted in a system in Brunovsky canonical form via an invertible transformation (see [11]), and thus our results on systems in Brunovsky canonical form is also useful for controllable systems.

8.1.1 Problem Setup

The dynamics of a system Σ_B in Brunovsky canonical form is

$$\Sigma_B : x(t+1) = \overline{A}x(t) + \overline{B}u(t) + d(t), \tag{8.1}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}$, $d(t) \in D \subseteq \mathbb{R}^n$, and

$$\overline{A} = \begin{bmatrix} \mathbf{0}_{(n-1)\times 1} & \mathbf{I}_{n-1} \\ 0 & \mathbf{0}_{1\times(n-1)} \end{bmatrix}, \overline{B} = \begin{bmatrix} \mathbf{0}_{(n-1)\times 1} \\ 1 \end{bmatrix}. \tag{8.2}$$

The $\mathbf{I}_k$ and $\mathbf{0}_{j\times k}$ in (8.2) represent the identity matrix in $\mathbb{R}^{k\times k}$ and the matrix with all zero entries in $\mathbb{R}^{j\times k}$. Suppose that D is a polytope in $\mathbb{R}^n$, and suppose $B_d = \Pi_{k=1}^n [c_{k,1}, c_{k,2}]$ is the smallest hyperbox containing D. We consider a hyperbox safe set $\mathbf{B} \times \mathbb{R}$, where the state x is constrained within hyperbox $\mathbf{B} = \Pi_{k=1}^n [b_{k,1}, b_{k,2}]$ and the input u is unconstrained.

Recall the definition of the p-augmented system in (7.2). We denote the p-augmented system corresponding to Σ_B by $\Sigma_{B,p}$. The p-augmented safe set is $\mathbf{B} \times D^p \times \mathbb{R}$. We want to solve the following problem in this section.

Problem 8.1. *Compute the maximal RCIS of the p-augmented system $\Sigma_{B,p}$ with respect to p-augmented safe set $\mathbf{B} \times D^p \times \mathbb{R}$.*

[1]The results in this section applies to multiple-input case, since in Brunovsky canonical form, a system with multiple inputs can be decoupled into several systems with single input [11].

8.1.2 Closed-form Expression of the Maximal RCIS for $\Sigma_{B,p}$

In this subsection, we first show several core properties of the maximal RCIS $C_{max,p}$ of $\Sigma_{B,p}$ within $\mathbf{B} \times D^p \times \mathbb{R}$, built on which we derive a closed-form expression of the maximal RCIS .

First, we want to derive a necessary condition under which there exists nonempty RCISs of $\Sigma_{\mathbf{B},p}$ within $\mathbf{B} \times D^p \times \mathbb{R}$. The idea is based on the following observation: given an input $u(t)$ at time $t \geq 0$, due to the special structure of $\overline{A}$ and $\overline{B}$, we have the exact expression of $x_{n-k+1}(t+k)$ for k with $1 \leq k \leq n$ as follows:

$$x_{n-k+1}(t+k) = u(t) + \sum_{i=0}^{k-1} d_{1,n-i}(t+i), \tag{8.3}$$

where $d_{1,n-i}(t+i)$ is the $n-i$ th entry of the vector $d_1(t+i) \in \mathbb{R}^n$ for i from 0 to $n-1$.

Suppose there exists a nonempty RCIS in $\mathbf{B} \times D^p \times \mathbb{R}$. Then there must exist at least one safe input $u(t) \in \mathbb{R}$ such that for all k from 1 to n, the state $x_{n-k+1}(t+k)$ in (8.3) satisfies the constraints provided by $\mathbf{B}$ robust to all possible future disturbances, that is, for k from 1 to n,

$$u(t) + \sum_{i=0}^{k-1} d_{1,n-i}(t+i) \in [b_{n-k+1,1}, b_{n-k+1,2}], \tag{8.4}$$

for all possible values of $\sum_{i=0}^{k-1} d_{1,n-i}(t+i)$; otherwise, for all $u(t) \in \mathbb{R}$, we can find future disturbances such that the state $x_{n-k+1}(t+k) \notin [b_{n-k+1,1}, b_{n-k+1,2}]$.

Note that if $i < p$, $d_{1,n-i}(t+i)$ is a fixed number known from preview at time t; otherwise $d_{1,n-i}(t+i)$ takes arbitrary values in $[c_{n-i,1}, c_{n-i,2}]$. Based on this observation, the condition of the existence of a safe input $u(t)$ satisfying (8.4) is given in Theorem 8.1, which is necessary for the existence of a nonempty controlled invariant set.

Theorem 8.1. *There exists a nonempty RCIS of $\Sigma_{B,p}$ within $\mathbf{B} \times D^p \times \mathbb{R}$ only if the following conditions are satisfied*

$$\forall v \in V_{d,\overline{p}}, \bigcap_{k=1}^{n} \left(\left[\widehat{b}_{k,1}, \widehat{b}_{k,2} \right] - \sum_{i=1}^{\min(n-k+1,\overline{p})} v_{\overline{p}-i+1} \right) \neq \emptyset \tag{8.5}$$

where $V_{d,\overline{p}}$ is the set of vertices of the hyperbox $\mathbf{B}_{d,\overline{p}} = \Pi_{k=n-\overline{p}+1}^{n} [c_{k,1}, c_{k,2}]$, and $\overline{p} = \min(p,n)$, and $\widehat{b}_{k,1} = b_{k,1} - \sum_{i=k}^{n-\overline{p}} c_{i,1}$ and $\widehat{b}_{k,2} = b_{k,2} - \sum_{i=k}^{n-\overline{p}} c_{i,2}$ for k with $n-\overline{p} \leq k \leq n$.

In practice, if we want to compute a RCIS of $\Sigma_{B,p}$, unnecessary computation can be avoided by checking the conditions provided by Theorem 8.1 first. However, the cardinality of $V_{d,\overline{p}}$ can be large, which prevents us from checking (8.5) efficiently. Fortunately, with some manipulation, the condition in (8.5) can be simplified to n^2 inequalities, independent of the cardinality of $V_{d,\overline{p}}$, namely

that for all j and k from 1 to n,

$$\begin{aligned}
&\forall j = k,\ b_{j,1} - b_{k,2} \leq \sum_{i=k}^{n-p} (c_{i,1} - c_{i,2}), \\
&\forall j < k,\ b_{j,1} - b_{k,2} \leq \sum_{i=j}^{n-p} c_{i,1} - \sum_{i=k}^{n-p} c_{i,2} + \sum_{i=\max(j,n-p+1)}^{\max(k-1,n-p)} c_{i,1}, \\
&\forall j > k,\ b_{j,1} - b_{k,2} \leq \sum_{i=j}^{n-p} c_{i,1} - \sum_{i=k}^{n-p} c_{i,2} - \sum_{i=\max(k,n-p+1)}^{\max(j-1,n-p)} c_{i,2}.
\end{aligned} \tag{8.6}$$

Next, suppose that there exists a nonempty RCIS , namely that (8.5) is satisfied. We want to derive conditions under which states $\xi = (x, d_1, d_2, ..., d_p) \in \mathbb{R}^{(p+1)n}$ are contained by the maximal RCIS .

We use x_i, $d_{k,i}$ to denote i th entry of x, d_k. According to the dynamics in (8.1), the first $(n-t)$ entries of the vector $x(t)$ for all $t = 0, 1, \cdots, n-1$ are independent from the control inputs and completely determined by the initial state $x(0)$ and disturbances $d(0)$, $d(1)$ ..., $d(n-2)$.

Thus, one necessary condition on $\xi(0) = (x(0), d_{1:p}(0)) \in C_{max,p}$ is that for all possible future disturbances in D that are not previewed yet at the initial time, for all t from 0 to $n-1$ and all k from 1 to $n-t$, the state $x(t)$ satisfies

$$x_k(t) \in [b_{k,1}, b_{k,2}]. \tag{8.7}$$

By expanding $x_k(t)$ using $x(0)$ and $d_{1:p}(0)$, we obtain the conditions stated in the following theorem.

Theorem 8.2. *A state $(x, d_{1:p})$ is contained in the maximal RCIS $C_{max,p}$ only if*

$$x \in \mathbf{B}, d_{1:p} \in D^p, \tag{8.8}$$

and for all k, $2 \leq k \leq n$ and for all j, $1 \leq j < k$:

$$x_k + \sum_{i=1}^{\min(k-j,p)} d_{i,k-i} \in [b_{j,1}, b_{j,2}] - \sum_{i=p+1}^{k-j} [c_{k-i,1}, c_{k-i,2}]. \tag{8.9}$$

where $d_{i,k-i}$ is the $k-i$ th entry of vector d_i.

To make notation clear, in the case of $k - j < p + 1$, the right hand set of (8.9) becomes $[b_{j,1}, b_{j,2}] - \emptyset = [b_{j,1}, b_{j,2}]$. We denote the set of states $(x, d_{1:p})$ satisfying constraints in (8.8) and (8.9) by C_p. The following theorem states that the maximal RCIS of $\Sigma_{B,p}$ within $\mathbf{B} \times D^p \times \mathbb{R}$ is exactly equal to C_p.

Theorem 8.3. *Suppose that* (8.5) *is satisfied. Define*

$$C_p = \{\xi = (x, d_{1:p}) \mid \xi \text{ satisfies (8.8), (8.9)}\}. \tag{8.10}$$

Then, C_p is the maximal RCIS of $\Sigma_{B,p}$ within the safe set $\mathbf{B} \times D^p \times \mathbb{R}$.

Corollary 8.3.1. The condition in (8.5) is the necessary and sufficient condition for the existence of nonempty RCISs of $\Sigma_{B,p}$ within $\mathbf{B} \times D^p \times \mathbb{R}$.

Corollary 8.3.2. If instead of Σ_B in (8.1), we consider a system in the following form:

$$\Sigma_v : x(t+1) = \overline{A}x(t) + \overline{B}u(t) + \overline{E}d(t), \tag{8.11}$$

for $d(t) \in D_v \subseteq \mathbb{R}^l$ and some $\overline{E} \in \mathbb{R}^{n \times l}$. Then, we first define system Σ'_B in Brunovsky canonical form

$$\Sigma'_B : x(t+1) = \overline{A}x(t) + \overline{B}u(t) + d(t), \tag{8.12}$$

with $d(t) \in \overline{E}D_v \subseteq \mathbb{R}^n$. We have the closed-form expression of the maximal RCIS C_p of the p-augmented system of Σ'_B within $\mathbf{B} \times D^p \times \mathbb{R}$. The maximal RCIS C_v of the p-augmented system of Σ_v within $\mathbf{B} \times D^p \times \mathbb{R}$ is nonempty if and only if C_p is nonempty and

$$C_v = \{(x, d_{1:p}) \mid (x, \overline{E}d_1, \overline{E}d_2, \cdots, \overline{E}d_p) \in C_p\}. \tag{8.13}$$

Remark 8.1. Adopting the idea from [8], if we have a more general safe set in form of $P \times \mathbb{R}$, where P is a polytope, we can construct a RCIS of $\Sigma_{B,p}$ within $P \times D^p \times \mathbb{R}$ in 2 moves: we first construct a polytope in a lifted space that encodes all hyperboxes $\mathbf{B}$ in P and all states $(x, d_{1:p})$ within the maximal RCIS within $\mathbf{B} \times D^p \times \mathbb{R}$ based on the nonemptyness condition (8.5) and the closed-form expression of C_p. Then, we project this lifted set onto its first $n(p+1)$ coordinates, which is equal to the union of the maximal RCIS within $\mathbf{B} \times D^p \times \mathbb{R}$ for all hyperboxes $\mathbf{B}$ contained by P. By construction, this set is a RCIS in $P \times D^p \times \mathbb{R}$.

Furthermore, as pointed out by Remark 1 in [9], any controllable system with a polytopic safe set (including input constraints) can be transformed into system in Brunovsky canonical form with a safe set in form of $P \times \mathbb{R}$. Thus, our results in this section can be used to compute controlled invariant sets for p-augmented systems of a controllable system. ■

Directly obtained from the closed-form expression of the maximal RCIS C_p, an interesting property of C_p for $p \geq n$ is revealed by the following theorem.

Theorem 8.4. *For preview time $p > n$, the maximal RCIS C_p is equal to the cartesian product of the maximal RCIS C_n of $\Sigma_{B,n}$ and the set D^{p-n}, that is $C_p = C_n \times D^{p-n}$.*

Theorem 8.4 indicates that for system Σ_B in Brunovsky canonical form with a safe set $\mathbf{B} \times \mathbb{R}$, the preview time longer than $p = n$ is not necessary. However, given a state $(x, d_{1:p})$ in the maximal RCIS $C_n \times D^{p-n}$, the admissible input set with the maximal size is obtained when preview is $n+1$, that is

$$\mathscr{A}(C_n, (x, d_{1:n})) \subseteq \mathscr{A}(C_{n+1}, (x, d_{1:n+1})) = \mathscr{A}(C_p, (x, d_{1:p})).$$

Finally, recall that we define safety regret as the Hausdorff distance between the maximal CIS of the disturbance-collaborative system and the projection of the maximal RCIS of the p-augmented system in Section 7.3. We wonder how the safety regret varies with preview time p for this class of systems.

Theorem 8.5. *For preview time $p > n$, if nonemptyness condition* (8.5) *holds, then the projection of C_p onto the first n coordinates is equal to the maximal RCIS $C_{max,co}$ of the disturbance-collaborative system $\mathscr{D}(\Sigma_B)$ within safe set $\mathbf{B} \times \mathbb{R}$, that is*

$$C_{max,co} = \pi_{[1,n]}(C_p) = \pi_{[1,n]}(C_n).$$

By Theorem 8.5, the safe regret for systems in Brunovsky canonical form with hyperbox safe sets always converges to zero in n steps.

8.1.3 Illustrative Examples

In this section, we show the effect of preview information on safety control via concrete examples.

8.1.3.1 Impact of Preview on Disturbance Tolerance

First, we want to demonstrate the impact of preview on disturbance tolerance via our results on systems in Brunovksy canonical form. We fix the state dimension $n = 10$ and the safe set $\mathbf{B} = \Pi_{i=1}^{n}[-1, 1]$. Then, we parametrize the disturbance set $D = \Pi_{i=1}^{n}[-c, c]$ by a positive number $c > 0$. We are interested in the largest c we can have such that the augmented system $\Sigma_{B,p}$ has nonempty RCISs within $\mathbf{B} \times D^p \times \mathbb{R}$. According to Corollary 8.3.1, we can utilize the condition on nonempty RCIS given by (8.6) to determine the largest possible c.

By plugging $b_{k,1} = -1$, $b_{k,2} = 1$, $c_{k,1} = -c$ and $c_{k,2} = c$ for all k from 1 to n into (8.6) , we can obtain an upper bound on c such that the nonemptyness condition in (8.6) holds. The largest c computed for different preview time p are shown in Fig. 8.1. As we expect, as the preview time increases, a larger disturbance set can be handled, due to the power of preview.

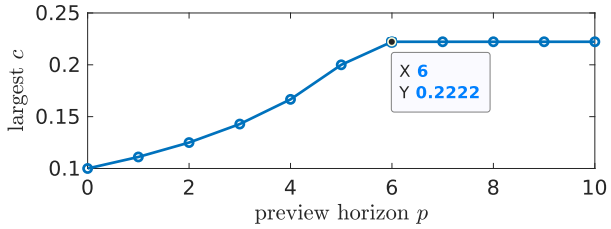


Figure 8.1: The largest disturbance bound c versus preview time p for the system in Brunovsky canonical form ($n = 10$) with hyperbox safe set.

In addition, we observe in Fig. 8.1 that the largest c stops increasing after $p \geq 6$. This observation suggests that a disturbance set with $c > 0.2222$ may lead to an empty RCIS for any preview time p. With some calculation, we can verify that for $c > 2/9$, the necessary condition (8.6) does not hold for all $p \geq 0$ and thus the maximal RCIS is always empty no matter how large the p is.

8.1.3.2 Lane Keeping Control with Preview

To show the usefulness of preview, we are going to present how preview helps the driver-assist system to keep a vehicle within lanes. We use a 4-dimensional linearized bicycle model with respect to constant longitudinal speed $30m/s$ from [112]. The state space consists of lateral displacement y, lateral velocity v, yaw angle $\Delta\Psi$ and yaw rate r. The disturbance r_d with $|r_d| \leq 0.04$ considered in this simplified model is a quantity related to the road curvature that perturbs the yaw angle. The control input u is the steering angle, with constraints $u \in [-\pi/2, \pi/2]$.

The safe set S_{xu} is the set of state-input pairs within bounds $|y| \leq 0.9$, $|v| \leq 1.2$, $|\Delta\Phi| \leq 0.05$ and $|r| \leq 0.3$, and $|u| \leq \pi/2$. We set the preview time $p = 5$. We first compute the maximal RCIS within S_{xu} for system without preview, denoted by $C_{max,0}$. Then, we use the inside-out algorithm (Alg. 2) to grow the seed set $C_{max,0} \times D^5$ for the p-augmented system over 10 iterations, the result of which is denoted by $C_{io,5}$. Numerically we find that $C_{io,5}$ strictly contains $C_{max,0} \times D^5$. We also try the idea in Remark 8.1 to obtain a RCIS based on our results in Section 8.1, but the resulting set is contained by $C_{max,0} \times D^p$, which is too conservative to be useful.

Next, we find a point $(x_0, d_1, \cdots, d_5)$ belonging to the set difference $C_{io,5} \setminus C_{max,0} \times D^5$ and simulate 2 trajectories starting at x_0 with the first 5 disturbances $d_{1:5}$, using the two RCISs $C_{max,0}$ and $C_{io,5}$ respectively. The controller consists of two parts: First, we synthesize a nominal state feedback controller, by solving the LQR problem for the p-augmented system. Then, at each time instant, we supervise the control input from the nominal controller by projecting that input onto the admissible input set at current state with respect to $C_{max,0}$ or $C_{io,5}$. If the admissible input set happens to be empty at some time instants, then we project the nominal input onto the input constraint set $[-\pi/2, \pi/2]$. The resulting vehicle maneuvers are shown by Fig. 8.2, where we find

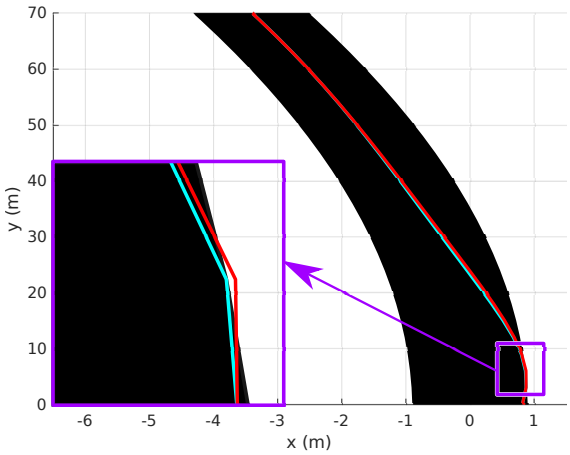


Figure 8.2: The vehicle maneuvers under the linearized bicycle model with supervised LQR controller. The dark region indicates the safe region in the plane, that is the lane. The cyan curve is the maneuver corresponding to $C_{oi,5}$. The red curve is the maneuver corresponding to $C_{max,0}$.

that the trajectory under the supervision of the admissible input set with respect to $C_{io,5}$ stays within the lane as required by the safety constraints during the simulation time span, but the trajectory under the supervision with respect to $C_{max,0}$ violates the constraints on lateral displacement y and drives out of the lane at the 2nd time step. This observation meets our expectation since we intentionally pick an initial state that is not in $C_{max,0}$. This example demonstrates how the preview on future disturbances enables controllers to deal with a larger set of initial states safely.

8.1.4 Summary

In this section, we study systems in Brunovsky canonical form with hyperbox safe sets, for which we derive the maximal RCIS of the p-augmented system in closed form. The closed-form expression enables us to directly see the impact of preview on the RCISs and lead us to some interesting properties for this special systems and safe sets, such as a finite-step convergence of the safety regret.

8.2 Safety Control with Preview Automaton

In this section, we consider discrete-time switched systems where the mode signal is controlled by external factors. We assume the system is equipped with sensors that provide preview information on the mode signal (i.e., some future values of the mode signal can be sensed/predicted at run-time). To capture how the mode signal evolves and how it is sensed/predicted at run-time, we introduce *preview automaton*. Then, we focus on safety specifications defined in terms of a safe set within

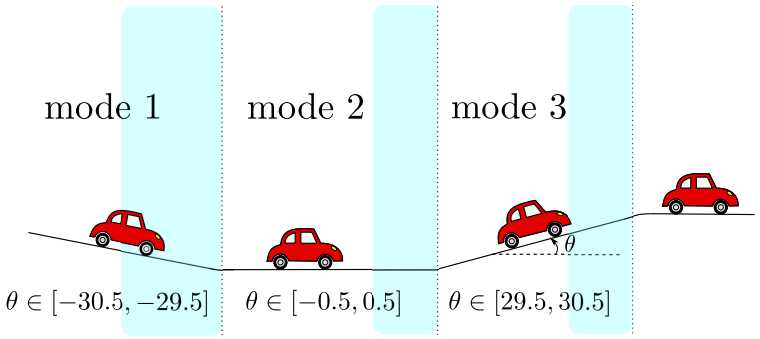


Figure 8.3: A simple example on autonomous vehicle cruise control. The road grade alternates between three ranges r_1, r_2, r_3, modeled by a switched system Σ with three modes. The blue shadows indicate the regions where the vehicle's sensors are able to look ahead and upon the detection of the upcoming change, a preview input is released.

each mode, and develop an algorithm that computes the maximal invariant set inside these safe sets while incorporating the preview information. A simple example where such information can be relevant is depicted in Fig. 8.3, where an autonomous vehicle can use its forward looking sensors or GPS and map information to predict when the road grade will change. The proposed framework provides a means to leverage such information to compute provably-safe controllers that are less conservative compared to their preview agnostic counterparts.

Our work is related to [62, 49, 123] where synthesis from linear temporal logic or general omega-regular specifications are considered for discrete-state systems. In [62], it is assumed that a fixed horizon lookahead is available; whereas, in our work, the preview or lookahead time is non-deterministic, and preview automaton can be composed with both discrete-state and continuous-state systems. While we restrict our attention to safety control synthesis, extensions to other logic specifications are also possible. Another main difference with [62] is that we use the preview automaton also to capture constraints on mode switching. The idea of using automata or temporal logics to capture assumptions on mode switching is used in [15] and [93]. In particular, the structure of the RCISs we compute is similar to the invariant sets (for systems without control) in [15]. However, neither [15] nor [93] takes into account preview information.

The remainder of this section is organized as follows. After briefly introducing the basic notations next, in Section 8.2.1, we describe the problem setup, define the preview automaton and formally state the safety control problem. An algorithm to solve the safety control problem with preview is proposed and analyzed in Section 8.2.2. In Section 8.2.3, we demonstrate the proposed algorithm by two case studies one on vehicle cruise control and another on lane keeping before we conclude this section in Section 8.2.4.

Notation. The symbol $\overline{\mathbb{N}}$ denotes the set $\mathbb{N} \cup \{\infty\}$ of extended natural numbers. Given a set X, the power set of X is denoted by 2^X.

8.2.1 Problem Setup

We consider switched systems Σ of the form:

$$x(t+1) \in f_{\sigma(t)}(x(t), u(t)), \tag{8.14}$$

where $\sigma(t) \in \{1, \ldots, s\}$ is the mode of the system, $x(t) \in X$ is the state and $u(t) \in U$ is the control input. We assume that the switching is uncontrolled (i.e., the mode $\sigma(t)$ is determined by the external environment) however $\sigma(t)$ is known when choosing $u(t)$ at time t. By defining each $f_i : X \times U \to 2^X$ to be set-valued, we capture potential disturbances and uncertainties in the system dynamics that are not directly measured at run-time but that affect the system's evolution.

We are particularly interested in scenarios where some *preview* information about the mode signal is available at run-time. That is, the system has the ability to lookahead and get notified of the value of mode signal before the mode signal switches value. More specifically, we assume that for each pair of modes (i, j), if the switching from i to j takes place next, a sensor can detect this switching for τ_{ij} time steps ahead of the switching time, where $\tau_{ij} \geq 0$ is called *the preview time* and belongs to a time interval T_{ij}. Mathematically, if $\sigma(t + \tau_{ij} - 1) = i$ and $\sigma(t + \tau_{ij}) = j$, then the value $\sigma(t + \tau_{ij})$ is available before choosing $u(t)$ at time t, for some $\tau_{ij} \in T_{ij}$.

In many applications, switching is not arbitrarily fast. That is, there is a minimal holding time (or, dwell time) between two consecutive switches. For each mode i of the switched system, we associate a *least holding time* $H_i \geq 1$ such that if the system switches to mode i at time t, the environment cannot switch to another mode at any time between t and $t + H_i - 1$. Note that $H_i = 1$ for all i is the trivial case where the system does not have any constraints on the least holding time. Moreover, there could be constraints on what modes can switch to what other modes.

Following example illustrates some of the concepts above.

Example 8.1. In Figure 8.3, a vehicle runs on a highway where the road grade can switch between ranges $r_1 = [-30.5, -29.5]$ and $r_2 = [-0.5, 0.5]$ and between r_2 and $r_3 = [29.5, 30.5]$ (no direct switching between r_1 and r_3). We use a switched system Σ of 3 modes to model the three ranges r_1, r_2 and r_3. Thanks to the perception system on the vehicle, the switching from mode i to mode j can be detected $\tau_{ij} \in T_{ij}$ steps ahead, where T_{ij} is a known interval of feasible preview times for $(i, j) \in \{(1,2), (2,1), (2,3), (3,2)\}$. Also, the least time steps for the vehicle in range r_i is H_i for $i = 1, 2, 3$. □

For simplicity, in the rest of the section, we will assume that the least holding time is greater

than or equal to the least feasible preview time among all modes that the system can switched to from mode i, i.e., $H_i \geq \min(\cup_j T_{ij})$ for any mode i. This assumption is justified in many applications where switching is "slow" compared to the worst-case sensor range. For instance, the road curvature or road grade does not change too frequently.

The main contribution of this section is two folds:

- to provide a new modeling mechanism for switched systems that can capture both the constraints on the switching and the preview information,
- to develop algorithms that can compute controllers to guarantee safety with preview information in a way that is less conservative compared to their preview agnostic counterparts.

8.2.1.1 Preview Automaton

Provided the prior knowledge on the preview time interval T_{ij} and the least holding time H_i, we model the allowable switching sequences of a switched system with preview with a mathematical construct we call *preview automaton*.

Definition 8.1. (**Preview Automaton**) A *preview automaton* G corresponding to a switched system Σ with s modes is a tuple $G = \{Q, E, T, H\}$, where

- $Q = \{1, 2, \cdots, s\}$ is a set of nodes (discrete states), where node $q \in Q$ corresponds to the mode q in Σ;
- $E \subseteq Q \times Q$ is a set of transitions;
- $T : E \to \{[t_1, t_2] : 0 \leq t_1 \leq t_2, t_1 \in \mathbb{N}, t_2 \in \overline{\mathbb{N}}\}$ labels each transition with the time interval of possible preview times corresponding to that transition;
- $H : Q \to (\mathbb{N} \backslash \{0\}) \cup \{\infty\}$ labels each node $q \in Q$ with the least holding time corresponding to that node.

We make a few remarks. First, we do not allow any self-loops, i.e., $(q, q) \notin E$ for all $q \in Q$. Second, the preview times $T(q_1, q_2)$ for any $(q_1, q_2) \in E$ is in one of the three forms: a singleton set $\{t_1\}$ (interval $[t_1, t_1]$), or a finite interval $[t_1, t_2]$ with $t_1 < t_2$, or an infinite interval $[t_1, \infty)$. Finally, if there is a state q with no outgoing edges, that is $\{(q, p) \in E : p \in Q\} = \emptyset$, we set $H(q) = \infty$ to indicate that once the system visits q, it remains in q indefinitely, so the deadlocks are not allowed. We call such a state a sink state. The set of sink states and the set of non-sink states in Q are denoted by Q_s and Q_{ns}.

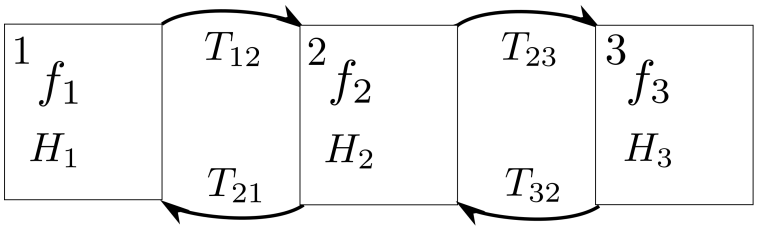


Figure 8.4: This preview automaton corresponds to the switched system in Example 8.1.

In Definition 8.1, the nodes of the preview automaton are chosen to be the modes of the switched system for simplicity. It is easy to extend the definition to allow multiple nodes in the preview automaton to correspond to the same mode. Alternatively, redefining the switched system by replicating certain modes and keeping the current definition can serve the same purpose.

Example 8.2. The preview automaton for the switched system in Example 8.1 has nodes $Q = \{1,2,3\}$ with transitions shown in Figure 8.4. The least holding mapping $H(q) = H_q$ for all $q \in Q$ and $T(q_1,q_2) = T_{q_1q_2}$ for $(q_1,q_2) \in \{(1,2),(2,1),(2,3),(3,2)\}$. □

Any transition in the preview automaton is associated with an input in the form of the preview of the switching mode. We assume that there is at most one preview between any two consecutive switches. During the execution of the preview automaton, if a preview takes place at time t, there is a corresponding *preview input* of the preview automaton, including the timestamp t of the occurrence of the preview, the destination state $d \in Q$ of the next transition and the remaining time steps (the preview time) τ from the current time t up to the next transition. If no preview takes place before the next switching time[2], the preview input corresponding to that switch is trivially $(t,0,q)$, where t and q are the time instant and destination of the next transition. Note that $t+\tau$ is the time that the system transits from the last mode to the mode d.

Definition 8.2. Given preview automaton $G = \{Q,E,T,H\}$ and initial state $q_0 \in Q$, a sequence of tuples $\{(t_k,\tau_k,d_k)\}_{k=1}^{N}$ ($N < \infty$ when the system remains in d_N after $t \geq t_N + \tau_N$) is a valid *preview input sequence* of G if for all $1 \leq k \leq N$, the sequence satisfies (with $t_0 = 0$, $\tau_0 = 0$, $d_0 = q_0$) that
(1) $\tau_{k-1} \geq 0$ and $t_{k-1} + \tau_{k-1} \leq t_k$ and
(2) $(d_{k-1},d_k) \in E$ and $\tau_k \in T(d_{k-1},d_k)$ and
(3) $(t_k + \tau_k - 1) - (t_{k-1} + \tau_{k-1}) \geq H(d_{k-1})$.

In above definition, conditions (1), (2) and (3) guarantee that only one preview input is received between two consecutive switches, the mode switch constraints and preview time constraints are met, and the holding time constraint is met, respectively. Once a valid input sequence is given, we can uniquely identify the transitions of the preview automaton over time, that is, the execution of the

[2]This is possible when the lower bound of the time interval of possible preview times is 0.

preview automaton with respect to that input sequence. In the rest of the section, we only consider valid preview input sequences and drop the word valid when it is clear from the context.

Definition 8.3. Given preview automaton $G = \{Q, E, T, H\}$ and a preview input sequence $\{t_k, \tau_k, d_k\}_{k=1}^{N}$, the *execution* of G with respect to the preview input sequence is a sequence of tuples $\{(I_k, q_k)\}_{k=0}^{N}$, where
(1) $I_0 = [0, t_1 + \tau_1 - 1]$ and $I_k = [t_k + \tau_k, t_{k+1} + \tau_{k+1} - 1]$ for all $k \geq 0$,
(2) $q_k = d_k$ for all $k \geq 1$.

Note that two different valid preview input sequence may have the same execution. According to Definition 8.2 and 8.3, the set of possible executions of one preview automaton is determined by the set of valid preview input sequences of the preview automaton.

Once we have the preview automaton G corresponding to a switched system Σ, we have a model of the allowable switching sequences for Σ, given by the executions of G. Therefore we can define the runs of a switched system with respect to a preview automaton.

Definition 8.4. A sequence $\{(q(t), x(t))\}_{t=0}^{\infty}$ is a *run of the switched system* Σ of s modes with preview automaton G under the control inputs $\{u(t)\}_{t=0}^{\infty}$ if (1) $\{q(t)\}_{t=0}^{\infty}$ is an execution of G for some preview input sequence and (2) $x(t+1) \in f_{q(t)}(x(t), u(t))$ for $t \geq 0$.

8.2.1.2 Problem Statement

Though the preview automaton can be useful in the existence of more general specification, we focus only on safety specifications here. Suppose that each mode k is associated with a *safe set* $S_k \subseteq X$, that is the set of states where we require the switched system to stay within when the system's active mode is k. Denote the collection of safe sets $\{S_i\}_{i \in Q}$ for each mode as a *safety specification* for the switched system Σ. Then, given a safety specification $\{S_i\}_{i \in Q}$, a run $\{(q(t), x(t))\}_{t=0}^{\infty}$ is *safe* if $(q(t), x(t)) \in \cup_{i \in Q}(i, S_i)$ for all $t \geq 0$. Otherwise, this run is *unsafe*.

A controller is usually assumed to know the partial run of the system up to the current time before making a control decision at each time instant. In our case, since the system can look ahead and see the next transition, reflected by the preview input signal, the controller for a switched system equipped with a preview automaton is assumed to have access to the preview inputs of the preview automaton up to the current time.

Definition 8.5. Denote $\{(p(t), x(t))\}_{t=0}^{t^*}$ and $\{(t_k, \tau_k, d_k)\}_{k=0}^{k^*}$ as the partial run and preview inputs of the switched system up to time t^* (k^* refers to the latest preview up to t^*, i.e., $k^* = \max k$ s.t. $t_k \leq t^*$). A *controller* $\mathscr{U}$ of the switched system Σ with preview automaton G is a function that maps the partial run $\{(p(t), x(t))\}_{t=0}^{t^*}$ and the preview inputs $\{(t_k, \tau_k, d_k)\}_{k=0}^{k^*}$ to a control input $u(t^*)$ of the switched system for any $t^* \geq 0$.

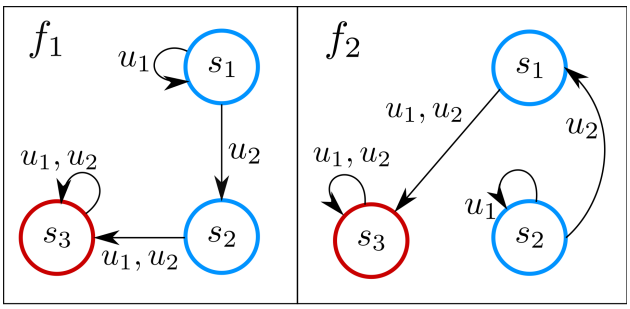


Figure 8.5: A switched finite transition system with 2 modes f_1 and f_2. The safe set (blue) is $\{s_1, s_2\}$ for each mode.

Definition 8.6. Given a switched system Σ and a safety specification $\{S_i\}_{i\in Q}$, a subset W_i of the state space of Σ is a single *winning set with respect to mode i* if there exists a controller $\mathscr{U}$ such that any run of the closed-loop switched system with initial state in $\{i\} \times W_i$ is safe. A winning set W_i is the *maximal winning set with respect to the mode i* if for any $x \notin W_i$, for any controller $\mathscr{U}$, there exists an unsafe run with initial state (i,x). A *winning set with respect to the switched system* Σ is the collection $\{W_i\}_{i=1}^n$ of single winning sets for all modes, which is called *winning set* for short.

We note that, by definition, arbitrary unions of winning sets with respect to one mode is still a winning set, and therefore the maximal winning set is unique under mild conditions [20] and contains all the winning sets with respect to that mode. Now, we are ready to state the problem of interest.

Problem 8.2. *Given a switched system* Σ *with corresponding preview automaton and safety specification* $\{S_i\}_{i\in Q}$*, find the maximal winning set* $\{W_i\}_{i\in Q}$.

Before an algorithm that computes the maximal winning set is introduced, we first study the following toy example to demonstrate the usefulness of preview information.

Example 8.3. A switched transition system with two modes is shown in Figure 8.5. The state space and input space of the switched system are $\{s_1, s_2, s_3\}$ and $\{u_1, u_2\}$ respectively. The safety specification is $S_1 = S_2 = \{s_1, s_2\}$. To satisfy this safety specification, the system state has to be s_1 when f_1 is active and be s_2 when f_2 is active. Thus by inspection, when there is no preview, the winning sets are empty and when there is a preview for at least one-step ahead before each transition, there is a non-empty winning set $W_1 = \{s_1\}$ and $W_2 = \{s_2\}$. □

Example 8.3 suggests that a winning set is not the same as a RCIS of the switched system. When the preview information is ignored or unavailable, they are the same and therefore Problem 8.2 can be solved by computing the RCISs within the safe sets. However, if the preview is available, the RCISs can be conservative since their computation does not take advantage of the online

preview information. Therefore, in Example 8.3, the maximal RCIS is empty, but the winning set is non-empty.

8.2.2 Maximal Winning Set Computation with Preview Information

In this section, we propose an algorithm to solve Problem 8.2. Recall that in Definition 8.1, the feasible preview time interval given by T can be unbounded from the right, which is difficult to deal with in general because it essentially corresponds to a potentially unbounded clock. However, the following theorem reveals an important property of the preview automaton, which allows us to replace the preview time interval with its lower bound in the computation of winning sets.

Theorem 8.6. *Let $G = \{Q, E, T, H\}$ and $\widehat{G} = \{Q, E, \widehat{T}, H\}$ be two preview automata of the switched system Σ with s modes, where $\widehat{T}(q_1, q_2) = \min(T(q_1, q_2))$ for any $(q_1, q_2) \in E$. Then given a safety specification $\{S_i\}_{i\in Q}$, $\{W_i\}_{i\in Q}$ is a winning set with respect to G if and only if $\{W_i\}_{i\in Q}$ is a winning set with respect to $\widehat{G}$.*

Proof. Note that G and $\widehat{G}$ are the same except the feasible preview time interval T. For each transition $(p_1, p_2) \in E$, $\widehat{T}(q_1, q_2)$ is equal to the lower bound of $T(q_1, q_2)$.

To show the "only if" direction, suppose that W_i is a winning set with respect to G for mode i. Then by Definition 8.6, there exists a controller $\mathscr{U}$ such that any run of the closed-loop system with initial state in $\{i\} \times W_i$ with respect to any valid preview input sequence of G is safe. By Definition 8.2 and 8.3, any preview input sequence and the corresponding execution of $\widehat{G}$ are also valid preview inputs and execution of G. Therefore, using the same controller U, any run of the closed-loop system with initial state in $\{i\} \times W_i$ with respect to any valid preview input sequence of $\widehat{G}$ is safe. Hence we conclude that each W_i, for $i \in Q$, is a winning set with respect to $\widehat{G}$ for mode i.

To show the "if" direction, suppose that W_i is a winning set with respect to $\widehat{G}$ for mode i, and $\mathscr{U}$ is a controller such that any run of the closed-loop system with initial state in $\{i\} \times W_i$ with respect to any valid preview input sequence for $\widehat{G}$ is safe. Note that any execution of G is also an execution of $\widehat{G}$, but the corresponding preview input sequences of G and $\widehat{G}$ can be different. Suppose that $\{(t_k, \tau_k, q_k)\}_{k=1}^N$ and $\{(t'_k, \tau'_k, q_k)\}_{k=1}^N$ are two preview input sequences of G and $\widehat{G}$ corresponding to the same execution $\pi = \{(I_k, q_k)\}_{k=0}^N$. Then by Definition 8.2, for all $k \geq 1$, $\tau'_k = \widehat{T}(q_k, q_{k+1}) = \min(T(q_k, q_{k+1}))$ and $\tau_k \in T(q_k, q_{k+1})$ and $t'_k + \tau'_k = t_k + \tau_k = \min(I_{k+1})$. Hence $\tau'_k \leq \tau_k$ and $t'_k \geq t_k$ for all $k \geq 1$, which implies that for any transition in π, a controller of G always knows the next mode from the preview input of G earlier than a controller of $\widehat{G}$.

Since the preview input of the next transition is earlier in G than in $\widehat{G}$, given an execution $\pi = \{(I_k, q_k)\}_{k=0}^N$, for any $k \geq 1$, $\mathscr{U}$ can always infer[3] the k^{th} preview input of $\widehat{G}$ from the k^{th} input

[3]Given the k^{th} preview input (t_k, τ_k, q_{k+1}) of G, the k^{th} preview input of $\widehat{G}$ is $(t_k + \tau_k - \tau'_k, \tau'_k, q_{k+1})$ with $\tau'_k = \widehat{T}(q_k, q_{k+1})$.

of G before time t'_k. We force controller $\mathscr{U}$ to generate control inputs for the switched system Σ with preview automaton G based on the inferred inputs of $\widehat{G}$. Then any run of Σ when closing the loop with the customized $\mathscr{U}$ and G is a run of the closed-loop system with respect to Σ, $\mathscr{U}$ and $\widehat{G}$, which is safe if the initial state is in $\{i\} \times W_i$. Hence W_i is a winning set of G for all $i \in Q$. □

Thanks to Theorem 8.6, in terms of maximal winning set computation, it is enough to consider the preview automaton whose preview time interval is a singleton set for all transitions without introducing any conservatism. This property stated in Theorem 8.6 can reduce computation cost and simplify the algorithms. Therefore, whenever there is a preview automaton G, we first convert G into the form of $\widehat{G}$ in Theorem 8.6. Algorithm 8 is designed to compute the maximal winning set for the preview automaton in the form of $\widehat{G}$, whose result is equal to the maximal winning set of G. We note that $\widehat{G}$ can be expanded to a non-deterministic finite transition system with $\sum_i (H_i - (\min_j T_{i,j}) + \sum_j T_{i,j})$ states. Taking a product of this finite transition system with the switched system, the problem can be reduced to an invariance computation (with measurable and unmeasurable non-determinism) on the product system. However, the algorithms we propose avoid product construction and directly define fixed-point operations on the switched system's state space.

In Algorithm 8, lines 4-6 compute the maximal winning set for each sink state in G. Lines 7-12 compute the winning sets of the non-sink states iteratively, with updates given by Algorithm 9. The main operators used in these algorithms are as follows. First, given a mode i of the switched system with state space X and action space U, and a subset V of X, the *one-step controlled predecessor* of V with respect to the dynamics f_i is defined as

$$Pre^{f_i}(V) = \{x \in X : \exists u \in U, f_i(x,u) \subseteq V\}, \tag{8.15}$$

that is the set of states that can be guaranteed to reach the set V in one time step by some control inputs in U.

Second, given a safe set $S_i \subset X$, the *one-step constrained controlled predecessors* $PreInt(\cdot)$ of an arbitrary set V with respect to the dynamics f_i as

$$PreInt^{f_i}(V, S_i) = Pre^{f_i}(V) \cap S_i. \tag{8.16}$$

Now define $V_0 = S_i$ and update V_k recursively for $k \geq 0$ by

$$V_{k+1} = PreInt^{f_i}(V_k, S_i). \tag{8.17}$$

Note that $\{V_k\}_{k=0}^{\infty}$ in (8.17) is monotonically non-increasing sequence of sets and the fixed point (reached when $V_{k+1} = V_k$) is *the maximal RCIS* within the safe set S_i with respect to the dynamics f_i, denoted as $Inv^{f_i}(S_i)$. Finally, given the preview automaton $G = \{Q, E, T, H\}$, the successors of

some node $i \in Q$ is defined as $Post^G(i) = \{j : (i,j) \in E\}$.

Algorithm 8 Winning Set for Problem 1

function $ConInv(\Sigma, S = \{S_i\}_{i\in Q}, G)$
  initialize $\{W_i\}_{i\in Q}$ with $W_i = S_i, \forall i \in Q$.
  initialize $\{V_i\}_{i\in Q}$ with $V_i = \emptyset$.
  for $i \in Q$ such that $H(i) = \infty$ (sink states) **do**
    $W_i \leftarrow Inv^{f_i}(S_i)$
  end for
  while $\exists i \in Q$ such that $W_i \neq V_i$ **do**
    $V_i \leftarrow W_i, \forall i \in Q$
    for $i \in Q$ such that $H(i) < \infty$ **do**
      $W_i \leftarrow InvPre^{f_i}(G, \{W_j\}_{j\in Post(i)}, S)$
    end for
  end while
  return $\{W_i\}_{i\in Q}$
end function

Some properties of these operators and the *InvPre* operator defined by Algorithm 9 are analyzed next. These properties are used later to prove the correctness of the main algorithm. In what follows we use $\{\widehat{W}_i\}_{i\in Q} \subseteq \{W_i\}_{i\in Q}$ to denote the element-wise set inclusion $\widehat{W}_i \subseteq W_i$ for all $i \in Q$. When we talk about maximality, maximality is in (element-wise) set inclusion sense.

Lemma 8.7. Consider two collections of subsets $\widehat{W} = \{\widehat{W}_i\}_{i\in Q}$ and $W = \{W_i\}_{i\in Q}$ of X. If $\widehat{W} \subseteq W \subseteq S$, then $\widehat{W}$ and W satisfy

$$InvPre^{f_i}(G, \widehat{W}, S) \subseteq InvPre^{f_i}(G, W, S) \subseteq S_i \tag{8.18}$$

for any non-sink state $i \in Q_{ns}$.

Lemma 8.8. $W = \{W_i\}_{i\in Q}$ is the maximal winning set with respect to the safe set $S = \{S_i\}_{i\in Q}$ if and only if $\{W_i\}_{i\in Q_{ns}}$ is the maximal solutions of the following equations:

$$W_i = InvPre^{f_i}(G, W, S), \forall i \in Q_{ns}, \tag{8.19}$$

where the components of the winning set W for sink states are chosen according to $W_j = Inv^{f_j}(S_j)$ for all $j \in Q_s$.

The proofs of Lemma 1 and 2 are given in the appendix.

Let us illustrate how Algorithm 8 works, using the switched system shown in Fig. 8.5 with preview automaton in Fig. 8.6 before proving that the proposed algorithm indeed computes the maximal winning set.

Algorithm 9 *InvPre* operator for Algorithm 8

function $InvPre^{f_i}(G,W,S)$
  for j in $Post^G(i)$ **do**
    $C_{0,j} = W_j$ and $T_{ij} = T(i,j)$
    for $l = 1,2,3,...,T_{ij}$ **do**
      $C_{l,j} = PreInt^{f_i}(C_{l-1,j}, S_i)$
    end for
  end for
  $T_{min} = \min_{j \in Post^G(i)} T(i,j)$
  $C_{T_{min}} = Inv^{f_i}(\bigcap_{j \in Post^G(i)} C_{T_{ij},j})$
  $H_i = H(i)$
  for $k = T_{min}+1, \cdots, H_i$ **do**
    $C_k = PreInt^{f_i}(C_{k-1}, S_i)$
    if $J_k = \{j \in Post^G(i) : T_{ij} \geq k\} \neq \emptyset$ **then**
      $C_k = C_k \cap \left(\bigcap_{j \in J_k} C_{T_{ij},j}\right)$
    end if
  end for
  return C_{H_i}
end function

Example 8.4. Since there are no sink nodes in the preview automaton in Fig. 8.6, lines 4-6 in Algorithm 8 are skipped. We use pair (k,l) to indicate the k^{th} iteration of the while loop and l^{th} iteration of the for loop in line 7 and 9 in Algorithm 8, and use $W_i^{k,l}$ to refer to the value of W_i after the (k,l) iteration. Note that at iteration (k,l), only W_l is being updated and the other W_i remains unchanged for $i \neq l$.

Initially $W_1^{0,0} = W_2^{0,0} = \{s_1, s_2\}$. In the iteration $(0,1)$, $W_1^{0,1} = InvPre^{f_1}(G, \{W_1^{0,0}, W_2^{0,0}\}, S) = \{s_1\}$ and $W_2^{0,1} = W_2^{0,0}$. In the iteration $(0,2)$, $W_2^{0,2} = InvPre^{f_1}(G, \{W_1^{0,1}, W_2^{0,1}\}, S) = \{s_2\}$ and $W_1^{0,2} = W_1^{0,1}$. In the following iterations $(1,1), (1,2)$, $W^{1,1}$ and $W^{1,2}$ are unchanged. Therefore, the termination condition in line 7 is satisfied and the output of Algorithm 8 of this example is $W_1 = \{s_1\}$ and $W_2 = \{s_2\}$. It is easy to verify that $W_1 = \{s_1\}$ and $W_2 = \{s_2\}$ form the maximal winning set for this problem. □

The main completeness result is provided next.

Theorem 8.9. *If Algorithm 8 terminates, the tuple of sets $\{W_i\}_{i=1}^n$ it returns is the maximal winning set within the safe set $S = \{S_i\}_{i \in Q}$ of the switched system Σ with the preview automaton G.*

Proof. Suppose that $\{W_i^*\}_{i \in Q}$ is the maximal winning set we are looking for. Let us partition the discrete state space Q into the set of sink states $Q_s = \{q \in Q : H(q) = \infty\}$ and the set of non-sink states $Q_{ns} = Q \backslash Q_s = \{q \in Q : H(q) < \infty\}$.

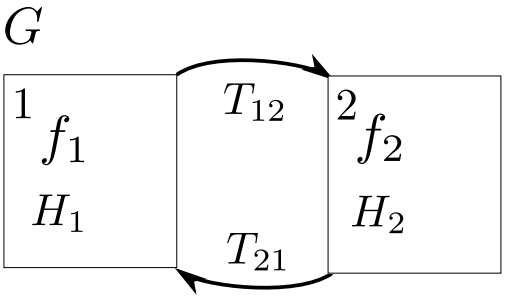


Figure 8.6: The preview automaton corresponding to the switched system in Fig. 8.5. $H_1 = H_2 = 3$ is the least holding time for both modes, and $T_{12} = T_{21} = 1$ is the preview time for transitions $(1,2)$ and $(2,1)$.

If q is a sink state, once the system enters the mode q, the system remains in mode q without any future switching. Therefore, the maximal winning set W_q is trivially the maximal RCIS within the safe set S_q with respect to the dynamics of mode q, that is $W_q^* = Inv^{f_q}(S_q)$. In line 4-6 of Algorithm 8, we compute the maximal winning sets for all the sink states.

We have solved W_i^* for sink state $i \in Q_s$. Let us consider the maximal winning sets for non-sink states. We want to show that W being updated based on lines 7-12 of Algorithm 8 converges to W^*. Without loss of generality, assume that $Q_{ns} = \{1,2,...,s_{ns}\}$ and let the "for" loop in line 9 iterate over the indices $1,2,...,s_{ns}$ in the natural order.

We use $W^{k,l}$ to indicate the updated value of W after the k^{th} iteration of the "while" loop (line 7) and the l^{th} iteration of the "for" loop (line 9). Then the initial value of W is $W^{0,0} = \{W_i^{0,0}\}_{i\in Q}$ where $W_i^{0,0} = S_i$ for all $i \in Q_{ns}$ and $W_j^{0,0} = W_j^*$ for all $j \in Q_s$. According to line 9-10, for all $k \geq 0$ and $0 \leq l \leq s_{ns} - 1$, $W_i^{k,l+1} = InvPre^{f_i}(G, W^{k,l}, S)$ for $i = l+1$ and $W_j^{k,l+1} = W_j^{k,l}$ for all $j \neq l+1$ and $W^{k+1,0} = W^{k,s_{ns}}$.

Now we want to prove that if $W^* \subseteq W^{k,0} \subseteq S$ for some $k \geq 0$, then $W^* \subseteq W^{k,l} \subseteq S$ for any $l \in \{1,2,...,s_{ns}\}$ by induction. (Base case 1) Since we have $W^* \subseteq W^{k,0} \subseteq S$, by Lemma 1 and 2, we have

$$W_i^* = InvPre^{f_i}(G, W^*, S) \subseteq InvPre^{f_i}(G, W^{k,0}, S) = W_i^{k,1} \subseteq S_i$$

for $i = 1$. Since $W^* \subseteq W^{k,0}$ and $W_j^{k,1} = W_j^{k,0}$ for all $j \neq 1$, we have $W^* \subseteq W^{k,1} \subseteq S$. (Induction hypothesis 1) Suppose that $W^* \subseteq W^{k,l} \subseteq S$ for some $0 \leq l \leq s_{ns} - 1$. Again, by Lemma 1 and 2, $W^* \subseteq W^{k,l+1} \subseteq S$. Finally by induction, if $W^* \subseteq W^{k,0} \subseteq S$, $W^* \subseteq W^{k,l} \subseteq S$ for all $l \in \{1,2,...,s_{ns}\}$.

Then next we want to prove by induction that $W^* \subseteq W^{k,0} \subseteq S$ for all $k \geq 0$. (Base case 2) $W^* \subseteq W^{0,0} \subseteq S$ by construction. (Induction hypothesis 2) Suppose $W^* \subseteq W^{k,0} \subseteq S$ for some $k \geq 0$. Then, we have proven that $W^* \subseteq W^{k,s_{ns}} = W^{k+1,0} \subseteq S$. Therefore by induction, $W^* \subseteq W^{k,0} \subseteq S$ for any $k \geq 0$.

The two induction arguments above prove that $W^* \subseteq W^{k,l}$ for any $k \geq 0$ and $0 \leq l \leq s_{ns}$.

Now let us show that $W^{0,0}, W^{0,1}, W^{0,2}, ..., W^{k,0}, W^{k+1,1}, ...$ is a non-expanding sequence. Since $W^{k,s_{ns}} = W^{k+1,0}$ for all $k \geq 0$, it suffices to show $W^{k,l+1} \subseteq W^{k,l}$ for any $k \geq 0$ and $0 \leq l \leq s_{ns} - 1$.

(Base case 3) Note that $InvPre^{f_i}(G, V, S) \subseteq S_i$ for arbitrary $V \subseteq X$ and $i \in Q$. Thus by definition $W_i^{0,1} = InvPre^{f_i}(G, W^{0,0}, S) \subseteq S_i = W_i^{0,0}$ for $i = 1$. Note that $W_j^{0,1} = W_j^{0,0}$ for all $j \neq 1$. Thus $W^{0,1} \subseteq W^{0,0}$. Now consider $W^{0,2}$. Note that $W_j^{0,2} = W_j^{0,1}$ for all $j \neq 2$. For $i = 2$, $W_i^{0,2} = InvPre^{f_i}(G, W^{0,1}, S) \subseteq S_i = W_i^{0,1}$. Thus $W^{0,2} \subseteq W^{0,1}$. Similarly, we have $W^{0,s_{ns}} \subseteq ... \subseteq W^{0,1} \subseteq W^{0,0}$.

(Induction hypothesis 3) Suppose $W^{k,s_{ns}} \subseteq ... \subseteq W^{k,1} \subseteq W^{k,0}$ for some $k > 0$. To show that $W^{k+1,l+1} \subseteq W^{k+1,l}$ for all l, we need another induction argument. (Base case 4) We know $W^{k+1,0} = W^{k,s_{ns}}$, and $W_j^{k+1,1} = W_j^{k+1,0}$ for $j \neq 1$. For $i = 1$, $W_i^{k+1,0} = W_i^{k,s_{ns}} = W_i^{k,1} = InvPre^{f_i}(G, W^{k,0}, S)$. By induction hypothesis 3, $W^{k,s_{ns}} \subseteq W^{k,0}$ and thus by Lemma 8.7 and 8.8,

$$W_i^{k+1,1} = InvPre^{f_i}(G, W^{k+1,0}, S) \subseteq InvPre^{f_i}(G, W^{k,0}, S) = W_i^{k+1,0}$$

for $i = 1$, and therefore $W^{k+1,1} \subseteq W^{k+1,0}$.

(Induction hypothesis 4) Suppose that $W^{k+1,l} \subseteq W^{k+1,l-1} \subseteq ... \subseteq W^{k+1,0}$. By definition, $W_j^{k+1,l+1} = W_j^{k+1,l}$ for all $j \neq l+1$. Also, for $i = l+1$, $W_i^{k+1,l} = W_i^{k,l+1} = InvPre^{f_i}(G, W^{k,l}, S)$. By the induction hypothesis 3 and 4, $W^{k+1,l} \subseteq W^{k,l}$ and thus by Lemma 8.7 and 8.8 again, for $i = l+1$,

$$W_i^{k+1,l+1} = InvPre^{f_i}(G, W^{k+1,l}, S) \subseteq InvPre^{f_i}(G, W^{k,l}, S) = W_i^{k+1,l}$$

and therefore $W^{k,l+1} \subseteq W^{k,l}$. Then by induction 4, we have $W^{k+1,s_{ns}} \subseteq ... \subseteq W^{k+1,1} \subseteq W^{k+1,0}$.

Therefore by the induction 3, we show that $W^{0,0}, ..., W^{k,0}, W^{k+1,1}, ...$ is non-expanding.

By far, we have shown that $W^{0,0}W^{0,1}...W^{k,0}W^{k,1}...$ is a monotonic non-expanding sequence within S, which implies that the limit of this sequence $W^{\infty,0}$ (the output of Algorithm 8) exists and is contained by S thus safe. By line 7-12 of Algorithm 8, $W^{\infty,0}$ is a solution of equations in (8.19). Also, since for any k and l, $W^* \subseteq W^{k,l}$ and W^* is the maximal solution of equations in (8.19), we have $W^{\infty,0} = W^*$. □

Note that the above proof also guarantees termination if the switched system under consideration has finitely many states. For switched systems with continuous state spaces, the non-expanding property of the computed sets guarantees convergence but termination in finite number of steps is not guaranteed, in general. For linear switched systems, termination can still be guaranteed using algorithms from [32, 103] by slightly sacrificing maximality (see also [112]).

Once the maximal winning set (or a winning set) $W^* = \{W_i^*\}_{i \in Q}$ is obtained, a controller can be extracted roughly as follows: for a sink node $i \in Q_s$, the allowable control inputs for each state in the RCIS W_i^* can be obtained by applying the *Pre* operator to W_i^*. For a non-sink node $j \in Q_{ns}$, we need a "invariance" controller to make sure the system state remain in W_i^* before a preview

happens, and a "reachability" controller for each transition $(j,k) \in E$ and each possible preview time $\tau_{jk} \in T(j,k)$ such that from the time point a preview is received by the controller, system state can guarantee to reach W_k in τ_{jk} steps, where the allowable control input for each step can be obtained by applying the *PreInt* recursively for τ_{jk} times. For the "invariance" controller, we also need to make sure that the system state reaches certain parts of the maximal winning set based on the holding time (time steps elapsed since last transition) such that once a preview occurs, the system state is within the domain of the corresponding "reachability" controller. The process of computing the "reachability" controllers actually corresponds to line 2-7 in Algorithm 9, and the process of computing the "invariance" controller corresponds to line 9 and 12-14, if the preview automaton has a singleton preview time interval. The process can be generalized to general preview time intervals from the Algorithm 9 based on the description above.

8.2.3 Illustrative Examples

In the following case studies, we apply the proposed algorithms to switched affine systems, where the state space and safe set are polytopes. In this case *Pre* and *PreInv* operators reduces to polytopic operations, which we implement using the MPT3 toolbox [48].

8.2.3.1 Vehicle Cruise Control

Our first example is a cruise control problem for the scenario shown in Example 8.1. The longitudinal dynamics of a vehicle with road grade is given by

$$\dot{v} = -\frac{f_0}{m} - \frac{f_1}{m} v + \frac{F_w}{m} - g \sin\theta \tag{8.20}$$

where v is the longitudinal speed, m is the vehicle mass, f_0 and f_1 are the coefficients related to frictions, F_w is the wheel force, g is the gravitational acceleration and θ is the road grade. We choose F_w as the control input and θ as a disturbance. We discretize (8.20) with time step $\Delta t = 0.1s$. The discrete-time dynamics with disturbance ranges r_1, r_2 and r_3 consist of the modes 1, 2, 3 in the switched system defined in Example 8.1.

The safety specification is to keep the longitudinal speed within $X = [31.95, 32] m/s$. The speed range is intentionally picked small enough so that the change the road grade induces on the dynamics makes the specification hard to be satisfied. The parameters are chosen as $m = 1650kg$, $f_0 = 0.1N$, $f_1 = 5N \cdot s/m$, $g = 10m/s^2$. The control input range is $F_w \in [-0.65mg, 0.66mg]$. For the preview automaton shown in Fig. 8.4, the holding time for each mode is 2 and the preview time for each transition is 1.

To make a comparison, we compute the maximal RCIS for the dynamics discretized from (8.20)

with disturbance in $[-30.5^\circ, 30.5^\circ]$ (convex hull of r_1, r_2, r_3). If such an invariant set exists, it is a feasible winning set for our problem. However, the resulting RCIS is empty, which suggests that the problem is infeasible if disturbance can vary arbitrarily in $[-30.5^\circ, 30.5^\circ]$. In contrast, the winning set obtained from Algorithm 8 is $\{W_i\}_{i=1}^3$ with $W_1 = W_2 = W_3 = X$. Therefore the preview automaton is crucial in this case study for the existence of a safety controller.

8.2.3.2 Vehicle Lane Keeping Control

In the second example, we apply the proposed method to synthesize a lane-keeping controller, which controls the steering to limit the lateral displacement of vehicle within the lane boundaries.

The lateral dynamics we use are from a linearized bicycle model [112]. The four states of the model consist of the lateral displacement y, lateral velocity v, yaw angle $\Delta\Psi$ and yaw rate r. The vehicle is controlled by the steering input δ_f in range $[-\pi/2, \pi/2]$. We assume that the longitudinal velocity u of the vehicle is constant and equal to $30m/s$. The disturbance r_d is a function of the road curvature, which is what we assume to have preview information on at run-time.

The maximal recommended range of r_d on Michigan highways [1] with respect to $u = 30m/s$ is about $[-0.06, 0.06]$. We divide $[-0.06, 0.06]$ evenly into 5 intervals $d_1 = [-0.06, -0.036]$, $d_2 = [-0.036, -0.012]$, ..., $d_5 = [0.036, 0.06]$ and construct a switched system with 5 modes, where each mode $i \in Q = \{1,2,3,4,5\}$ corresponds to a lateral dynamics with r_d bounded in d_i, denoted by f_i. The corresponding preview automaton is shown in Fig. 8.7, where transitions are only between any two modes with adjacent r_d intervals. For simplicity, the preview time interval $T(i,j) = \tau_c$ for all $(i,j) \in E$, and the least holding time $H(i) = \tau_d$ for all $i \in Q$ for some constants τ_c and τ_d.

The safe set is given by the constraints $|y| \leq 0.9$, $|v| \leq 1.2$, $|\Delta\Psi| \leq 0.05$, $|r| \leq 0.3$ for all modes.

Table 8.1: Computation costs for different (τ_c, τ_d)

(τ_c, τ_d)	#iterations	time (min)
$(1,2)$	4	18.9
$(2,2)$	4	18.0
$(1,1)$	5	20.3
$(5,5)$	3	16.8

We apply Algorithm 8 to compute the maximal winning sets for various τ_c and τ_d. The values of (τ_c, τ_d) with corresponding numbers of iterations at termination and running time are listed in Table 8.1. Denote the maximal winning set with respect to mode 2 for each pair (τ_c, τ_d) in Table 8.1 as $W_{2,(\tau_c,\tau_d)}$.

As a comparison, we compute the maximal RCIS for the lateral dynamics with r_d in $[-0.06, 0.06]$, denoted by W_{inv}. The projections of $W_{2,(\tau_c,\tau_d)}$ and W_{inv} onto 3-dimensional subspaces are shown in Figs. 8.8 and 8.9.

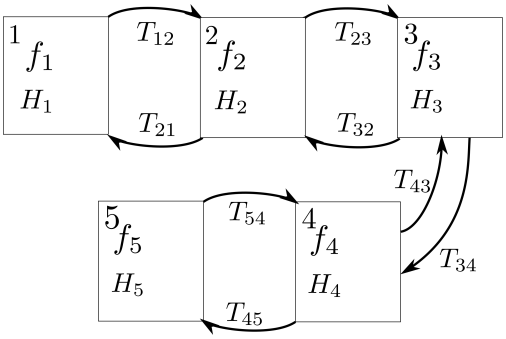
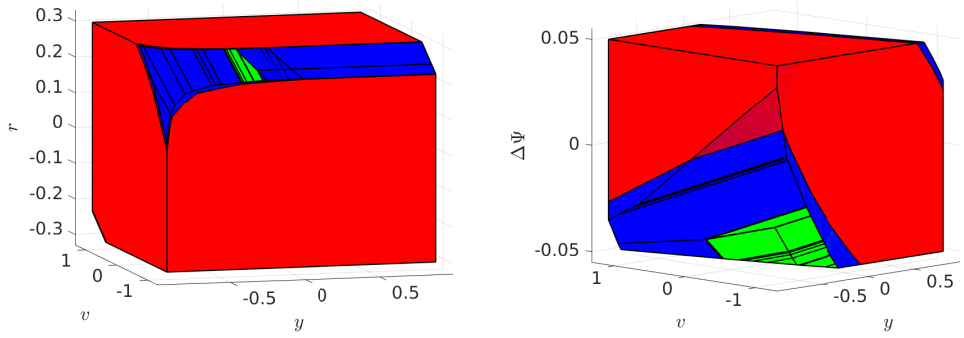


Figure 8.7: Preview automaton for the lane-keeping case study

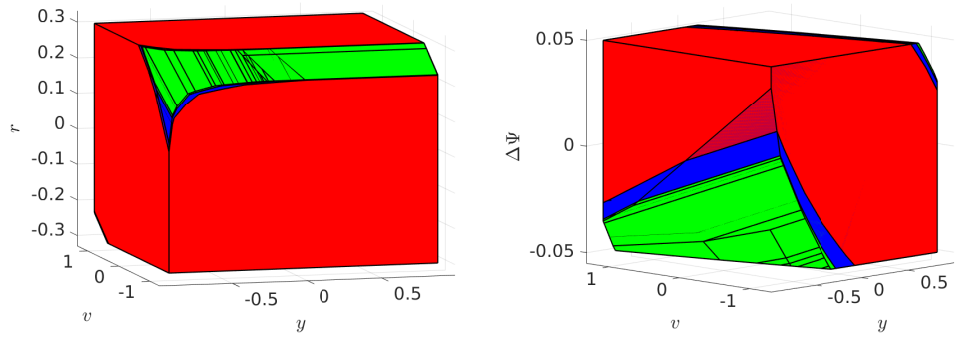


(a) project on $(y, v, \Delta\Psi)$ (b) project on (y, v, r)

Figure 8.8: Projections of $W_{inv}, W_{2,(1,2)}, W_{2,(2,2)}$ onto two subspaces. The red, blue and green regions are the projection of W_{inv}, the difference of projections of $W_{2,(1,2)}$ and W_{inv} and the difference of projections of $W_{2,(2,2)}$ and $W_{2,(1,2)}$.

Fig. 8.8 compares W_{inv}, $W_{2,(1,2)}$ and $W_{2,(2,2)}$, where the holding time τ_d is fixed and the preview time τ_c are tuned to show the effect of preview time on winning set. In theory, $W_{inv} \subseteq W_{(\tau_c,\tau_d)} \subseteq W_{(\tau_c',\tau_d')}$ for any $\tau_c \leq \tau_c'$ and $\tau_d \leq \tau_d'$, which is verified by the numerical result where $W_{inv} \subseteq W_{2,(1,2)} \subseteq W_{2,(2,2)}$. The blue region in Fig. 8.8 shows the difference of $W_{2,(1,2)}$ and W_{inv}, indicating how much we gain from the preview information with $(\tau_c, \tau_d) = (1,2)$ compared to no preview. The green region in Figure 8.8 shows the difference of $W_{2,(2,2)}$ and $W_{2,(1,2)}$, which indicates how much the maximal winning set grows as the preview time τ_c increases from 1 to 2 while the least holding time $\tau_d = 2$ is fixed. As revealed by the size of the green region in Fig. 8.8, the growth of the maximal winning set decreases as the preview time becomes one step longer. Understanding the conditions under which a longer preview does or does not help the growth of the maximal winning set is subject of our future work.

Fig. 8.9 compares W_{inv}, $W_{2,(1,1)}$ and $W_{2,(1,5)}$, where we fix the preview time τ_c and change the least holding time τ_d. The blue and green regions show the difference of $W_{2,(1,1)}$ and W_{inv} and the



(a) project on $(y, v, \Delta\Psi)$ (b) project on (y, v, r)

Figure 8.9: Projections of $W_{inv}, W_{2,(1,1)}, W_{2,(1,5)}$ onto two subspaces. The red, blue and green regions are the projection of W_{inv}, the difference of projections of $W_{2,(1,1)}$ and W_{inv} and the difference of projections of $W_{2,(1,5)}$ and $W_{2,(1,1)}$.

difference of $W_{2,(1,5)}$ and $W_{2,(1,1)}$. Therefore, the size of the green region indicates how much the winning set grows as we increase the least holding time τ_d from 1 to 5. Compared to Fig. 8.8, the winning set is more sensitive to the change of the least holding time τ_d than the change of the preview time τ_c.

Finally, W_{inv}, $W_{2,(1,1)}$, $W_{2,(5,5)}$ are compared in Fig. 8.9, where we increase τ_c and τ_d simultaneously. $W_{2,(5,5)}$ is numerically equal to $W_{2,(1,5)}$, and thus its projections are the same as the projections of $W_{2,(1,5)}$ shown in Fig. 8.9. In fact, the winning sets with respect to modes 2, 3, 4 for $\tau_c = 1, \tau_d = 5$ and $\tau_c = 5, \tau_d = 5$ are numerically equal; the winning set with respect to mode 1 and 5 slightly grows when (τ_c, τ_d) changes from $(1,5)$ to $(5,5)$, but the growth is too small to be visualized. The observation in Fig. 8.8 and 8.9 reveals one theoretical conjecture: If the preview time and the least holding time are large enough, a longer preview time and/or a longer holding time will not increase the size of the maximal winning set. That is, the size of the maximal winning set converges as the preview time and the least holding time increase. To verify this conjecture is part of our future work.

8.2.4 Summary

In this section, we introduce preview automaton and provide an algorithm for safety control synthesis in the existence of preview information. The proposed algorithm is shown to compute the maximal winning set upon termination. These ideas are demonstrated with two examples from the autonomous driving domain. As shown in these examples, incorporation of preview information in control synthesis leads to less conservative safety guarantees compared to standard RCIS based approaches. In the future, we will investigate the use of preview automaton for synthesizing

controllers from more general specifications. We also have some ongoing work investigating the connections of preview automaton with discrete-time I/O hybrid automaton with clock variables representing preview and holding times.

8.3 Safety Control with Uncertain Preview

In Chapter 7 and Section 8.1, we study the safety control problem for systems with preview under the assumption that the preview on future disturbances is accurate. This assumption can be satisfied in certain control applications, such as reference tracking [114, 28]. However, if the preview information is obtained from a perception module, it is likely that the preview is corrupted by measurement noise and thus becomes uncertain. Motivated by these applications, in this section, we study how to synthesize safety controllers for systems with uncertain preview.

Notation. For a collection $\mathscr{X} = \{X_k\}_{k=1}^N$ of sets, the union $\cup_{X\in\mathscr{X}} X$ is denoted by $\cup\mathscr{X}$ for short.

8.3.1 Problem Setup

8.3.1.1 Systems with uncertain preview

Consider a discrete-time dynamical system Σ in form of

$$\Sigma : x(t+1) = f(x(t), u(t), d(t)), \tag{8.21}$$

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$ and disturbance $d \in D \subseteq \mathbb{R}^l$. The disturbance set D is assumed to be compact. A system Σ in form of (8.21) is said to have *p-step (uncertain) preview* if at each time step t, the controller has access to

- the current state $x(t)$, and
- the noisy measurements $(\widehat{d}(t+k))_{k=0}^{p-1}$ of current and incoming disturbances $(d(t+k))_{k=0}^{p-1}$ in p steps, denoted by $\widehat{d}_{1:p}(t)$ for short.

To simplify the problem setup, we only assume that the disturbance measurements are corrupted by noise, while the state can be accurately measured at each time instance. Moreover, we assume that there exist p measurement functions $h_k : D \to 2^D \backslash \emptyset$ such that $\widehat{d}_k(t) \in h_k(d_k(t))$.

In the next two subsections, we first introduce the output-feedback control problem for general discrete-time systems, and then formulate the safety control problem for a system with uncertain preview as a special output-feedback safety control problem.

8.3.1.2 Output Feedback Safety Control Problem

A discrete-time system Σ with output function h is given by

$$(\Sigma, h): \begin{array}{c} x(t+1) = f(x(t), u(t), d(t)), \\ y(t) \in h(x(t)), \end{array} \tag{8.22}$$

where Σ is a discrete-time system as in (8.21), $y \in Y \subseteq \mathbb{R}^q$ is the output of Σ, and $h : \mathbb{R}^n \to 2^Y \backslash \emptyset$ is a function mapping each state $x \in \mathbb{R}^n$ to a nonempty subset $h(x)$ of Y. The set Y contains all possible outputs of the system. For any $y \in Y$, we define the inverse of h by $h^{-1}(y) := \{x \in \mathbb{R}^n \mid y \in h(x)\}$. We make the following assumption on the output function h.

Assumption 8.1. *For all $y \in Y$, the inverse $h^{-1}(y)$ is nonempty.*

Note that we can always make the output function h satisfy Assumption 8.1 by redefining $Y = \cup_{x \in \mathbb{R}^n} h(x)$. For a system in (8.22) , since its states cannot be directly observed, we can infer where the current state is located based on the system outputs over time. Specifically, suppose that the initial state $x(0)$ is known to be within some subset X_{-1} of $\mathbb{R}^n$. Then, $X_0 := X_{-1} \cap h^{-1}(y(0))$ is the set of all the initial state $x(0) \in X_{-1}$ consistent with the output $y(0)$. For $t \geq 1$, given $y(0:t)$ and $u(0:t-1)$, we define X_t as the set of states $x(t)$ such that there exists $x(0:t-1)$ satisfying

(a) $x(0) \in X_0$;

(b) for all k from 0 to $t-1$, $x(k+1) \in f(x(k), u(k), D)$ and $y(k+1) \in h(x(k+1))$.

One can interpret X_t as the set of state estimates of $x(t)$ consistent with the outputs $y(0:t)$ and inputs $u(0:t-1)$ observed up to time t. In the remainder of this section, we call X_t the *belief set* of the state $x(t)$ (also known as information set in the literature [13]). Note that in our previous discussion, the construction of X_0 is different from X_t with $t \geq 1$ due to the extra assumption $x(0) \in X_{-1}$. In practice, this set X_{-1} may be inferred from additional observations on $x(0)$, and thus can vary with the initial state $x(0)$. The following assumption allows us to deal with all different X_{-1} in an unified manner.

Assumption 8.2. *There exists a collection $\mathscr{X}_0$ of belief sets X_0 of the initial state $x(0)$ such that given the output $y(0)$ and any additional information on $x(0)$ available at time $t = 0$, one can identify at least one element $X_0 \subseteq h^{-1}(y_0)$ from $\mathscr{X}_0$ such that the actual initial state $x(0) \in X_0$.*

A trivial collection $\mathscr{X}_0$ of initial belief sets satisfying Assumption 8.2 is $\{h^{-1}(y) \mid y \in Y\}$. By definition, given an initial belief set $X_0 \in \mathscr{X}_0$, the belief set X_t on $x(t)$ can be determined by the inputs $u(0:t-1)$ and the outputs $y(1:t)$. The following lemma shows that X_t is actually Markovian, namely that X_t is a function of X_{t-1}, $u(t-1)$, and $y(t)$.

Lemma 8.10. For all $t \geq 1$, the set X_t is uniquely determined by X_{t-1}, $u(t-1)$, and $y(t)$, satisfying

$$X_t = f(X_{t-1}, u(t-1), D) \cap h^{-1}(y(t)) =: F(X_{t-1}, u(t-1), y(t)). \tag{8.23}$$

Proof. Let $\overline{X}_t$ be the RHS of (8.23). We want to show that $X_t = \overline{X}_t$.

First, pick an arbitrary $x(t) \in X_t$. By definition of X_t, there exist $x(0:t-1)$ such that $x(k) \in X_k$ for k from 0 to $t-1$, and $x(t) \in f(x(t-1), u(t-1), D) \cap h^{-1}(y(t))$. Therefore, $x(t) \in \overline{X}_t$, and thus, $X_t \subseteq \overline{X}_t$.

Next, pick an arbitrary $x(t) \in \overline{X}_t$. Then, there exists $x_{t-1} \in X_{t-1}$ such that $x(t) \in f(x(t-1), u(t-1), D)$ and $y(t) \in h(x(t))$. If $t = 1$, it is obvious that $x(t) \in X_t$ by definition. If $t > 1$, since $x(t-1) \in X_{t-1}$, there further exists $x(0:t-2)$ satisfying the points (a) and (b) in the definition of X_{t-1} with respect to $x(t-1)$. Then, it can be shown that $x(0:t-1)$ satisfies the points (a) and (b) in the definition of X_t with respect to $x(t)$. Hence, $x(t) \in X_t$ and thus, $\overline{X}_t \subseteq X_t$. □

Given the collection $\mathscr{X}_0$ of the initial belief sets, we denote the collection of all possible belief sets of $x(t)$ by $\mathscr{X}_t$. By Lemma 8.10, we can characterize $\mathscr{X}_t$ with $t \geq 1$ with the recursive equation

$$\mathscr{X}_t = \{F(X_{t-1}, u, y) \mid X_{t-1} \in \mathscr{X}_{t-1}, u \in \mathbb{R}^m, y \in \cup_{x^+ \in f(X_{t-1}, u, D)} h(x^+)\}, \tag{8.24}$$

where the set $\cup_{x^+ \in f(X_{t-1}, u, D)} h(x^+)$ contains all possible outputs at time t given that the state $x(t-1)$ is in $\mathscr{X}_{t-1}$. We call the set $\mathscr{X} := \cup_{t \geq 0} \mathscr{X}_t$ of all possible belief sets the *belief space* of the system.

Since the belief set X_t is the minimal region containing the state $x(t)$ one can infer from the historical inputs and outputs at time t, the control input $u(t)$ should be determined based on the belief set X_t. Therefore, the controllers considered in this section are functions $\mathbf{u} : \mathscr{X} \to \mathbb{R}^m$ from the belief space $\mathscr{X}$ to the input space $\mathbb{R}^m$. Specifically, given a controller $\mathbf{u}$, the control input $u(t)$ at time t is $\mathbf{u}(X_t)$, where X_t is by definition determined by X_0, $y(0:t)$, and $u(0:t-1)$.

Remark 8.2. To implement a controller $\mathbf{u} : \mathbf{X} \to \mathbb{R}^n$, we have to represent the belief set X_t numerically. One simple representation of X_t is the tuple $(X_0, y(0:t), u(0:t-1))$, from which one can uniquely determine X_t. However, as t goes to infinity, this representation of X_t requires infinite memory. Therefore, a output-feedback controller $\mathbf{u} : \mathscr{X} \to \mathbb{R}^m$ in general requires infinite memory, unless every element in $\mathscr{X}$ can be represented with finite memory.

Now we are ready to define the output-feedback safety control problem. Let $S_{xu} \in \mathbb{R}^{n+m}$ be a safe set of a system Σ with output h. Given an initial belief set $X_0 \in \mathscr{X}_0$, a controller $\mathbf{u} : \mathscr{X} \to \mathbb{R}^m$ is *safe* if all possible state-input trajectories $(x(t), u(t))_{t \geq 0}$ starting from any $x(0) \in X_0$ under the control of $\mathbf{u}$ stay in the safe set S_{xu} indefinitely. A set $X_0 \in \mathscr{X}_0$ is a *winning* belief set if there exists a safe controller $\mathbf{u}$ with respect to X_0.

Problem 8.3. *Given a system Σ with output function h, safe set S_{xu}, and a collection $\mathscr{X}_0$ of initial belief sets, identify the set $\mathscr{W}_{max}$ of all winning belief sets, and the corresponding safe controllers.*

8.3.1.3 Problem Statement

Consider a system Σ with p-step uncertain preview, as described in Section 8.3.1.1. Let S_{xu} be a safe set of Σ. Recall the p-augmented system Σ_p of Σ and its safe set $S_{xu,p}$, defined in (7.2) and (7.3). The safety control problem for a system Σ with p-step uncertain preview can be formulated as the output-feedback safety control problem for the p-augmented system Σ_p with respect to the safe set $S_{xu,p}$. The output function h of Σ_p is given by

$$h(x, d_{1:p}) = \{x\} \times h_1(d_1) \times \cdots \times h_p(d_p). \tag{8.25}$$

We denote the output $(x(t), \widehat{d}_{1:p}(t))$ of Σ_p at time t by $(x(t), \widehat{d}_{1:p}(t)) \in h(x(t), d_{1:p}(t))$. We make the following assumption to simplify the derivation later.

Assumption 8.3. *At time $t = 0$, for each j from 1 to p, we have $(p-j)$ additional measurements $(\widehat{d}_{j+k}(-k))_{k=1}^{p-j}$ of the state $d_j(0)$ of Σ_p satisfying $\widehat{d}_{j+1}(-1) \in h_{j+1}(d_j(0))$, ..., $\widehat{d}_p(j-p) \in h_p(d_j(0))$.*

Assumption 8.3 essentially states that we start previewing disturbances of Σ at time $t = -(p-1)$. This assumption can be easily fulfilled in real-world applications if we allow the perception module that generates preview to start running p steps earlier than the control module. According to Assumption 8.3, the collection $\mathscr{X}_0$ of the initial belief sets for Σ_p is

$$\mathscr{X}_0 := \mathbb{R}^n \times \mathscr{D}_1 \times \mathscr{D}_2 \times \cdots \times \mathscr{D}_p, \tag{8.26}$$

where the collection $\mathscr{D}_j := \bigcup_{d \in D} \left\{ \bigcap_{k=j}^{p} h_k^{-1}(y_k) \mid y_k \in h_k(d) \right\}$ contains all possible belief sets of the state $d_j(0)$ given its $(p-j+1)$ measurements $\widehat{d}_j(0)$, ..., $\widehat{d}_p(j-p)$. At run time, given the initial output $(x(0), \widehat{d}_{1:p}(0)) \in h(x(0), d_{1:p}(0))$ and the additional preview information in Assumption 8.3, the initial belief set X_0 is

$$X_0 = \{x(0)\} \times \bigcap_{k=1}^{p} h_k^{-1}(\widehat{d}_k(1-k)) \times \cdots \times \bigcap_{k=j}^{p} h_k^{-1}(\widehat{d}_k(j-k)) \times \cdots \times h_p^{-1}(\widehat{d}_p(0)) \in \mathscr{X}_0. \tag{8.27}$$

Now we have all the ingredients for the safety control problem of systems with uncertain preview.

Problem 8.4. *Given a system Σ with p-step uncertain preview and its safe set S_{xu}, find a solution to Problem 8.3 with respect to the system Σ_p with output function h in* (8.25)*, the safe set $S_{xu,p}$, and the collection $\mathscr{X}_0$ of initial belief sets in* (8.26)*.*

Remark 8.3. Let $\mathscr{X}_0' := \{h^{-1}(x, \widehat{d}_{1:p}) \mid (x, \widehat{d}_{1:p}) \in \cup h(\mathbb{R}^n \times D^p)\}$, that is, the collection of initial belief sets determined only by the initial output $(x(0), \widehat{d}_{1:p}(0)) \in h(x(0), d_{1:p}(0))$ for all possible $(x(0), d_{1:p}(0)) \in \mathbb{R}^n \times D^p$. It can be shown that $\mathscr{X}_0$ in (8.26) is equal to $\mathscr{X}_p'$, where $\mathscr{X}_p'$ is defined according to the recursive equation in (8.24) with respect to $\mathscr{X}_0'$. Furthermore, it can be shown that any safe controller with respect to $X_0' \in \mathscr{X}_0'$ can be extended from a safe controller with respect to some $X_0 \in \mathscr{X}_0$. In this sense, we do not lose any generality by making Assumption 8.3.

8.3.2 Set-Invariance Based Approach

In this subsection, we first characterize the solution to Problem 8.3, addressing the output-feedback safety control problem for any system. While computing the general solution for Problem 8.3 is challenging, we present in the latter part of this subsection an efficient computation method for the solution to Problem 8.4, a specific case of Problem 8.3. This approach leverages the inherent structure within Σ_p.

8.3.2.1 General Approach

Consider a general system Σ as in (8.21) with output h, safe set S_{xu}, and a collection $\mathscr{X}_0$ of initial belief sets. We define a set dynamics Σ_F by

$$\Sigma_F : X(t+1) = F(X(t), u(t), d_y(t)), \tag{8.28}$$

with state $X(t) \in \mathscr{X}$, input $u(t) \in \mathbb{R}^m$, and disturbance $d_y(t) \in D_y(X(t), u(t)) := \bigcup_{x^+ \in f(X(t),u(t),D)} h(x^+)$. Recall that F is defined in (8.23), and $\mathscr{X}$ is the belief space with respect to $\mathscr{X}_0$. It can be checked that for any $X(t) \in \mathscr{X}$, $u(t) \in \mathbb{R}^m$, and $d_y(t) \in D_y(X(t), u(t))$, $X(t+1) \in \mathscr{X}$, and thus, Σ_F is well-defined over $\mathscr{X}$.

According to Lemma 8.10, any sequence $(X_t)_{t\geq 0}$ of belief sets of Σ with respect to inputs $(u(t))_{t\geq 0}$ and outputs $(y(t))_{t\geq 0}$ must be the trajectory of Σ_F with $X(0) = X_0$ corresponding to the same inputs $(u(t))_{t\geq 0}$ and the disturbance sequence $d_y(t) = y(t+1)$ for all $t \geq 0$, and vice versa. We define the safe set S_F of Σ_F by

$$S_F := \{(X, u) \mid X \in \mathscr{X}, X \times \{u\} \subseteq S_{xu}\}. \tag{8.29}$$

Definition 8.7. For a system $x^+ = f(x, u, d)$ with state-input-dependent disturbance set $d \in D(x, u)$, a set C is an RCIS of this system with respect to a safe set S_{xu} if for all $x \in C$, there exists u such that $(x, u) \in S_{xu}$ and $f(x, u, D(x, u)) \subseteq C$. Given an RCIS of the system, its admissible input set at state x is defined by $\mathscr{A}(x, C) := \{u \mid (x, u) \in S_{xu}, f(x, u, D(x, u)) \subseteq C\}$.

Definition 8.7 extends Definition 2.3 for systems with state-input-dependent disturbance, such as Σ_F. Let $\mathscr{C}_{max}$ be the maximal RCIS of Σ_F with respect to S_F. The following theorem reveals the relation between $\mathscr{C}_{max}$ and the maximal winning set $\mathscr{W}_{max}$ in Problem 8.3.

Theorem 8.11. *Given a system (Σ, h) with output function h, safe set S_{xu}, and a collection $\mathscr{X}_0$ of initial belief sets, the maximal winning set $\mathscr{W}_{max}$ is equal to the intersection of $\mathscr{X}_0$ and the maximal RCIS $\mathscr{C}_{max}$ of Σ_F with respect to S_F. That is, $\mathscr{W}_{max} = \mathscr{C}_{max} \cap \mathscr{X}_0$. Furthermore, any controller $\mathbf{u} : \mathscr{X} \to \mathbb{R}^m$ satisfying $\mathbf{u}(X) \in \mathscr{A}(X, \mathscr{C}_{max})$ for all $X \in \mathscr{C}_{max}$ is safe with respect to any $X_0 \in \mathscr{W}_{max}$.*

Proof. Pick any $X_0 \in \mathscr{C}_{max} \cap \mathscr{X}_0$. By the definition of $\mathscr{C}_{max}$, there exists a controller $\mathbf{u} : \mathscr{X} \to \mathbb{R}^m$ satisfying that for all $X \in \mathscr{C}_{max}$, $\mathbf{u}(X) \in \mathscr{A}(X, \mathscr{C}_{max})$. Therefore, all feasible state-input trajectories $(X(t), u(t))_{t \geq 0}$ of Σ_F from $X(0) = X_0$ under the control of $\mathbf{u}$ stay within S_F indefinitely, implying that $\mathbf{u}$ is a safe controller of (Σ, h) with respect to X_0. Thus, $X_0 \in \mathscr{W}_{max}$. Since X_0 is picked arbitrarily from $\mathscr{C}_{max} \cap \mathscr{X}_0$, $\mathscr{C}_{max} \cap \mathscr{X}_0 \subseteq \mathscr{W}_{max}$.

Next, pick an arbitrary $X_0 \in \mathscr{W}_{max}$. There exists a safe controller $\mathbf{u} : \mathscr{X} \to \mathbb{R}^m$ with respect to X_0. From the definition of safe controller, it can be shown that any closed-loop trajectory of Σ_F from X_0 under the control of $\mathbf{u}$ stays within S_F indefinitely, implying that $X_0 \in \mathscr{C}_{max}$. Thus, $\mathscr{W}_{max} \subseteq \mathscr{C}_{max}$. □

According to Theorem 8.11, the maximal winning set of the output-feedback safety control problem is exactly $\mathscr{C}_{max} \cap \mathscr{X}_0$. In other words, to solve Problem 8.3, one only needs to find the maximal RCIS of Σ_F with respect to S_F. Since the system Σ_F is a set dynamics, its maximal RCIS is in general challenging to compute. However, in the next part of this subsection, we show that when restricting to the system (Σ_p, h) defined in Section 8.3.1.3, the corresponding set dynamics can be converted into a finite-dimensional nonlinear system, whose maximal RCIS can be more efficiently computed.

8.3.2.2 Efficient Computation for Systems with Uncertain Preview

Consider the p-augmented Σ_p with output function h, safe set $S_{xu,p}$, and the collection $\mathscr{X}_0$ of initial belief sets in Section 8.3.1.3. We denote the system Σ_F in (8.28) and the safe set S_F in (8.29) defined with respect to (Σ_p, h), $S_{xu,p}$, and $\mathscr{X}_0$ by $\Sigma_{F,p}$ and $S_{F,p}$. The reminder of this subsection focuses on the computation of the maximal RCIS of $\Sigma_{F,p}$ with respect to $S_{F,p}$, denoted by $\mathscr{C}_{max,p}$.

Definition 8.8. Consider two discrete-time systems $\Sigma_x : \ x^+ = f(x,u,d)$ and $\Sigma_z : \ z^+ = g(z,u,w)$ with states $x \in X$ and $z \in Z$, input $u \in U$, and disturbances $d \in D(x,u)$, and $w \in W(z,u)$. The system Σ_x *emulates* the system Σ_z if there exists a surjective function $\tau : X \to Z$ such that for all $x \in X$ and $u \in U$, $\tau(f(x,u,D(x,u))) = g(\tau(x),u,W(\tau(x),u))$.

Lemma 8.12. For two systems Σ_x and Σ_z as in Definition 8.8, suppose that Σ_x emulates Σ_z. Then, the maximal RCIS $C_{max,z}$ of the system Σ_z with respect to a safe set S_{zu} is equal to $\tau(C_{max,x})$, where $C_{max,x}$ is the maximal RCIS of the system Σ_x with respect to the safe set $S_{xu} := \{(x,u) \mid (\tau(x),u) \in S_{zu}\}$.

Proof. First, we want to show that $\tau(C_{max,x})$ is an RCIS of Σ_z with respect to S_{zu}. Let $z \in \tau(C_{max,x})$. Then, there exists $(x,u) \in S_{xu}$ such that $x \in C_{max,x}$, $\tau(x) = z$, and $f(x,u,D(x,u)) \subseteq C_{max,x}$. Since Σ_x emulates Σ_z, we have $g(\tau(x),u,W(\tau(x),u)) = \tau(f(x,u,D(x,u))) \subseteq \tau(C_{max,x})$. By the definition of S_{xu}, $(x,u) \in S_{xu}$ implies $(z,u) = (\tau(x),u) \in S_{zu}$. Thus, $\tau(C_{max,x})$ is an RCIS of Σ_z with respect to S_{zu}, and thereby, $\tau(C_{max,x}) \subseteq C_{max,z}$.

Second, let $x \in \tau^{-1}(C_{max,z})$. Then, $\tau(x) \in C_{max,z}$, and there exists u such that $g(\tau(x),u,W(\tau(x),u)) \subseteq C_{max,z}$. Since Σ_x emulates Σ_z,

$$f(x,u,D(x,u)) \subseteq \tau^{-1}(\tau(f(x,u,D(x,u)))) = \tau^{-1}(g(\tau(x),u,W(\tau(x),u))) \subseteq \tau^{-1}(C_{max,z}).$$

Also, since $(\tau(x),u) \in S_{zu}$, $(x,u) \in S_{xu}$ by definition. Thus, $\tau^{-1}(C_{max,z})$ is an RCIS of Σ_x with respect to S_{xu}, and thus, $\tau^{-1}(C_{max,z}) \subseteq C_{max,x}$. Since τ is surjective, $\tau(\tau^{-1}(C_{max,z})) = C_{max,z} \subseteq \tau(C_{max,x})$, which completes the proof. □

Recall that we want to compute the maximal RCIS $\mathscr{C}_{max,p}$ of the set dynamics $\Sigma_{F,p}$. According to Lemma 8.12, if we can find an auxiliary system Σ_{aux} that emulates $\Sigma_{F,p}$, then the maximal RCIS of $\Sigma_{F,p}$ can be obtained from the maximal RCIS of the auxiliary system. Next, we show how to construct such an auxiliary system.

First, note that due to the nilpotent structure in the dynamics of Σ_p corresponding to the states $d_{1:p}$ and the specific output function h, the belief set X_t at time t only depend on $x(t)$, $\{\widehat{d}_k(t+1-k)\}_{k=1}^{p}$ (p measurements of $d_1(t)$), $\{\widehat{d}_k(t+2-k)\}_{k=2}^{p}$ ($(p-1)$ measurements of $d_2(t)$), ..., $\widehat{d}_p(t)$ (the first measurement of $d_p(t)$). Specifically, we have

$$X_t = \{x(t)\} \times \bigcap_{k=1}^{p} h_k^{-1}(\widehat{d}_k(t+1-k)) \times \cdots \times \bigcap_{k=j}^{p} h_k^{-1}(\widehat{d}_k(t+j-k)) \times \cdots \times h_p^{-1}(\widehat{d}_p(t)). \tag{8.30}$$

Comparing (8.27) and (8.30), it can be shown that $\mathscr{X}_t = \mathscr{X}_0$ (thanks to Assumption 8.3) and thus $\mathscr{X} = \mathscr{X}_0$. Furthermore, by (8.30), each $X_t \in \mathscr{X}_t = \mathscr{X}$ can be uniquely determined by the vector $(x(t), \widehat{d}_{1,1:p}(t), \widehat{d}_{2,2:p}(t), \cdots, \widehat{d}_{p,p}(t))$, where $\widehat{d}_{j,k}(t)$ denotes $\widehat{d}_k(t+j-k)$, the $(p+1-k)$ th measurement of the state $d_j(t)$ of Σ_p (recall that the first measurement of $d_j(t)$ is $\widehat{d}_p(t+j-p)$). We denote the set of all feasible $(x(t), \widehat{d}_{1,1:p}(t), \widehat{d}_{2,2:p}(t), \cdots, \widehat{d}_{p,p}(t))$ by Ξ, that is,

$$\Xi = \mathbb{R}^n \times \widehat{D}_1 \times \cdots \times \widehat{D}_p, \tag{8.31}$$

where $\widehat{D}_j := \bigcup_{d\in D} \Pi_{k=j}^{p} h_k(d)$ is the set of all potential measurements $\widehat{d}_{j,j:p}$ of a disturbance $d \in D$ obtained from the measurement functions $h_{j:p}$. We define the function $\tau : \Xi \to \mathscr{X}$ by

$$\tau(x, \widehat{d}_{1,1:p}, \cdots, \widehat{d}_{p,p}) := \{x\} \times \Delta_1(\widehat{d}_{1,1:p}) \times \cdots \times \Delta_j(\widehat{d}_{j,j:p}) \times \cdots \times \Delta_p(\widehat{d}_{p,p}), \tag{8.32}$$

where $\Delta_j(\widehat{d}_{j,j:p}) := \bigcap_{k=j}^{p} h_k^{-1}(\widehat{d}_{j,k})$ is the belief set of the state d_j. Based on the previous discussion, it can be shown that τ is surjective, that is, $\tau(\Xi) = \mathscr{X}$. Next, we define an auxiliary system Σ_{aug} over Ξ by

$$\Sigma_{aux} : \begin{bmatrix} x(t+1) \\ \widehat{d}_{1,1}(t+1) \\ \widehat{d}_{1,2:p}(t+1) \\ \widehat{d}_{2,2}(t+1) \\ \widehat{d}_{2,3:p}(t+1) \\ \vdots \\ \widehat{d}_{p,p}(t+1) \end{bmatrix} = \begin{bmatrix} f(x(t), u(t), \delta_0(t)) \\ \delta_1(t) \\ \widehat{d}_{2,2:p}(t) \\ \delta_2(t) \\ \widehat{d}_{3,3:p}(t) \\ \vdots \\ \delta_p(t) \end{bmatrix}, \tag{8.33}$$

with the state $\xi := (x, \widehat{d}_{1,1:p}, \cdots, \widehat{d}_{p,p}) \in \Xi$, input $u \in \mathbb{R}^m$, and disturbances $\delta_{0:p} \in D_{aux}(\xi) := \Delta_1(\widehat{d}_{1,1:p}) \times \Pi_{j=1}^{p-1} \cup h_j(\Delta_{j+1}(\widehat{d}_{j+1,j+1:p})) \times \cup h_p(D)$. Recall that for a subset X of $\mathbb{R}^n$, $\cup h_j(X)$ denotes $\cup_{x\in X} h_j(x)$.

Lemma 8.13. The system Σ_{aux} in (8.33) emulates the system $\Sigma_{F,p}$.

Proof. According to Definition 8.8, we want to check that for all $\xi \in \Xi$, for all $u \in \mathbb{R}^m$,

$$\tau(f_{aux}(\xi, u, D_{aux}(\xi)) = f_{F,p}(\tau(\xi), u, D_{y,p}(\tau(\xi), u)), \tag{8.34}$$

where f_{aux} is dynamics function of Σ_{aux} (that is, RHS of (8.33)), and $f_{F,p}$ is the dynamics function of $\Sigma_{F,p}$, and $D_{y,p}$ is the disturbance set of $\Sigma_{F,p}$ (that is, D_y in (8.28) with respect to Σ_p and h).

First, by plugging the dynamics f_{aux} and the disturbance D_{aux} into τ, it can be shown that

$$\begin{aligned} &\tau(f_{aux}(\xi, u, D_{aux}(\xi)) \\ =&f(x, u, \Delta_1(\widehat{d}_{1,1:p})) \times \Pi_{j=1}^{p-1} \Delta_j(\cup h_j(\Delta_{j+1}(\widehat{d}_{j+1,j+1:p})) \times \{\widehat{d}_{j+1,j+1:p}\}) \times \Delta_p(\cup h_p(D)). \end{aligned} \tag{8.35}$$

By the definition of Δ_j, it can be shown that

$$\begin{aligned} &\Delta_j(\cup h_j(\Delta_{j+1}(\widehat{d}_{j+1,j+1:p})) \times \{\widehat{d}_{j+1,j+1:p}\}) \\ =&\{h_j^{-1}(d) \cap \Delta_{j+1}(\widehat{d}_{j+1,j+1:p}) \mid d \in \cup h_j(\Delta_{j+1}(\widehat{d}_{j+1,j+1:p}))\} \end{aligned} \tag{8.36}$$

Next, by definition,

$$D_{y,p}(\tau(\xi),u) = f(x,u,\Delta_1(\widehat{d}_{1,1:p})) \times \Pi_{j=1}^{p-1} \cup h_j(\Delta_{j+1}(\widehat{d}_{j+1,j+1:p})) \times \cup h_p(D). \tag{8.37}$$

Thus,

$$\begin{aligned}
&f_{F,p}(\tau(\xi),u,D_{y,p}(\tau(\xi),u)) \\
=&\{f_p(X_t,u,D) \cap h^{-1}(y) \mid y \in D_{y,p}(\tau(\xi),u)\} && (8.38)\\
=&\{\{x^+\} \times \Pi_{j=1}^{p-1}(h_j^{-1}(\widehat{d}_j^+) \cap \Delta_{j+1}(\widehat{d}_{j+1,j+1:p})) \times h_p^{-1}(\widehat{d}_p^+)) \mid (x^+,\widehat{d}_{1:p}^+) \in D_{y,p}(\tau(\xi),u)\} && (8.39)\\
=&f(x,u,\Delta_1(\widehat{d}_{1,1:p})) \times \Pi_{j=1}^{p-1}\{h_j^{-1}(d) \cap \Delta_{j+1}(\widehat{d}_{j+1,j+1:p}) \mid d \in \cup h_j(\Delta_{j+1}(\widehat{d}_{j+1,j+1:p}))\} \times \\
&\Delta_p(\cup h_p(D)) && (8.40)\\
=&f(x,u,\Delta_1(\widehat{d}_{1,1:p})) \times \Pi_{j=1}^{p-1}\Delta_j(\cup h_j(\Delta_{j+1}(\widehat{d}_{j+1,j+1:p})) \times \{\widehat{d}_{j+1,j+1:p}\}) \times \Delta_p(\cup h_p(D)) && (8.41)
\end{aligned}$$

The last equality above is due to (8.36). Since (8.41) is the same as (8.35), we have proven the equality in (8.34). □

Theorem 8.14. *The maximal winning set for the system Σ with p-step uncertain preview is equal to $\tau(C_{max,aux})$, where the function τ is defined in* (8.32)*, and the set $C_{max,aux}$ is the maximal RCIS of Σ_{aux} with respect to the safe set $\{(\xi,u) \mid \tau(\xi,u) \in S_{F,p}\}$.*

Proof. By Lemmas 8.12 and 8.13, $\tau(C_{max,aux})$ is the maximal RCIS of $\Sigma_{F,p}$ with respect to $S_{F,p}$. Then, by Theorem 8.11, the maximal winning set is equal to $\tau(C_{max,aux}) \cap \mathscr{X}_0 = \tau(C_{max,aux})$, since $\tau(C_{max,aux}) \subseteq \mathscr{X} = \mathscr{X}_0$. □

Thanks to Theorem 8.14, the solution to Problem 8.4 can be obtained by computing the maximal RCIS $C_{max,aux}$ of the auxiliary system Σ_{aux}. Compared with the set dynamics $\Sigma_{F,p}$, the auxiliary system Σ_{aux} is of finite dimensions, and thus, its maximal RCIS can be approximated using existing backward reachability toolboxes for nonlinear systems (such as [73]).

Remark 8.4. Since any belief set $X \in \mathscr{X}$ can be represented by a finite-dimensional vector $\xi \in \Xi$, output-feedback controllers for the p-augmented system Σ_p only require finite amount of memory.

8.3.3 Summary

In this section, we study the safety control problem for systems with uncertain preview. We first characterize the solution to the safety control problem for an arbitrary system with output, establishing a connection between the maximal winning set and the maximal RCIS of a set dynamics. Subsequently, we show that when dealing specifically with systems with uncertain preview, the

maximal winning set can be derived from the maximal RCIS of a finite-dimensional nonlinear system, which can be approximately more efficiently than the maximal RCIS of the set dynamics.

CHAPTER 9

Opportunistic Safety Outside the Maximal Controlled Invariant Set

Constraints are ubiquitous in control tasks for safety-critical systems, such as lane keeping for autonomous vehicles, overload protection in power systems, and obstacle avoidance for mobile robots. The goal of safety control is to synthesize controllers that can guarantee a system operates under its safety constraints indefinitely. Many methods have been developed over the years that can provide such safety guarantees, such as viability theory [16], reference governors [41], safety supervisory control [91, 112], robust control barrier function [53], and Hamilton-Jacobi reachability [19]. A common property shared by those methods is that they all yield an RCIS of the system with respect to the safety constraints.

Recall that there exists a unique *maximal RCIS* that contains all possible RCISs given some safety constraints. Controllers synthesized by the aforementioned methods are defined only if the system initially operates in the maximal RCIS, since otherwise the worst-case disturbance is able to force the system to violate the safety constraints in finite time. However, in practice, the system may be initialized outside the maximal RCIS or exit the maximal RCIS due to unexpected disturbances. In those cases, the system may still operate safely and even re-enter the maximal RCIS, as long as the disturbance is not completely adversarial (or to put it differently, the disturbance behaves collaboratively to some extent). The core question here is how to synthesize controllers that can seize the opportunity to keep the system safe when the disturbance is not entirely adversarial. Apparently the aforementioned methods do not answer this question since they become undefined outside the maximal RCIS.

Similar issues also arise in the field of reactive synthesis for finite transition systems and have been addressed by recent works [6, 37]. The main idea there is that if a winning strategy robust to all disturbances does not exist, one should pick a strategy at least as good as the other strategies in terms of the amount of disturbances it can be robust to. These ideas are also applied in the context of abstraction-based control [106] and finite-horizon constrained optimal control [39]. Inspired by this line of work, in this chapter, we present a novel safety control framework that finds control

inputs that are safe against the largest possible disturbance set. Compared with the existing safety control methods restricted by the maximal RCIS, our method has the following benefits:

- The proposed controller provides the same safety guarantees as the existing methods when the system operates inside the maximal RCIS.
- When outside the maximal RCIS, the proposed controller is robustly safe against the largest amount of disturbance within a predefined template set.
- The proposed controller is well-defined as long as a constraint violation is evitable with some possible collaboration from the disturbance.

In addition, we show that the proposed controller can be synthesized by finding the maximal RCIS (or its inner approximation) of an auxiliary system one dimension higher than the original system, using a technique from [121]. Therefore, toolboxes developed for existing safety control methods can be directly applied to synthesize the proposed controllers. Numerical examples show that our method improve the safety of the system outside the maximal RCIS significantly.

Chapter Overview. In Section 9.1, we provide necessary definitions and the problem statement. A solution to the proposed problem is presented in Section 9.2, followed by numerical examples in Section 9.3 that illustrate the efficacy of our approach. We conclude our work in Section 9.4.

Notation. For two sets X and Y, $f : X \rightrightarrows Y$ denotes a set-valued function from X to Y. For two sets X and Y, their set difference and symmetric set difference are denoted by $X \backslash Y$ and $X \ominus Y := (X \backslash Y) \cup (Y \backslash X)$ respectively. For a function $f : X \to \mathbb{R}$, the infinity norm of f is denoted by $\|f\|_\infty := \sup_{x \in X} |f(x)|$. For two functions $f : X \to Y$ and $g : Y \to Z$, their composition is denoted by $g \circ f$.

9.1 Problem Setup

We consider a discrete-time linear system Σ

$$\Sigma : x^+ = Ax + Bu + Ed, \tag{9.1}$$

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$, and disturbance $d \in D \subseteq \mathbb{R}^l$. The disturbance set D contains all possible disturbances. Let S_{xu} denote the set of desired state-input pairs, which we call the *safe set* of the system. We assume that both D and S_{xu} are convex polytopes, and moreover D is compact.

A *disturbance model* $\Delta : \mathbb{R}^n \rightrightarrows D$ is a function that assigns a subset $\Delta(x)$ of D to each $x \in \mathbb{R}^n$. Given a disturbance model Δ, if the disturbance input satisfies the constraint $d \in \Delta(x)$ at each time

step, we say the disturbance is *generated by* Δ. Given a controller $\mathbf{u} : \mathbb{R}^n \to \mathbb{R}^m$ and a disturbance model Δ, the *k-step forward reachable set* $\mathscr{R}^k_\Sigma(x_0, \mathbf{u}, \Delta)$ from the initial state x_0 is defined recursively by

$$\mathscr{R}^{k+1}_\Sigma(x_0, \mathbf{u}, \Delta) = \{(x^+, \mathbf{u}(x^+)) \mid \exists (x,u) \in \mathscr{R}^k_\Sigma(x_0, \mathbf{u}, \Delta), \\ x^+ \in Ax + Bu + E\Delta(x)\}, \tag{9.2}$$

with $\mathscr{R}^0_\Sigma(x_0, \mathbf{u}, \Delta) = \{(x_0, \mathbf{u}(x_0))\}$. Intuitively, $\mathscr{R}^k_\Sigma(x_0, \mathbf{u}, \Delta)$ contains all possible state-input pairs reached at time k from x_0 by the closed-loop system when the disturbance d is generated by Δ.

9.1.1 Robust Safety Control Framework

Given the system Σ, the safe set S_{xu}, and a disturbance model Δ, the robust safety control problem tries to solve for the set of all the initial states x_0 where there exists $\mathbf{u} : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$\mathscr{R}^k_\Sigma(x_0, \mathbf{u}, \Delta) \subseteq S_{xu},\ \forall k \geq 0. \tag{9.3}$$

Indeed, the maximal set of such initial states is called the *maximal RCIS* C_{max} of Σ with respect to S_{xu} and Δ. There is also an alternative characterization of the maximal RCIS C_{max}: Given the system Σ, the safe set S_{xu}, and a disturbance model Δ, a set $C \subseteq \mathbb{R}^n$ is an RCIS of Σ with respect to S_{xu} and Δ if for all $x \in C$, there exists an input u such that $(x,u) \in S_{xu}$ and $Ax + Bu + E\Delta(x) \subseteq C$. Then, the maximal RCIS C_{max} is the union of all RCIS of Σ with respect to S_{xu} and Δ. Given an RCIS C with respect to S_{xu} and Δ, the *admissible input set* $\mathscr{A}(x, C)$ at a state x is defined as

$$\mathscr{A}(x, C) = \{u \mid (x,u) \in S_{xu},\ Ax + Bu + E\Delta(x) \subseteq C\}. \tag{9.4}$$

A controller $\mathbf{u}$ satisfies the condition in (9.3) for any initial state $x_0 \in C_{max}$ if and only if for all $x \in C_{max}$,

$$\mathbf{u}(x) \in \mathscr{A}(x, C_{max}). \tag{9.5}$$

Most of existing works on safety control consider a special case of the robust safety control problem, which we denote as **Problem 9.1**, where the disturbance model Δ_{all} is such that $\Delta_{all}(x) = D$ for all $x \in \mathbb{R}^n$. We denote the corresponding maximal RCIS as $C_{max,1}$ [20]. As an application of (9.5), if a reference controller $\mathbf{u}_{ref}$ is given, a robust safety supervisor $\mathbf{u}$ that satisfies (9.3) with respect to

Δ_{all} for all $x_0 \in C_{max,1}$ can be synthesized by minimally modifying the reference controller,

$$\mathbf{u}(x) = \min_{u \in \mathscr{A}(x, C_{max,1})} \|u - \mathbf{u}_{ref}(x)\|_2^2, \ \forall x \in C_{max,1}. \tag{9.6}$$

Note that the robust safety supervisor in (9.6) is not defined for states outside $C_{max,1}$. In particular, the admissible input set $\mathscr{A}(x, C_{max,1})$ is empty for any $x \notin C_{max,1}$. This is not problematic if the system starts from $C_{max,1}$ and the disturbance is always in D, ensuring that the system stays in $C_{max,1}$ indefinitely. But those assumptions may be unreliable in practice, potentially causing an inadvertent exit from $C_{max,1}$. As a result, the safety control framework described in this subsection is exceedingly susceptible to potential violations of those assumptions.

9.1.2 An Opportunistic Safety Control Problem

Let $\mathscr{D}$ be a collection of Borel subsets of the disturbance set D, with $D \in \mathscr{D}$. We call $\mathscr{D}$ the *disturbance template set*. For a given controller $\mathbf{u}$ and an initial state x_0, let $\mathscr{P}(\mathbf{u}, x_0)$ be the collection of disturbance models $\Delta : \mathbb{R}^n \to \mathscr{D}$ for which the safety specification in (9.3) is satisfied, that is

$$\mathscr{P}(\mathbf{u}, x_0) := \{\Delta : \mathbb{R}^n \to \mathscr{D} \mid \mathscr{R}_\Sigma^k(x, \mathbf{u}, \Delta) \subseteq S_{xu}, \forall k \geq 0\}.$$

We further define $\mathscr{P}(x_0) := \cup_{\mathbf{u}} \mathscr{P}(\mathbf{u}, x_0)$, that is the set of disturbance models a controller can possibly be robust to when the system starts at x_0.

With these new notations, Problem 9.1 can be rephrased as finding $\mathbf{u}$ such that the worst-case disturbance model Δ_{all} is in $\mathscr{P}(\mathbf{u}, x_0)$ for a given x_0. Note that Δ_{all} is the worst-case disturbance model in the sense that if the safety specification in (9.3) is satisfied for Δ_{all}, it is satisfied for any $\Delta : \mathbb{R}^n \to \mathscr{D}$ with respect to the same x_0 and $\mathbf{u}$. Then, it is clear that when Δ_{all} is not contained by $\mathscr{P}(x_0)$ (that is when $x_0 \notin C_{max,1}$), Problem 9.1 has no solutions. Apparently, a better strategy is to find a controller $\mathbf{u}$ robust to the worst-case disturbance model available in $\mathscr{P}(x_0)$ (if Δ_{all} is not)[1]. In this case, we synthesize a controller that is doing its best to keep the system within the safety constraints, as long as $\mathscr{P}(x_0)$ is nonempty.

To formalize this idea, we need to identify the worst-case disturbance model in $\mathscr{P}(x_0)$. Here we use a simple criterion. Let μ be a Borel measure on D. For any disturbance models Δ_1 and Δ_2, we define $\gamma(\Delta_1, \Delta_2)(x) := \sup_{x \in \mathbb{R}^n} |\mu(\Delta_1(x) \ominus \Delta_2(x))|$. This function γ is a pseudometric in the space of disturbance models, measuring the distance between two disturbance models. Then, since we know Δ_{all} is the worst-case disturbance model among all disturbance models, we simply consider a nearest point in $\mathscr{P}(x_0)$ to Δ_{all} with respect to γ as a worst-case disturbance model in $\mathscr{P}(x_0)$. As

[1] Under the partial order given by $\Delta_1 \leq \Delta_2$ iff $\Delta_1(x) \subseteq \Delta_2(x), \forall x \in \mathbb{R}^n$, it can be shown that Δ_{all} is the unique maximal element in the set of disturbance models, and $\mathscr{P}(\mathbf{u}, x)$ is a lower set with respect to this partial order. Thus, being safe against one disturbance model $\Delta \in \mathscr{P}(x_0)$ implies being safe against all disturbance models less than Δ.

a result, finding a controller robust to a worst-case disturbance model in $\mathscr{P}(x_0)$ is equivalent to finding $\mathbf{u}$ that minimizes the distance $\gamma(\mathscr{P}(\mathbf{u},x_0),\Delta_{all}) := \inf_{\Delta\in\mathscr{P}(\mathbf{u},x_0)}\gamma(\Delta,\Delta_{all})$ between the set $\mathscr{P}(\mathbf{u},x_0)$ and Δ_{all} (by default $\gamma(\mathscr{P}(\mathbf{u},x_0),\Delta_{all}) = +\infty$ if $\mathscr{P}(\mathbf{u},x_0)$ is empty). Based on the above discussion, we pose an opportunistic safety control problem.

Problem 9.2. *Given the system Σ with its safe set S_{xu}, synthesize a controller $\mathbf{u}^* : \mathbb{R}^n \to \mathbb{R}^m$ such that*

(i) $\Delta_{all} \in \mathscr{P}(\mathbf{u}^*,x)$ for $x \in C_{max,1}$;

(ii) $\mathbf{u}^*$ minimizes[2] the distance between $\mathscr{P}(\cdot,x)$ and Δ_{all} with respect to the pseudometric γ for $x \notin C_{max,1}$.

Point (i) above assures that any solution $\mathbf{u}^*$ to Problem 9.2 provides safety guarantees as strong as that to Problem 9.1 when the system operates in the maximal RCIS $C_{max,1}$. When outside $C_{max,1}$, point (ii) assures that $\mathbf{u}^*$ provides extra robustness guarantees compared with solutions to Problem 9.1. Besides, if $C_{max,1}$ is empty, Problem 9.1 has no solution, but a solution $\mathbf{u}^*$ to Problem 9.2 may still exist.

Remark 9.1. The conservativeness and computational tractability of the solutions $\mathbf{u}^*$ to Problem 9.2 depend on the measure μ over D and the disturbance template set $\mathscr{D}$. A trivial choice is $\mathscr{D} = \{D\}$, under which Problem 9.2 degrades to Problem 9.1 .

9.2 Construction of $\mathbf{u}^*$

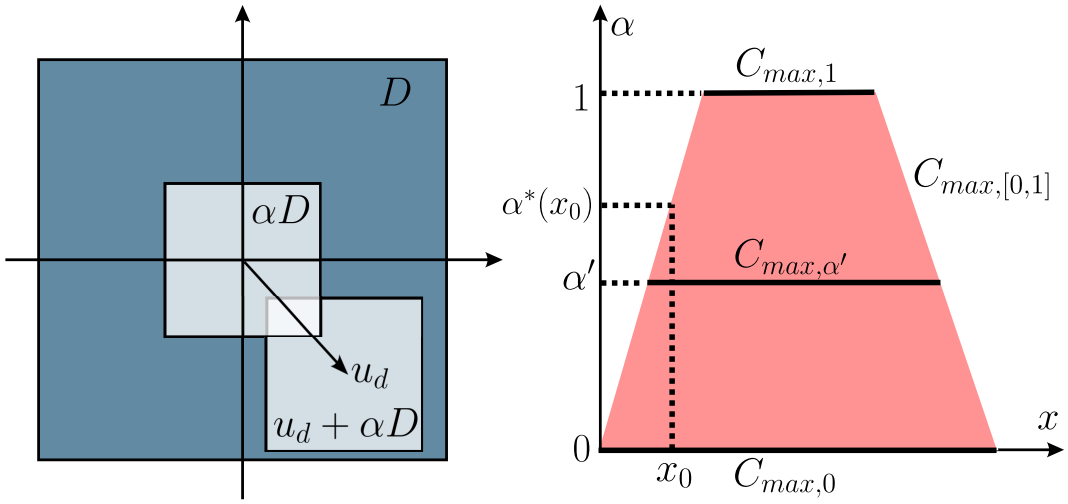
In this section, we show how to construct a solution $\mathbf{u}^*$ to Problem 9.2. As noted in Remark 9.1, we need to first specify the measure μ and the disturbance template set $\mathscr{D}$. We choose the measure μ to be the Lebesgue measure on $\mathbb{R}^l$ but restrict it to D. We assume that $\mu(D) > 0$ [3]. The disturbance template set is chosen to be

$$\mathscr{D} = \{u_d + \alpha D \mid u_d \in (1-\alpha)D,\ \alpha \in [0,1]\}. \tag{9.7}$$

That is, the disturbance template set $\mathscr{D}$ contains all the subsets of D that have the same shape as D. This collection of disturbance sets is rich enough since it contains uncountably many subsets of D scaled to different sizes and positioned at various places, as demonstrated in Fig. 9.1a. At the same time, $\mathscr{D}$ is simple enough for constructing $\mathbf{u}^*$, which is shown next.

[2]Point (ii) does not necessarily imply point (i) since $\gamma(\mathscr{P}(\mathbf{u},x),\Delta_{all}) = 0$ does not imply $\Delta_{all} \in \mathscr{P}(\mathbf{u},x)$ (unless $\mathscr{P}(\mathbf{u},x)$ is closed).

[3]Otherwise, D lies in a subspace of $\mathbb{R}^l$, which implies we should lower l.



(a) $\mathscr{D}$ and $u_d + \alpha D$ (b) $C_{max,\alpha}$ and $C_{max,[0,1]}$

Figure 9.1: Demonstration of the disturbance template set $\mathscr{D}$ in (9.7) and the maximal RCIS $C_{max,\alpha}$ of Σ_α.

For each $\alpha \in [0,1]$, we define an auxiliary system Σ_α

$$\Sigma_\alpha : \; x^+ = Ax + Bu + E(u_d + d), \tag{9.8}$$

with A, B, and E the same as in (9.1), and u_d, $d \in \mathbb{R}^l$. In addition to u, we introduce a new control input u_d. The maximal RCIS of Σ_α with respect to the safe set $S_{xu,\alpha} := S_{xu} \times (1-\alpha)D$ and the disturbance model $\Delta_\alpha := \alpha\Delta_{all}$ is denoted by $C_{max,\alpha}$.

Intuitively, in Σ_α, we split the disturbance input in Σ into two parts, namely that $u_d \in (1-\alpha)D$ and $d \in \alpha D$, and turn u_d into a control input. When $\alpha = 1$, $C_{max,\alpha}$ is just the maximal RCIS $C_{max,1}$ of Σ with respect to S_{xu} and Δ_{all} defined in Section 9.1.1. As α goes to 0, Σ_α has more control power and less uncertainty, and thus $C_{max,\alpha}$ monotonically expands as α goes to 0, as demonstrated in Fig. 9.1b . When $\alpha = 0$, we have full control of the disturbances in D. Hence for any initial state x_0 not in $C_{max,0}$, we cannot find a controller and a disturbance model such that (9.3) is satisfied. In other words, $\mathscr{P}(x_0)$ is empty if and only if $x \notin C_{max,0}$. The following theorem draws a connection between $C_{max,\alpha}$ and solutions $\mathbf{u}^*$ to Problem 9.2.

Theorem 9.1. *For any state $x \in C_{max,0}$, let $\alpha^*(x)$ be the maximal $\alpha \in [0,1]$ such that $x \in C_{max,\alpha}$. Then, a controller $\mathbf{u}^*$ is a solution to Problem 9.2 if and only if for all $x \in C_{max,0}$, there exists $u_d \in \mathbb{R}^l$ such that*

$$(\mathbf{u}^*(x), u_d) \in \mathscr{A}(x, C_{max,\alpha^*(x)}), \tag{9.9}$$

where $\mathscr{A}(x, C_{max,\alpha^*(x)})$ *is the admissible input set of the system* Σ_α *with respect to* $S_{xu,\alpha}$ *and* Δ_α*, with* $\alpha = \alpha^*(x)$*. In addition, the distance between* $\mathscr{P}(\mathbf{u}^*, x)$ *and* Δ_{all} *satisfies*

$$\gamma(\mathscr{P}(\mathbf{u}^*, x), \Delta_{all}) = \begin{cases} (1-\alpha^*(x)^l)\mu(D) & x \in C_{max,0}, \\ +\infty & o.w. \end{cases} \tag{9.10}$$

The proof of Theorem 9.1 is in the appendix. Intuitively, if a state x is in $C_{max,\alpha}$, we can find $\mathbf{u} : \mathbb{R}^n \to \mathbb{R}^m$ and $\mathbf{u}_d : \mathbb{R}^n \to \mathbb{R}^l$ such that the disturbance model $\mathbf{u}_d + \Delta_\alpha$ is in $\mathscr{P}(\mathbf{u}, x)$. It can be shown that by taking $\alpha = \alpha^*(x)$, $\mathbf{u}_d + \Delta_\alpha$ is actually a worst-case disturbance model in $\mathscr{P}(x_0)$ and is contained by $\mathscr{P}(\mathbf{u}^*, x)$ for any $\mathbf{u}^*$ satisfying (9.9). Furthermore, when $\alpha^*(x) = 1$, $\mathbf{u}_d + \Delta_\alpha = \Delta_{all}$. Thus, both points in Problem 9.2 are fulfilled by $\mathbf{u}^*$. Applying Theorem 9.1, given a reference controller $\mathbf{u}_{ref}$, we propose an opportunistic safety supervisor

$$\mathbf{u}(x) = \min_{(u,u_d) \in \mathscr{A}(x, C_{max,\alpha^*(x)})} \|u - \mathbf{u}_{ref}(x)\|_2^2 \tag{9.11}$$

This opportunistic safety supervisor is defined over $C_{max,0}$, larger than the domain $C_{max,1}$ of the robust safety supervisor in (9.6). Recall that when x_0 is not in $C_{max,0}$, it becomes inevitable to violate the safety constraints no matter what the controller and the disturbance do. The opportunistic safety supervisor becomes undefined only in this extreme case.

Remark 9.2. Our results also hold for Problem 9.2 with discretized α: Let α_i for i from 0 to N be an increasing sequence of scalars such that $0 = \alpha_0 < \alpha_1 < \cdots < \alpha_N = 1$. Let

$$\mathscr{D} = \{u_d + \alpha_i D \mid u_d \in (1-\alpha_i)D, i \in \{0, \cdots, N\}\}. \tag{9.12}$$

We redefine $\alpha^*(x)$ to be the maximal α_i such that $x \in C_{max,\alpha_i}$. Theorem 9.1 with this redefined $\alpha^*(x)$ holds for the disturbance set $\mathscr{D}$ in (9.12). The only benefit for using $\mathscr{D}$ in (9.12) instead of (9.7) is that to construct $\mathbf{u}^*$, we only need $C_{max,\alpha}$ for finitely many α (versus a continuum of α in $[0,1]$).

9.2.1 The One-shot Computation of $C_{max,\alpha}$

In this subsection, we show how to compute $C_{max,\alpha}$ for all $\alpha \in [0,1]$ in one shot. Consider a new auxiliary system $\Sigma_{[0,1]}$

$$\Sigma_{[0,1]} : \begin{bmatrix} x^+ \\ \alpha^+ \end{bmatrix} = \begin{bmatrix} Ax + Bu + E(u_d + d) \\ \alpha \end{bmatrix}, \tag{9.13}$$

where we introduce a new state $\alpha \in [0,1]$ and a new control input $u_d \in \mathbb{R}^l$. Define the safe set $S_{xu,[0,1]}$ of $\Sigma_{[0,1]}$ by

$$S_{xu,[0,1]} = \{(x, \alpha, u, u_d) \,|(x, u, u_d) \in S_{xu,\alpha}, \alpha \in [0,1]\}. \tag{9.14}$$

Let $\Delta_{[0,1]}$ be the disturbance model such that $\Delta_{[0,1]}(x,\alpha) = \alpha D$ for all $(x,\alpha) \in \mathbb{R}^n \times [0,1]$. We denote the maximal RCIS of $\Sigma_{[0,1]}$ with respect to $S_{xu,[0,1]}$ and $\Delta_{[0,1]}$ by $C_{max,[0,1]}$. Since S_{xu} and D are both polytopes, it can be shown that $S_{xu,[0,1]}$ is a polytope and the maximal RCIS $C_{max,[0,1]}$ can be approximated by the standard iterative method [20]. The implementation details of this method are outlined in [91, 121]. Once we have $C_{max,[0,1]}$, $C_{max,\alpha'}$ is just equal to the slice of $C_{max,[0,1]}$ through $\alpha = \alpha'$, for any $\alpha' \in [0,1]$, as shown in Fig. 9.1b. Furthermore, given $C_{max,[0,1]}$ and x, the value $\alpha^*(x)$ can be easily obtained by solving a linear program:

$$\begin{aligned} \alpha^*(x) = \max_{\alpha \in [0,1]} \; & \alpha \\ \text{s.t. } & (x,\alpha) \in C_{max,[0,1]}. \end{aligned} \tag{9.15}$$

Remark 9.3 (Computational cost). Compared with the robust safety control framework, our method needs to compute the maximal RCIS $C_{max,[0,1]}$ for a system with one additional state $\alpha \in \mathbb{R}$ and one extra control input $u_d \in \mathbb{R}^l$ (cf., (9.13) vs. (9.1)) and thus has a higher offline computational cost.

At runtime, given the current state x and $C_{max,[0,1]}$, we first solve the linear program in (9.15) to check if $x \in C_{max,0}$ and find $\alpha^*(x)$, and then solve the quadratic program in (9.11). For a comparison, the robust safety control framework solves one linear program to check if $x \in C_{max,1}$ and then solve the quadratic program in (9.6). The runtime computational cost of the two frameworks should be similar.

Remark 9.4. If the maximal RCIS $C_{max,[0,1]}$ cannot be computed exactly, one can use any controlled invariant inner approximation of $C_{max,[0,1]}$ in (9.11), with the cost of extra conservativeness.

9.3 Numerical Examples

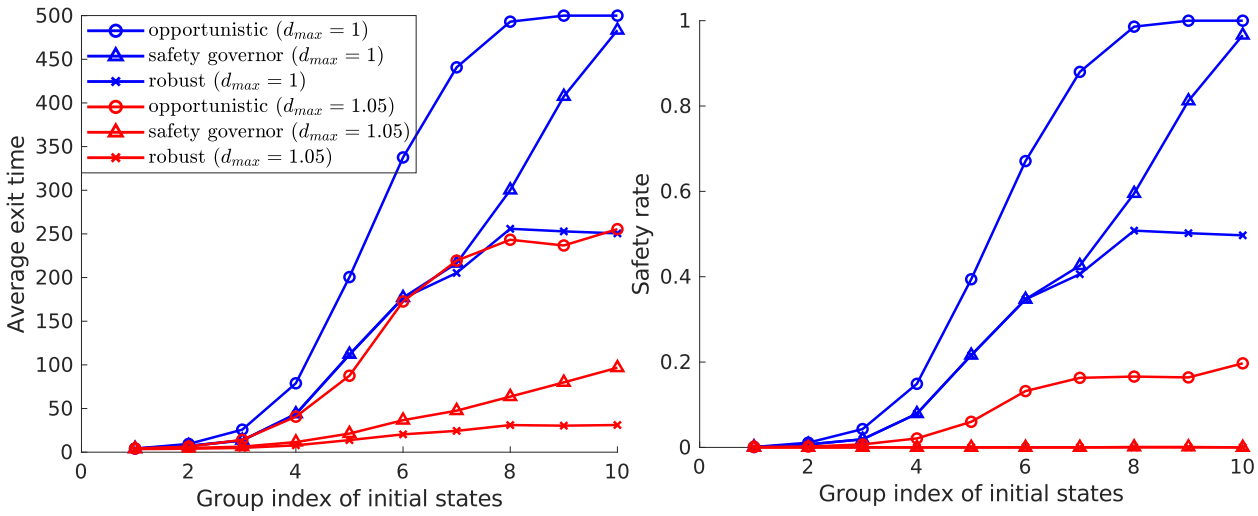
The maximal RCISs $C_{max,[0,1]}$ in the examples are computed using MPT3 [48] equipped with GUROBI [44] in MATLAB. The code and video can be accessed from https://ozay-group.github.io/OppSafe/.

9.3.1 Adaptive Cruise Control

We consider the car-following example in [69]. The goal is to maintain the relative distance Δs and the relative velocity Δv between the ego vehicle and the front vehicle within a safe range. The system is modeled by a discretized double integrator with states $x = (\Delta s, \Delta v)$. The model parameters can be found in [69]. The control input and the disturbance are the acceleration u of the ego vehicle and the acceleration $d \in [-d_{max}, d_{max}]$ of the front vehicle respectively. The safe set is given by $|\Delta s - 15| \leq 5$, $|\Delta v| \leq 5$, and $|u| \leq 2$.

The reference controller $\mathbf{u}_{ref} = 0.2842\Delta s + 0.8056\Delta v$, with a saturation limit at ± 2. We implemented the robust and the opportunistic safety supervisors in (9.6) and (9.11) and the safety protection and extension governor outlined in Section V of [69], assuming $d_{max} = 1$. To evaluate those three safety supervisors at states with different values of $\alpha^*(x)$, we generated 10 groups $\mathscr{X}_{0,i}$ of initial states, where $\mathscr{X}_{0,i}$ contains 1000 states uniformly sampled in $C_{max,0.1*(i-1)} \backslash C_{max,0.1*i}$ for i from 1 to 10. That is, each $x_0 \in \mathscr{X}_{0,i}$ has $\alpha^*(x_0)$ between $0.1(i-1)$ and $0.1i$. Note that $\mathscr{X}_{0,i}$ is disjoint from the maximal RCIS $C_{max,1}$ for all i, since we want to evaluate how the controllers perform outside the maximal RCIS.

For each initial state x_0 in $\mathscr{X}_{0,i}$, we generate a random disturbance sequence from a uniform distribution in the interval $[-d_{max}, d_{max}]$ and then run simulations for each safety supervisor for 500 steps. During the simulation, if a safety supervisor becomes undefined, we switch to the reference controller. Thus, for each group index i and d_{max}, we have 1000 trajectories starting from $\mathscr{X}_{0,i}$ under each safety supervisor. We evaluate the performance with two metrics: the *average exit time* the system first exits the safe set (taken to be 500 when the system never exits S_{xu}), and the *safety rate*, the ratio of trajectories remaining in S_{xu} through the entire simulation period out of a total of 1000 trajectories. The average exit time and the safety rates of the three safety supervisors for $d_{max} = 1$ and 1.05 are shown in Fig. 9.2. First note that both metrics of all the safety supervisors grow with the group index i. This is expected since the initial states with a higher value of $\alpha^*(x)$ have worst-case disturbances in $\mathscr{P}(x_0)$ closer to Δ_{all} and thus are easier to be kept within the safe set. Comparing curves in different colors in Fig. 9.2, we observe that the performance of the safety supervisors degrade as the disturbances is sampled in a range larger than the assumed one. Finally, comparing curves in the same color, we observe that the proposed safety supervisor outperforms the safety supervisors in (9.6) and in [69] in both metrics for all groups of initial states and all d_{max}. In particular, when unexpected disturbances appear (by increasing d_{max} from 1 to 1.05), the proposed safety supervisor has much larger average exit time than the other two, as shown in Fig.9.2a, and is the only one among the three that has nonzero safety rates, as shown in Fig. 9.2b, showing that the proposed method enhances the safety of the system significantly when the system operates outside the maximal RCIS. Similar results are observed when we repeat this experiment with disturbances generated by nonuniform distributions, suggesting that our approach works well across



(a) Average exit time

(b) Safety rate

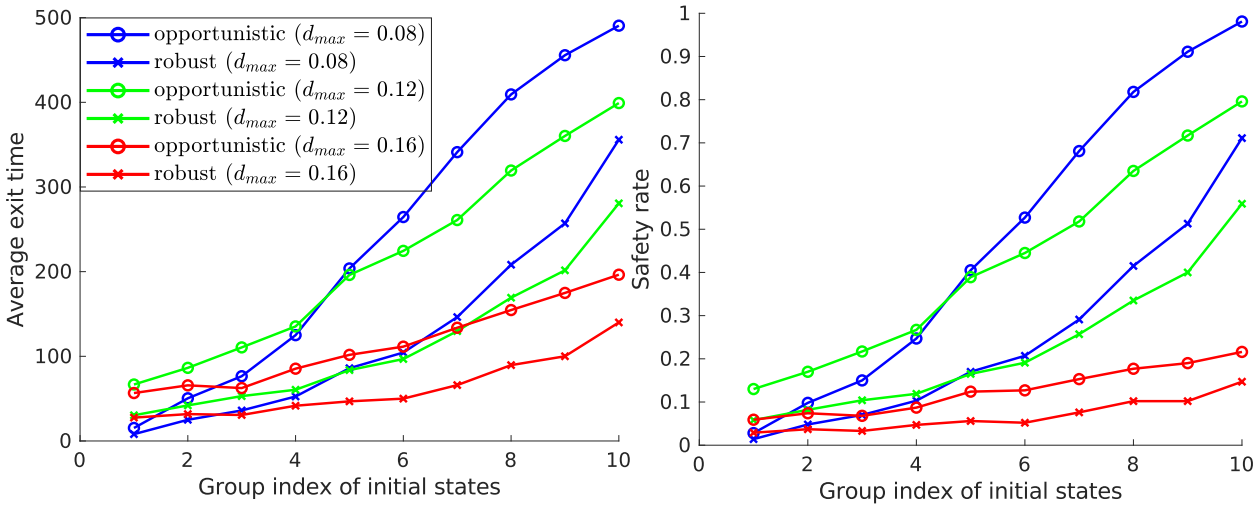
Figure 9.2: The average exit time and safety rates of the safety supervisors in (9.6) (robust) and (9.11) (opportunistic), and the safety governor in [69] in the adaptive cruise control example.

different disturbance distributions. We omit the results for these other experiments but the code for reproducing them is available in our code repository.

9.3.2 Lane Keeping Control

We consider a highway driving scenario where we want to keep the lateral position of a vehicle within given lane boundaries. We use the 4-dimensional linearized bicycle model in [112] with respect to the constant longitudinal velocity $30m/s$, discretized with time step $0.1s$. The states are the lateral displacement y, the lateral velocity v, the yaw angle $\Delta\Psi$, and the yaw rate r. The control input is the steering angle u. The safe set is given by constraints $|y| \leq 0.9$, $|v| \leq 1.2$, $|\Delta\Psi| \leq 0.05$, $|r| \leq 0.3$, and $|u| \leq \pi/2$. The disturbance of the system is the road curvature d with $|d| \leq d_{max}$.

The reference controller $\mathbf{u}_{ref}$ is $\mathbf{u}_{ref} = -Kx$ subject to a saturation limit at $\pm\pi/2$, where K is determined through solving an LQR problem (with $Q = I$ and $R = 0$). Then, we implement the proposed safety supervisor in (9.11) and the robust safety supervisor in (9.6), assuming $d_{max} = 0.08$. For this example, our implementation of the approach in Section V of [69] fails to find a nonempty RCIS and thus its simulation results are omitted. We assess the safety supervisors in the same manner as in Section 9.3.1. Fig. 9.3 illustrates the average exit time and the safety rate for both safety supervisors under simulations with $d_{max} = 0.08$, 0.12, and 0.16. Similar to the previous example, the performance of the safety supervisors is improved as the initial states have a higher value of $\alpha^*(x)$, and degrades as the disturbance range used in the simulation exceeds that used in control synthesis. Comparing curves in the same color in Fig. 9.3 , the proposed safety supervisor



(a) Average exit time

(b) Safety rate

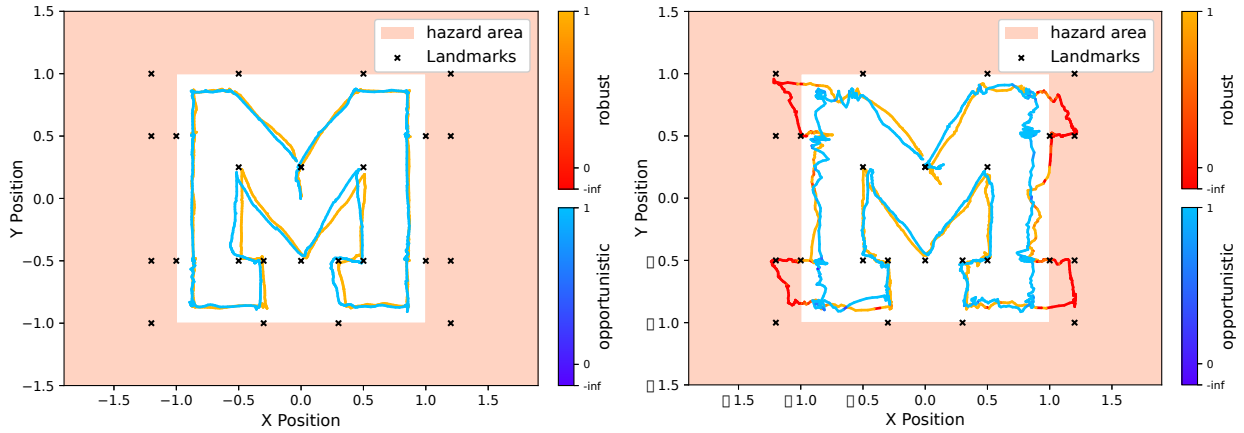
Figure 9.3: The average exit time and the safety rate of the robust safety supervisor in (9.6) and the opportunistic safety supervisor in (9.11) in the lane keeping example.

consistently outperforms the robust safety supervisor across all groups of initial states and all d_{max}. Notably, the performance of our approach at $d_{max} = 0.12$ (50% larger than the assumed d_{max}) is even better than the performance of the robust safety supervisor at $d_{max} = 0.08$, highlighting its enhanced safety and resilience to unexpected disturbances when the system operates beyond the maximal RCIS $C_{max,1}$.

9.3.3 Safe Tracking for Aerial Vehicle

We tested our approach on the drone platform Crazyflie 2.1 in a task of cruising around designated waypoints in the horizontal plane while avoiding entering hazardous region (red region in Fig. 9.4). We use a built-in controller to keep the altitude of the drone constant, and then control its horizontal motion by sending the reference velocities u_x and u_y in the x, y axes to a lower-level controller in the period of $0.1s$. Validated by the flight data, the drone dynamics in x and y axes under the lower-level controller are decoupled, homogeneous, and linear. Thus, we model the dynamics in x and y by two identical 3-dimensional linear systems with states $s_x = (x, v_x, u_{x,-1})$ (or $s_y = (y, v_y, u_{y,-1})$, respectively), where x and v_x are the position and velocity in x axis, and $u_{x,-1}$ is the previous reference velocity u_x. The system matrices are learned from data via least square with the disturbance set the convex hull of the prediction error. The safe set of Σ_x is given by constraints $|x| \leq 1$, $|v_x| \leq 1$, $|u_x| \leq 1$, and $|u_{x,-1} - u_x| \leq 0.5$. The setup for the system in y axis is the same.

We synthesize one reference tracking controller for each subsystem in form of $\mathbf{u}_{ref,*} = -K(s_* - s_{ref,*})$ ($*$ is a placeholder for x or y) subject to a saturation limit of ± 1, where K is determined



(a) No measurement noise

(b) With measurement noise

Figure 9.4: The drone trajectories in x-y plane under the safety supervisors in (9.6) (yellow-red) and (9.11) (blue-dark blue). The color of the curves reflects the value of $\alpha^*(x)$ along the trajectories.

through solving a LQR problem (with $Q = 3I$ and $R = I$). We then implement the safety supervisors in (9.6) and (9.11) to supervise $\mathbf{u}_{ref,*}$. For the experiments, we pick 24 waypoints to form a big "M", as shown by the checkmarks in Figure 9.4. Part of the waypoints is picked outside the safe region such that the reference controller without any supervision would steer the drone to the unsafe region. During the experiments, we switch to the reference controller whenever the safety supervisor is undefined. Since the drone is initialized within $C_{max,1}$, both safety supervisors are able to maintain the drone within the safe region, as shown by Fig. 9.4a. To make this task more challenging, we repeat this experiment with the state measurements corrupted by an additional Gaussian noise with standard deviation 0.05. Subject to this unexpected measurement noise, our opportunistic safety supervisor still successfully keeps the drone within the safe region, while the robust safety supervisor in (9.6) fails as shown in Fig. 9.4b.

9.4 Conclusion

In this chapter, we present an opportunistic safety control framework that extends the domain of safety controllers beyond the maximal RCIS. This is achieved by designing a controller that is robustly safe against as much disturbance as possible. Our approach can be trivially extended for nonlinear systems, which we consider in future. We also want to extend these results to probabilistic settings, by using a given or learned probability measure μ instead of the Lebesgue measure used in this chapter.

CHAPTER 10

Summary and Outlook

10.1 Summary

This dissertation addresses challenging safety control problems for discrete-time systems, by leveraging inherent structures in system dynamics, controllers, and disturbances.

The first part of the dissertation develops scalable safety controller synthesis algorithms. It first analyzes the convergence properties of the inside-out algorithm for computing inner approximations of the maximal RCIS. Then, efficient methods for computing implicit RCISs are introduced, making use of an eventually periodic or automaton-based structure in control. Compared with existing methods of computing implicit RCIS, our approach does not need a "seed" RCIS, and provide weak completeness and performance guarantees. Lastly, we reveal a key structural property of the maximal RCIS of input-delayed systems. Based on this property, we propose an efficient method for computing the maximal RCIS of input-delayed systems that scales well with delay time.

The second part of this dissertation focuses on reducing conservativeness in safety control by utilizing the structure of disturbance, particularly preview information on disturbance. We introduce the concept of safety regret, which measures the incremental value of preview information in safety control. Then we prove the exponential convergence of safety regret with preview horizon for discrete-time linear systems and provides numerical algorithms to estimate the convergence rate. For three classes of systems with preview, we develop efficient methods for computing the maximal RCIS for systems with preview, by exploiting the structural properties in the system model. Finally, we propose the opportunistic safety control framework that works both inside and outside the maximal RCIS without relying on preview information. By steering the system against the worst-case disturbance that can be tolerant at each state, this opportunistic safety control framework outperforms the existing safety control frameworks in experiments, and shows great resilience to unexpected disturbance.

10.2 Future Work

In this section, we discuss a few research directions that extend the works presented in this dissertation.

10.2.1 Safety Control for Systems with Uncertain Preview

In Section 8.3, we study the safety control problem for systems with uncertain preview. While we have shown that the maximal winning set for such a problem can be derived from the maximal RCIS of a finite-dimensional auxiliary system, the computational cost for approximating this maximal RCIS can be high due to the nonlinearity and the high dimensionality of the auxiliary system. It would be of interest to develop approaches that can efficiently approximate the maximal winning set, and moreover, synthesize safe controllers for systems with uncertain preview.

10.2.2 Opportunistic Safety for Nonlinear Systems

In Chapter 9, we propose the opportunistic safety control framework for discrete-time linear systems. It is straightforward to extend this approach for nonlinear systems either in continuous time or discrete time. The main challenge in these extensions is how to approximate the maximal RCIS $C_{max,[0,1]}$ of the auxiliary system $\Sigma_{[0,1]}$ in (9.13), which would be a nonlinear system with state-dependent disturbance constraints. One potential solution is to use existing toolboxes such as helperOC [19] for continuous-time systems and ROCS [73] for discrete-time systems. Furthermore, our current framework does not utilize any statistics regarding the disturbance. Another interesting research direction is to investigate how to incorporate disturbance statistics into our framework when they are available.

10.2.3 Safety Control for Nonlinear Systems Based on Koopman Operator Theory

The safety control methods presented in this dissertation are mainly developed for discrete-time linear systems, limiting their applicability. Recent results in applied Koopman operator theory have shown that many nonlinear control systems can be approximated well by high-dimensional linear systems via a nonlinear lifting function [60, 61, 87]. This high-dimensional linear representation of a nonlinear system allows one to use methods developed for linear systems, such as LQR or linear MPC, to control nonlinear systems. However, there are few works that combine Koopman operation theory with safety control. The main challenge is that many safety control methods are developed for linear systems with polytopic constraints, but the lifting function that maps states

of the nonlinear system to states of its linear representation is typically nonlinear and does not preserve linear inequality constraints. Our recent work [18] shows that if the lifting function has a linear left inverse, we can map polytopic constraints on the nonlinear system to some polytopic constraints on its linear representation, enabling efficient BRS computation for nonlinear systems. However, experiments in [18] also show that the conservativeness of the resulting BRS highly depends on the choice of the lifting function. One open research problem is how to design or learn a lifting function that can systematically reduce the conservativeness in BRS computation. Another interesting research direction is to investigate if we can compute RCIS for nonlinear systems using similar ideas.

APPENDIX A

Complementary Materials for Chapter 3

A.1 On the existence of stationary points

The work [27, Theorem 12] considers disturbance-free systems with decoupled safe set $S_{xu} = X \times U$ for a convex compact $X \subseteq \mathbb{R}^n$ and a closed convex $U \subseteq \mathbb{R}^m$. To apply [27, Theorem 12] to our setting, given the system Σ of the form (3.1), we construct the disturbance-free system Σ':

$$\begin{bmatrix} x(t+1) \\ u(t+1) \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} v_1(t) + \begin{bmatrix} E \\ 0 \end{bmatrix} v_2(t), \tag{A.1}$$

where (x,u) is the state and v_1, v_2 are two inputs of the system Σ'. Given a compact convex RCIS C_0 of Σ in S_{xu}, construct the set C_0' as

$$C_0' = \{(x,u) \in S_{xu} \mid x \in C_0, \exists d \in D, Ax + Bu + Ed \in C_0\}. \tag{A.2}$$

It is easy to check that C_0' is a compact convex controlled invariant set[27] of the system Σ' in the decoupled safe set $S_{xu} \times \mathbb{R}^m \times D$. Then, by [27, Theorem 12], there exists $(x_e, u_e) \in C_0'$, $v_{1,e} \in \mathbb{R}^m$ and $v_{2,e} \in D$ such that

$$\begin{bmatrix} x_e \\ u_e \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ u_e \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} v_{1,e} + \begin{bmatrix} E \\ 0 \end{bmatrix} v_{2,e}. \tag{A.3}$$

Define $d_e = v_{2,e}$. By (A.3), $x_e = Ax_e + Bu_e + Ed_e$. By construction of C_0', we have $x_e \in C_0$ and $(x_e, u_e) \in S_{xu}$.

A.2 Proofs of Theorems 3.2-3.5

The following lemmas reveal properties of the BRS, which are crucial for proving Theorems 3.2-3.5.

Lemma A.1. For a linear system Σ in form of (3.1),

$$Pre_{\Sigma}(X_1+X_2, S_{xu,1}+S_{xu,2}, D_1+D_2) \supseteq Pre_{\Sigma}(X_1, S_{xu,1}, D_1) + Pre_{\Sigma}(X_2, S_{xu,2}, D_2). \tag{A.4}$$

Proof. Let $x_i \in Pre_{\Sigma}(X_i, S_{xu,i}, D_i)$ for $i = 1,2$. Then, there exists u_i such that $(x_i, u_i) \in S_{xu,i}$ and $Ax_i + Bu_i + ED_i \subseteq X_i$, for $i = 1,2$. Thus, we have

$$(x_1, u_1) + (x_2, u_2) \in S_{xu,1} + S_{xu,2} \tag{A.5}$$

$$A(x_1+x_2) + B(u_1+u_2) + ED_1 + ED_2 \subseteq X_1 + X_2. \tag{A.6}$$

Thus, $x_1 + x_2 \in Pre_{\Sigma}(X_1+X_2, S_{xu,1}+S_{xu,2}, D_1+D_2)$. □

Lemma A.2. For a linear system Σ in form of (3.1), if X, S_{xu} and D are convex, then for any $a \in [0,1]$, $b \in [0,1]$ and $k \geq 1$, we have

$$Pre^k_{\Sigma}(X, S_{xu}, D) \supseteq Pre^k_{\Sigma}(aX, bS_{xu}, bD) + Pre^k_{\Sigma}((1-a)X, (1-b)S_{xu}, (1-b)D).$$

Proof. For any convex set C, $C = \alpha C + (1-\alpha)C$ for any $\alpha \in [0,1]$. Thus, by Lemma A.1,

$$Pre_{\Sigma}(X, S_{xu}, D) \supseteq Pre_{\Sigma}(aX, bS_{xu}, bD) + Pre_{\Sigma}((1-a)X, (1-b)S_{xu}, (1-b)D).$$

Now assume that (A.7) holds for $k = N-1$. By applying Lemma A.1 again, it can be easily proven that (A.7) holds for $k = N$. Thus, by induction that (A.7) holds for all k. □

Lemma A.3. For a linear system Σ in form of (3.1) and any $a \geq 0$,

$$Pre_{\Sigma}(aX, aS_{xu}, aD) = aPre_{\Sigma}(X, S_{xu}, D). \tag{A.7}$$

Proof. Let $x \in Pre_{\Sigma}(X, S_{xu}, D)$. Then, there exists u such that $(x,u) \in S_{xu}$ and $Ax + Bu + ED \subseteq X$. Thus, $(ax, au) \in aS_{xu}$ and $Aax + Bau + aED \subseteq aX$. That is, $ax \in Pre_{\Sigma}(aX, aS_{xu}, aD)$. Therefore, $aPre_{\Sigma}(X, S_{xu}, D) \subseteq Pre_{\Sigma}(aX, aS_{xu}, aD)$.

Next, let $ax \in Pre_{\Sigma}(aX, aS_{xu}, aD)$. Then, there exists au such that $(ax, au) \in aS_{xu}$ and $Aax + Bau + aED \subseteq aX$. Thus, $(x,u) \in S_{xu}$ and $Ax + Bu + ED \subseteq X$. That is, $x \in Pre_{\Sigma}(X, S_{xu}, D)$. Therefore, $Pre_{\Sigma}(aX, aS_{xu}, aD) \subseteq aPre_{\Sigma}(X, S_{xu}, D)$. □

Lemma A.4. Let $\{C_i\}_{i=0}^{\infty}$ be an expanding family of nonempty compact convex set. That is, $C_i \subseteq C_j$ for any i, j such that $i \leq j$. Suppose that $int(C_i) \neq \emptyset$ for all $i \geq i_0$ for some $i_0 \geq 0$, and $C_{\infty} = \cup_{i=0}^{\infty} C_i$ is bounded. Then, for any closed set $U \subseteq int(C_{\infty})$, there exists i such that $C_i \supseteq U$.

Proof. Without loss of generality, we assume $i_0 = 0$. Since C_i is convex and compact, it is easy to check that $cl(int(C_i)) = C_i$.

Define $\tilde{C}_\infty = \lim_{k\to\infty} \cup_{i=0}^{k} int(C_i)$. Thus, $\tilde{C}_\infty$ is an open convex set. Note that

$$cl(C_\infty) \supseteq cl(\tilde{C}_\infty) \supseteq cl(int(C_i)) = C_i, \forall i \geq 0. \tag{A.8}$$

Thus,

$$cl(C_\infty) \supseteq cl(\tilde{C}_\infty) \supseteq \lim_{k\to\infty} \cup_{i=0}^{k} C_i = C_\infty. \tag{A.9}$$

Thus, $cl(C_\infty)$ is the closure of $\tilde{C}_\infty$. By convexity of C_∞, $\tilde{C}_\infty$ is the interior of $cl(C_\infty)$, and also the interior of C_∞.

Next, let U be a closed subset of $int(C_\infty) = \tilde{C}_\infty$. Since C_∞ is bounded, U is compact. Also, since $U \subseteq \tilde{C}_\infty = \cup_{i=0}^{\infty} int(C_i)$, $\{int(C_i)\}_{i=0}^{\infty}$ forms a open cover of U. By compactness, there exists a finite subcover $\{C_{i_k}\}_{k=0}^{K}$ of U for some $K \geq 0$. Then, $U \subseteq int(C_{i_K})$. □

Lemma A.5. Let X be a nonempty compact convex set in $\mathbb{R}^n$. Under Assumption 3.1, $Pre_\Sigma(X, S_{xu}, D)$ is compact and convex.

Proof. A proof for similar results can be found in [22]. We provide a separate proof here for completeness.

Define $C_{xu} = \{(x,u) \in S_{xu} \mid Ax + Bu + ED \subseteq X\}$. By definition of backward reachable set, $Pre_\Sigma(X, S_{xu}, D) = \pi_{[1,n]}(C_{xu})$. For now, we assume that C_{xu} is compact and convex. Since projection $\pi_{[1,n]}(\cdot)$ is continuous and C_{xu} is compact, $Pre_\Sigma(X, S_{xu}, D)$ is compact. Since the projection of a convex set is convex, $Pre_\Sigma(X, S_{xu}, D)$ is convex. It is left to show the convexity and compactness of C_{xu}.

We first show that C_{xu} is convex. Let (x_1, u_1), (x_2, u_2) be two points in C_{xu} and α be a constant in $(0,1)$. Denote $(x,u) = \alpha(x_1,u_1) + (1-\alpha)(x_2,u_2)$. Since S_{xu} is convex, (x,u) is in S_{xu}. For any $d \in D$ and $i \in \{1,2\}$, $x_i^+ = Ax_i + Bu_i + Ed \in X$. Since X is convex, for the same $d \in D$, $Ax + Bu + Ed = \alpha x_1^+ + (1-\alpha)x_2^+$ is in X. Thus, $(x,u) \in C_{xu}$. That is, C_{xu} is convex.

Next, we show that C_{xu} is compact. Let $\{(x_n, u_n)\}_{n=1}^{\infty}$ be an arbitrary convergent sequence in C_{xu}. Suppose that $(x_n, u_n) \to (x,u)$. Since S_{xu} is compact, $(x,u) \in S_{xu}$. For any $d \in D$, $Ax_n + Bu_n + Ed \to Ax + Bu + Ed$. Since X is compact and $Ax_n + Bu_n + Ed \in X$, $Ax + Bu + Ed \in X$. Since $(x,u) \in S_{xu}$ and $Ax + Bu + ED \subseteq X$, $(x,u) \in C_{xu}$. Thus, C_{xu} is closed. Since C_{xu} is closed subset of the compact set S_{xu}, C_{xu} is compact.

□

Lemma A.6. Suppose Assumption 3.1 holds. Let C_0 be a compact convex RCIS of Σ in S_{xu}. Define $C_k = Pre_\Sigma^k(C_0, S_{xu}, D)$ and $C_\infty = \cup_{k=1}^\infty C_k$. Then, C_∞ satisfies the following properties:

(a) $\{C_k\}_{k=1}^\infty$ is an expanding family of compact convex RCISs;

(b) C_∞ is bounded and convex;

(c) If $int(C_\infty)$ is nonempty, then there exists a finite $k \geq 0$ such that $int(C_k)$ is nonempty;

(d) For any compact subset C contained in the interior of C_∞, there exists an $\varepsilon > 0$ such that $C + B_\varepsilon(0) \subseteq C_\infty$.

Proof. We first prove (a): Since C_0 is an RCIS in S_{xu}, by definition, $C_0 \subseteq Pre_\Sigma(C_0, S_{xu}, D) = C_1$. Since C_0 is compact and convex, by Lemma A.5, C_1 is compact and convex. Now suppose that $C_{k-1} \subseteq C_k$, and both C_{k-1} and C_k are compact and convex. Then, $C_k = Pre_\Sigma(C_{k-1}, S_{xu}, D) \subseteq Pre_\Sigma(C_k, S_{xu}, D) = C_{k+1}$. Since C_k is compact and convex, by Lemma A.5, C_{k+1} is compact and convex. By induction, $\{C_k\}_{k=1}^\infty$ is an expanding family of nonempty compact convex sets. Furthermore, since $C_k \subseteq C_{k+1} = Pre_\Sigma(C_k, S_{xu}, D)$, C_k is an RCIS of Σ in S_{xu} for all $k \geq 0$.

(b) Since S_{xu} is bounded and $C_k \subseteq \pi_{[1,n]}(S_{xu})$ for all $k \geq 0$, C_∞ is bounded. Let x_1 and x_2 be two points in C_∞. Since $\{C_k\}_{k=1}^\infty$ is an expanding family of sets and $C_\infty = \cup_{k=1}^\infty C_k$, there exists k_0 such that $x_1, x_2 \in C_{k_0}$. According to (a), C_{k_0} is convex. Thus, any convex combination of x_1 and x_2 is in $C_{k_0} \subseteq C_\infty$. Hence, C_∞ is convex.

(c) Since $int(C_\infty)$ is nonempty, we can fit a small hypercube in the interior of C_∞. By definition of C_∞, each vertex of this hypercube is contained by C_k for some finite k. Since the hypbercube has finitely many vertices, there exists a finite k_0 such that C_{k_0} contains all the vertices of the hypercube. Since C_{k_0} is convex, C_{k_0} contains the hypercube and thus has nonempty interior.

(d) Suppose that for all $\varepsilon > 0$, $C + B_\varepsilon(0) \not\subseteq C_\infty$. Then there exists $x_n \in C$ such that $B_{1/n}(x_n) \not\subseteq C_\infty$. Since C is compact, there exists a convergent subsequence $(x_{n_i})_{i=1}^\infty$ such that $x_{n_i} \to x$ for some $x \in C$. Since C is contained by the interior of C_∞, there exists a constant $\varepsilon' > 0$ such that $B_{\varepsilon'}(x) \subseteq C_\infty$. Since $n_i \to \infty$ and $x_{n_i} \to x$, there exists i' such that $1/n_{i'} < \varepsilon'/2$ and $\|x_{n_{i'}} - x\|_2 < \varepsilon'/2$. Thus, $B_{1/n_{i'}}(x_{n_{i'}}) \subseteq B_{\varepsilon'}(x) \subseteq C_\infty$. However, by construction, $B_{1/n_{i'}}(x_{n_{i'}}) \not\subseteq C_\infty$, contradiction! Thus, there exists $\varepsilon > 0$ such that $C + B_\varepsilon(0) \subseteq C_\infty$. □

Proof of Theorem 3.2. If C_0 is contained in the interior of C_{L_0} for some $L_0 > 0$, it is trivial that C_0 is contained in the interior of C_∞. Now suppose that C_0 is contained in the interior of C_∞. By Lemma A.6 (d), there exists $\varepsilon > 0$ such that $cl(C_0 + B_\varepsilon(0)) \subseteq int(C_\infty)$. Also, by Lemma A.6 (c), there exists k_0 such that $int(C_{k_0})$ is nonempty. Thus by Lemma A.4, there exists k such that $C_k \supseteq cl(C_0 + B_\varepsilon(0))$. For the same k, it is clear that $C_0 \subseteq int(C_k)$. □

Proof of Theorem 3.3. We first show that there exists a constant $\alpha \in (0,1)$ such that $\alpha C_{max} + (1-\alpha)C_0$ is contained by the interior of C_∞. Since C_0 is contained in the interior of C_∞, by Lemma A.6, there exists $\varepsilon > 0$ such that $C_0 + B_\varepsilon(0) \subseteq C_\infty$. Then, $C_0 + B_{\varepsilon/2}(0)$ is in the interior of C_∞. Since S_{xu} is compact, C_{max} is bounded, and thus there exists $\alpha > 0$ such that $\alpha C_{max} \subseteq B_{\varepsilon/2}(0)$. Thus, $C_0 + \alpha C_{max} \subseteq int(C_\infty)$. Since $0 \in C_0$, $(1-\alpha)C_0 + \alpha C_{max} \subseteq C_0 + \alpha C_{max} \subseteq int(C_\infty)$.

Next, it can be easily shown that for any RCIS C in the compact convex safe set S_{xu}, the convex hull of C and the closure of C are also RCIS in S_{xu} (a special case is proven by [32, Proposition 20]). Thus, the maximal RCIS C_{max} is compact and convex. Since both C_{max} and C_0 are compact and convex, $\alpha C_{max} + (1-\alpha)C_0$ is compact and convex. By Lemmas A.4 and A.6, since $\alpha C_{max} + (1-\alpha)C_0$ is a closed subset of $int(C_\infty)$, there exists k_0 such that $\alpha C_{max} + (1-\alpha)C_0 \subseteq C_{k_0}$.

We denote $\overline{C}_0 = \alpha C_{max} + (1-\alpha)C_0$. Since C_{max} and C_0 are RCISs, it is easy to check that $\overline{C}_0$ is also an RCIS in S_{xu}. Define $\overline{C}_k = Pre^k_\Sigma(\overline{C}_0, S_{xu}, D)$. Then, by Lemmas A.1 and A.3,

$$\begin{aligned}
\overline{C}_k = Pre^k_\Sigma(\overline{C}_0, S_{xu}, D) \supseteq & \alpha Pre^k_\Sigma(C_{max}, S_{xu}, D) + \\
& (1-\alpha) Pre^k_\Sigma(C_0, S_{xu}, D) && \text{(A.10)} \\
= & \alpha C_{max} + (1-\alpha) Pre^k_\Sigma(C_0, S_{xu}, D) \\
= & \alpha C_{max} + (1-\alpha) C_k && \text{(A.11)}
\end{aligned}$$

Define $\overline{C}_\infty = \cup_{k=0}^\infty \overline{C}_k$. Then,

$$\begin{aligned}
\overline{C}_\infty &\supseteq \cup_{k=0}^\infty (\alpha C_{max} + (1-\alpha)C_k) && \text{(A.12)} \\
&= \alpha C_{max} + (1-\alpha) \cup_{k=0}^\infty C_k && \text{(A.13)} \\
&= \alpha C_{max} + (1-\alpha) C_\infty. && \text{(A.14)}
\end{aligned}$$

Since by construction of k_0, $\overline{C}_0 \subseteq C_{k_0}$,

$$\overline{C}_\infty \subseteq \cup_{k=1}^\infty Pre^k_\Sigma(C_{k_0}, S_{xu}, D) = \cup_{k=1}^\infty C_{k_0+k} = C_\infty. \quad \text{(A.15)}$$

Thus, by (A.14) and (A.15),

$$\begin{aligned}
&\alpha C_{max} + (1-\alpha)C_\infty \subseteq \overline{C}_\infty \subseteq C_\infty && \text{(A.16)} \\
&cl(\alpha C_{max} + (1-\alpha)C_\infty) \subseteq cl(\overline{C}_\infty) \subseteq cl(C_\infty). && \text{(A.17)}
\end{aligned}$$

Since C_{max} and $cl(C_\infty)$ are compact, $\alpha C_{max} + (1-\alpha)cl(C_\infty)$ is compact. Thus,

$$cl(\alpha C_{max} + (1-\alpha)C_\infty) \subseteq \alpha C_{max} + (1-\alpha)cl(C_\infty).$$

Let $x \in \alpha C_{max} + (1-\alpha)cl(C_\infty)$. Then, there exist points $x_1 \in C_{max}$ and $x_2 \in cl(C_\infty)$ such that $x = \alpha x_1 + (1-\alpha)x_2$. Since $x_2 \in cl(C_\infty)$, there exists $\{x_{2,k}\}_{k=1}^{\infty} \subseteq C_\infty$ such that $x_{2,k} \to x_2$. Since $\alpha x_1 + (1-\alpha)x_{2,k} \in \alpha C_{max} + (1-\alpha)C_\infty$ and $\alpha x_1 + (1-\alpha)x_{2,k} \to x$, $x \in cl(\alpha C_{max} + (1-\alpha)C_\infty)$ and thus $cl(\alpha C_{max} + (1-\alpha)C_\infty)$ is equal to $\alpha C_{max} + (1-\alpha)cl(C_\infty)$. Hence, (A.17) implies

$$\alpha C_{max} + (1-\alpha)cl(C_\infty) \subseteq cl(C_\infty) = \alpha cl(C_\infty) + (1-\alpha)cl(C_\infty). \tag{A.18}$$

Since $cl(C_\infty)$ is convex and compact, by order cancellation theorem [42, Theorem 4], (A.18) implies $\alpha C_{max} \subseteq \alpha cl(C_\infty)$ and thus $C_{max} \subseteq cl(C_\infty)$. Also, since C_{max} is the maximal RCIS, $C_\infty \subseteq C_{max}$. Thus, $C_{max} = cl(C_\infty)$. □

Proof of Theorem 3.4. Let β_0 be the infimum β such that $C_0 \subseteq \beta cl(C_\infty)$. We first want to show that $\beta_0 < 1$. By Lemma A.6, $cl(C_\infty)$ is a compact and convex set and there exists $\varepsilon > 0$ such that $C_0 + B_\varepsilon(0) \subseteq cl(C_\infty)$. Since $cl(C_\infty)$ is bounded, there exists $\alpha > 0$ such that $\alpha cl(C_\infty) \subseteq B_\varepsilon(0)$. Thus, $C_0 + \alpha cl(C_\infty) \subseteq cl(C_\infty) = \alpha cl(C_\infty) + (1-\alpha)cl(C_\infty)$. Since $\alpha cl(C_\infty)$ is a nonempty compact set and $(1-\alpha)cl(C_\infty)$ is convex, by the order cancellation theorem (Theorem 4 in [42]), $C_0 \subseteq (1-\alpha)cl(C_\infty)$. Thus, $\beta_0 \leq 1-\alpha < 1$.

Pick β_1 in $(\beta_0, 1)$. For now, let us assume that there exists a $N > 0$ such that

$$C_N = Pre_\Sigma^N(C_0, S_{xu}, D) \supseteq \beta_1 cl(C_\infty). \tag{A.19}$$

Then, $Pre_\Sigma^N(\beta_0 cl(C_\infty), S_{xu}, D) \supseteq \beta_1 cl(C_\infty)$. Let $\lambda = \beta_0/\beta_1$. Then, $\beta_1 cl(C_\infty)$ is an N-step λ-contractive set.

Now it is left to show that there exists a $N > 0$ such that (A.19) holds. It is easy to show that (i) $\{C_k\}_{k=1}^{\infty}$ is an expanding family of convex compact sets. (ii) Since $C_\infty = \cup_{k=1}^{\infty} C_k$ contains 0 in the interior, by Lemma A.6 (c), there exists k_0 such that C_{k_0} contains 0 in the interior. (iii) C_∞ is bounded since S_{xu} is bounded. Finally, since $cl(C_\infty)$ contains 0 in the interior, $cl(C_\infty) = \beta_1 cl(C_\infty) + (1-\beta_1)cl(C_\infty) \supseteq \beta_1 cl(C_\infty) + B_\varepsilon(0)$ for some small ε. Thus, $\beta_1 cl(C_\infty)$ is a compact set in the interior $int(cl(C_\infty))$. In the proof of Lemma A.4, we have shown that $int(cl(C_\infty)) = int(C_\infty)$. Thus, (iv) $\beta_1 cl(C_\infty)$ is a compact set in the interior of C_∞. Then, by Lemma A.4, there exists N such that $C_N \supseteq \beta_1 cl(C_\infty)$. □

Proof of Theorem 3.5 . Define

$$\lambda_{0,1} = \frac{\lambda_0 - \lambda\gamma}{1-\lambda\gamma},\ \lambda_{0,2} = \frac{\lambda\gamma(1-\lambda_0)}{1-\lambda\gamma}. \tag{A.20}$$

It is easy to check that $\lambda_{0,1} \geq 0$, $\lambda_{0,2} \geq 0$, $\lambda_0 = \lambda_{0,1} + \lambda_{0,2}$ and $1 - \lambda_{0,1} = \lambda_{0,2}/(\lambda\gamma)$. Thus, by

Lemmas A.2 and A.3, we have

$$\begin{aligned}
&Pre^N_\Sigma(\lambda_0 C, S_{xu}, D)\\
\supseteq &Pre^N_\Sigma(\lambda_{0,1} C, \lambda_{0,1} S_{xu}, \lambda_{0,1} D) + Pre^N_\Sigma(\lambda_{0,2} C, \frac{\lambda_{0,2}}{\lambda\gamma} S_{xu}, \frac{\lambda_{0,2}}{\lambda\gamma} D)\\
= &\lambda_{0,1} Pre^N_\Sigma(C, S_{xu}, D) + \frac{\lambda_{0,2}}{\lambda\gamma} Pre^N_\Sigma(\lambda\gamma C, S_{xu}, D)\\
\supseteq &\lambda_{0,1} C + \frac{\lambda_{0,2}}{\lambda\gamma}\gamma C = g(\lambda_0) C.
\end{aligned}$$

The inclusion in last row above holds since γC is N-step λ-contractive. □

APPENDIX B

Complementary Materials for Chapter 4

B.1 Claims of Theorem 4.12

Proof of Claim 1: Since $\overline{S}_{xu}$ contains the origin, we have $\overline{W}_\infty \times \{0\} \subseteq S_{xu}$. Since $0 \in W$, it is easy to verify from (4.2) and (4.3) that $\overline{W}_k \subseteq \overline{W}_\infty$ for all $k \geq 1$. Thus, $\overline{W}_k \times \{0\} \subseteq S_{xu}$ for all $k \geq 1$. According to (4.4), if $x = 0$ and $u_t = 0$ for all $t \geq 0$, the reachable set $\left(\mathscr{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) = \overline{W}_t \times \{0\} \subseteq S_{xu}$. Thus, $0 \in \mathscr{C}_{xv,0}$.

Proof of Claim 2: Recall from (4.30) that $\mathscr{C}_{outer,\nu}$ is:

$$\begin{aligned}\mathscr{C}_{outer,\nu} = \Big\{ & x \in \mathbb{R}^n \mid \exists \{u_i\}_{i=0}^{\nu-1} \in \mathbb{R}^{m\nu}, \\ & \left(\mathscr{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu}, t = 0, \ldots, \nu-1, \\ & \mathscr{R}_\Sigma\left(x, \{u_i\}_{i=0}^{\nu-1}\right) \subseteq \overline{\mathscr{C}}_{\max} + \overline{W}_\infty \Big\}.\end{aligned}$$

Due to Lemma 4.9, $\mathscr{R}_\Sigma\left(x, \{u_i\}_{i=0}^{\nu-1}\right) \subseteq \overline{\mathscr{C}}_{\max} + \overline{W}_\infty$ if and only if $\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i \in \overline{\mathscr{C}}_{max}$. Based on this observation, it is easy to verify that $\mathscr{C}_{outer,\nu} = \pi_n(\mathscr{C}_{xv,max})$ where $\mathscr{C}_{xv,max} = \mathscr{C}_{xv,0} \cap (\mathbb{R}^n \times \mathscr{U}(\overline{\mathscr{C}}_{max}))$.

Proof of Claim 3: We show that $\widehat{\mathscr{C}}_{xv,(\tau,\lambda)} = \pi_{n+\nu m}(\mathscr{C}_{xv,(\tau,\lambda)})$. Using the matrices H and P as in (4.16), the definition of $\mathscr{C}_{xv,(\tau,\lambda)}$, and Lemma 4.9, we write $\mathscr{C}_{xv,(\tau,\lambda)}$ as:

$$\begin{aligned}\mathscr{C}_{xv,(\tau,\lambda)} = \Big\{ & (x_0, u_{0:\tau+\lambda-1}) \mid \\ & \left(\mathscr{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu}, t = 0, \cdots, \nu-1, \\ & (\mathscr{R}_{\overline{\Sigma}}(\sum_{i=1}^{v} A^{i-1} B u_{v-i}, \{u_{v+i}\}_{i=0}^{k-1}), u_{v+k}) \in \overline{S}_{xu}, \\ & k = 0, \cdots, \tau+\lambda-1 \Big\}.\end{aligned} \tag{B.1}$$

By (B.1) , the projection $\pi_{n+\nu m}(\mathscr{C}_{xv,(\tau,\lambda)})$ is:

$$\begin{aligned}\pi_{n+\nu m}(\mathscr{C}_{xv,(\tau,\lambda)}) = \Big\{ & (x_0, u_{0:\nu-1}) \mid \exists u_{\nu:\tau+\lambda-1}, \\ & \left(\mathscr{R}_{\Sigma}\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu}, t = 0, \cdots, \nu-1, \\ & (\mathscr{R}_{\overline{\Sigma}}(\sum_{i=1}^{\nu} A^{i-1} B u_{\nu-i}, \{u_{\nu+i}\}_{i=0}^{k-1}), u_{\nu+k}) \in \overline{S}_{xu}, \\ & k = 0, \cdots, \tau+\lambda-1\Big\}.\end{aligned} \tag{B.2}$$

Again, using the matrices H and P as in (4.16), by the definition of $\overline{\mathscr{C}}_{x,(\tau-\nu,\lambda)}$, we have:

$$\begin{aligned}\overline{\mathscr{C}}_{x,(\tau-\nu,\lambda)} = \Big\{ & x_0 \in \mathbb{R}^n \mid \exists u_{0:\tau-\nu+\lambda}, \\ & (\mathscr{R}_{\overline{\Sigma}}(x_0, \{u_i\}_{i=0}^{t-1}), u_t) \in \overline{S}_{xu}, \\ & t = 0, \cdots, \tau+\lambda-1\Big\}.\end{aligned} \tag{B.3}$$

Comparing the right hand sides of (B.2) and (B.3), we have:

$$\begin{aligned}\pi_{n+\nu m}(\mathscr{C}_{xv,(\tau,\lambda)}) = \Big\{ & (x_0, u_{0:\nu-1}) \mid \\ & \left(\mathscr{R}_{\Sigma}\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu}, t = 0, \cdots, \nu-1, \\ & \sum_{i=1}^{\nu} A^{i-1} B u_{\nu-i} \in \overline{\mathscr{C}}_{x,(\tau-\nu,\lambda)}\Big\}.\end{aligned} \tag{B.4}$$

Note that $\mathscr{C}_{xv,0}$ and $\mathscr{U}(\overline{\mathscr{C}}_{x,(\tau-\nu,\lambda)})$ respectively impose the first and second constraints on $(x_0, u_{0:\nu-1})$ on the right hand side of (B.4). Thus, $\pi_{n+\nu m}(\mathscr{C}_{xv,(\tau,\lambda)})$ is equal to the intersection of $\mathscr{C}_{xv,0}$ and $\mathscr{U}(\overline{\mathscr{C}})_{x,(\tau-\nu,\lambda)}$. That is:

$$\widehat{\mathscr{C}}_{xv,(\tau,\lambda)} = \pi_{n+\nu m}(\mathscr{C}_{xv,(\tau,\lambda)}). \tag{B.5}$$

Since (B.5) implies (4.41), the third claim is proven.

Proof of Claim 4: We define the k-step null-controllable set $\mathscr{C}_k$ as the set of states of $\overline{\Sigma}$ that reach the origin at kth step under the state-input constraints S_{xu}:

$$\begin{aligned}\mathscr{C}_k = \Big\{ & x \in \mathbb{R}^n \mid \exists u_{0:k-1} \in \mathbb{R}^{km}, \\ & (\mathscr{R}_{\overline{\Sigma}}(x, \{u_i\}_{i=0}^{t-1}), u_t) \in \overline{S}_{xu}, t = 0, \cdots, k-1, \\ & \mathscr{R}_{\overline{\Sigma}}(x, \{u_i\}_{i=0}^{k-1}) = 0\Big\}.\end{aligned} \tag{B.6}$$

Obviously, $\mathscr{C}_0 = \{0\}$. Since $A^\nu = 0$ and the fixed point $(0,0) \in \mathbb{R}^n \times \mathbb{R}^m$ is in the interior of $\overline{S}_{xu}$, there exists an $\varepsilon > 0$ such that the ε-ball $B_\varepsilon(0)$ at the origin satisfies that for $u_{0:\nu-1} = 0 \in \mathbb{R}^{\nu m}$ and for all $t \in [0, k-1]$:

$$\begin{aligned} &(\mathscr{R}_{\overline{\Sigma}}(B_\varepsilon(0), \{u_i\}_{i=0}^{t-1}), u_t) = (A^t B_\varepsilon(0), 0) \subseteq \overline{S}_{xu}, \\ &\mathscr{R}_{\overline{\Sigma}}(x, \{u_i\}_{i=0}^{\nu-1} = A^\nu B_\varepsilon(0) = 0. \end{aligned} \tag{B.7}$$

By (B.7) and the definition of $\mathscr{C}_k$, $B_\varepsilon(0)$ is contained by $\mathscr{C}_\nu$, and thus $\mathscr{C}_0 = \{0\}$ is contained in the interior of $\mathscr{C}_\nu$. Then, by Theorem 3.1, since $\mathscr{C}_0$ is contained in the interior of $\mathscr{C}_\nu$, there exists $\tau_2 \geq 0$, $c_2 \in [0,1]$ and $a \in [0,1)$ such that for all $k \geq \tau_2$, the Hausdorff distance $d(\mathscr{C}_k, \overline{\mathscr{C}}_{max})$ satisfies that:

$$d(\mathscr{C}_k, \overline{\mathscr{C}}_{max}) \leq c_2 a^k. \tag{B.8}$$

Furthermore, let $k = \tau$. For any $x \in \mathscr{C}_\tau$ and the corresponding $u_{0:\tau-1}$ satisfying the constraints on the right hand side of (B.6), it is easy to check that $(x, u_{0:\tau-1}, 0) \in \mathbb{R}^n \times \mathbb{R}^{(\tau+\lambda)m}$ is contained in $\overline{\mathscr{C}}_{xv,(\tau,\lambda)}$. Thus, we have for all $\tau \geq 0$:

$$\mathscr{C}_\tau \subseteq \overline{\mathscr{C}}_{x,(\tau,\lambda)} \subseteq \overline{\mathscr{C}}_{max}. \tag{B.9}$$

Thus, by (B.8) and (B.9), for any $\tau \geq \tau_2$, the Hausdorff distance $d(\overline{\mathscr{C}}_{x,(\tau,\lambda)}, \overline{\mathscr{C}}_{max})$ satisfies:

$$d(\overline{\mathscr{C}}_{x,(\tau,\lambda)}, \overline{\mathscr{C}}_{max}) \leq c_2 a^\tau. \tag{B.10}$$

From the properties of Hausdorff distance, (B.10) implies that:

$$\overline{\mathscr{C}}_{max} \subseteq \overline{\mathscr{C}}_{x,(\tau,\lambda)} + B_{c_2 a^\tau}(0), \tag{B.11}$$

where $B_{c_2 a^\tau}(0)$ is the ball at origin with radius $c_2 a^\tau$. Recall that $\mathscr{C}_\nu$ contains a ε-ball $B_\varepsilon(0)$ for some $\varepsilon > 0$. Since $\mathscr{C}_\nu \subseteq \overline{\mathscr{C}}_{max}$, we have $(c_2 a^\tau/\varepsilon)\overline{\mathscr{C}}_{max} \supseteq B_{c_2 a^\tau}(0)$. Thus, by (B.11), we have for any $\tau \geq \tau_2$:

$$\overline{\mathscr{C}}_{max} \subseteq \overline{\mathscr{C}}_{x,(\tau,\lambda)} + \frac{c_2 a^\tau}{\varepsilon} \overline{\mathscr{C}}_{max}. \tag{B.12}$$

Select a big enough τ_1 such that $\tau_1 \geq \tau_2$ and $c_2 a^{\tau_1} \leq \varepsilon$. Then, by Lemma 4.9 and (B.12), we have for any $\tau \geq \tau_1$:

$$\overline{\mathscr{C}}_{x,(\tau,\lambda)} \supseteq (1 - c_0 a^\tau)\overline{\mathscr{C}}_{max},$$

where $c_0 = \frac{c_2}{\varepsilon}$. Thus, the fourth claim is proven.

APPENDIX C

Complementary Materials for Chapter 5

Proof of Proposition 5.2 . Let $x \in C$ and $d \in D$. By the Definition 5.2, $\mathscr{R}(\Sigma, f(x,d)) \subseteq \mathscr{R}(\Sigma_a, x) \subseteq S$ and thus $f(x,d) \in C$. Hence, C is an RPIS by definition.

Suppose x belongs to an arbitrary RPIS within S. By definition, $\mathscr{R}(\Sigma_a, x) \subseteq S$. Thus, $x \in C$ and C is the maximal RPIS. □

Proof of Proposition 5.1. The "only if" direction is obvious. It is left to show the "if" direction. Suppose the safe set is S. Let C be an RCIS for the system Σ' in S and x be a point in C. We want to show that for all $d \in D$, there exists u such that $Ax + Bu + d \in S$. Since $D = \text{conv}(D_v)$, there exists a finite $K > 0$ such that $d = \sum_{i=1}^{K} \alpha_i d_i$ for some d_1, ..., $d_K \in D_v$ and some α_1, ..., $\alpha_K \geq 0$ satisfying $\sum_{i=1}^{K} \alpha_i = 1$. Since C is controlled invariant for Σ', for each $d_i \in D_v$, there exists u_i such that $(x, u_i) \in S$ and $Ax + Bu_i + Ed_i \in C$. Define $u = \sum_{i=1}^{K} \alpha_i u_i$. It is easy to show that $(x,u) \in S$ and $Ax + Bu + Ed \in C$, by the convexity of S and C. Thus, C is an RCIS within S for Σ. □

Proof of Theorem 5.3. Denote $\text{conv}(\pi_{[1,n]}(C_{cl}))$ by C_p. Let $x \in C_p$ and $d \in D$. We want to show that there exists u such that $(x,u) \in S$ and $Ax + Bu + d \in C_p$.

By definition of convex hull, there exist a positive integer $k > 0$, k vectors $x_i \in \pi_{[1,n]}(C_{cl})$ and k scalars $\alpha_i \in [0,1]$ for i from 1 to k such that $\sum_{i=1}^{k} \alpha_i = 1$ and $\sum_{i=1}^{k} \alpha_i x_i = x$. For each i, there exists θ_i and s_i such that $(x_i, \theta_i, s_i) \in C_{cl}$. We define $u_i = o(s_i, d)$. Note that by the definition of S_{cl}, $(x_i, u_i) \in S$. Also, since C_{cl} is an RPIS, $(Ax_i + Bu_i + d, \theta_i, \mathscr{T}(s_i, d)) \in C_{cl}$ and thus $Ax_i + Bu_i + d \in C_{proj}$. We define $u = \sum_{i=1}^{k} \alpha_i u_i$. Since S is convex and $(x_i, u_i) \in S$, $(x,u) = \sum_{i=1}^{k} \alpha_i (x_i, u_i) \in S$. Since C_p is convex and $Ax_i + Bu_i + d \in C_p$, $Ax + Bu + d = \sum_{i=1}^{k} \alpha_i (Ax_i + Bu_i + d) \in C_p$. Thus, C_p is an RCIS for the system Σ in S. □

Proof of Theorem 5.4. We want to show $\mathscr{R}(\Sigma_{cl}, (x, \theta, s))$ is finite. Let $((x(t), \theta(t), s(t)))_{t=0}^{\infty}$ be the trajectory of Σ_{cl} with initial state $(x(0), \theta(0), s(0)) = (x, \theta, s)$. Let $(d(t))_{t=0}^{\infty}$ be the disturbance

sequence. Given A is nilpotent, that is $A^h = 0$ for some $h \geq 0$, we have that

$$x(t) = \begin{cases} A^t x + \sum_{i=0}^{t-1} A^{t-1-i}[Bo(s(i), d(i); \theta) + d(i)] & t < h, \\ \sum_{i=t-h}^{t-1} A^{t-1-i}[Bo(s(i), d(i); \theta) + d(i)] & t \geq h. \end{cases} \tag{C.1}$$

Since $s(t)$ and $d(t)$ belong to finite sets Q and D, $o(s(t), d(t); \theta)$ belongs to the finite set $U(\theta) = \{o(s, d; \theta)\}_{s \in Q, d \in D}$. Thus, according to (C.1), $x(t)$, as a function of $o(s(t), d(t); \theta)$ and $d(t)$ for $t \geq h$, must belongs to a finite set, denoted by $X(\theta)$. Thus, the reachable set $\mathscr{R}(\Sigma_{cl}, (x, \theta, s)) \subseteq X(\theta) \times \{\theta\} \times Q$ is a finite set. □

Proof of Theorem 5.5. We want to derive a sufficient condition under which $\pi_{[1,n]}(C_{sub}(s_1)) \supseteq \pi_{[1,n]}(C_{sub}(s_2))$. Note that if for all $(x, \theta_2) \in C_{sub}(s_2)$, there exists θ_1 such that $(x, \theta_1) \in C_{sub}(s_1)$, then we have $\pi_{[1,n]}(C_{sub}(s_1)) \supseteq \pi_{[1,n]}(C_{sub}(s_2))$.

Similar to how we define $o^*(s, d)$, we define the parametrized nested output function as

$$o^*(s, (d(t))_{t=0}^k; \theta) = \begin{cases} o(s, d(0); \theta) & k = 0, \\ o(\mathscr{T}^*(s, (d(t))_{t=0}^{k-1}), d(k); \theta) & k > 0. \end{cases} \tag{C.2}$$

Given s and θ, the parametrized nested output function $o^*(s, \cdot; \theta)$ becomes a function of $(d(t))_{t=0}^k$ in $\cup_{i=1}^{\infty} D^i$. If for any θ_2, we can always find a θ_1 such that the functions $o^*(s_2, \cdot; \theta_2) = o^*(s_1, \cdot; \theta_1)$, then for all $(x, \theta_2) \in C_{sub}(s_2)$, $(x, \theta_1) \in C_{sub}(s_1)$. Intuitively, recall that $(x, \theta_2) \in C_{sub}(s_2)$ if $(x(k), o^*(s_2, (d(t))_{t=0}^k)) \in S$ for all $k \geq 0$ and $(d(t))_{t=0}^k \in D^k$. If we know that $(x(k), o^*(s_2, (d(t))_{t=0}^k; \theta_2)) \in S$ for all $k \geq 0$, then we know $(x(k), o^*(s_2, (d(t))_{t=0}^k; \theta_2)) \in S$ for all $k \geq 0$ since $o^*(s_2, \cdot; \theta_2) = o^*(s_1, \cdot; \theta_1)$. Thus, $(x, \theta_1) \in C_{sub}(s_1)$.

Now our goal is to derive a sufficient condition under which there exists a θ_1 such that $o^*(s_1, \cdot; \theta_1) = o^*(s_2, \cdot; \theta_2)$ for all θ_2.

Lemma C.1. Given s_1, $s_2 \in Q$ and θ_1, θ_2, the functions $o^*(s_1, \cdot; \theta_1) = o^*(s_2, \cdot; \theta_2)$ if and only if $o^*(s_1, (d(t))_{t=0}^k; \theta_1) = o^*(s_2, (d(t))_{t=0}^k; \theta_2)$ for all $k \leq |Q|^2$ and all $(d(t))_{t=0}^k \in D^k$.

According to Lemma C.1, given any θ_2, we can directly solve for a θ_1 satisfying $o^*(s_1, (d(t))_{t=0}^k; \theta_1) = o^*(s_2, (d(t))_{t=0}^k; \theta_2)$ for all $k \leq |Q|^2$ and all $(d(t))_{t=0}^k \in D^k$, which is a system of linear equations on θ_1. It can be checked that given any θ_2, the solvability of the system of equations on θ_1 is guaranteed if for all $(d_1(t))_{t=0}^{k_1}$ and $(d_2(t))_{t=0}^{k_2}$ with k_1, $k_2 \leq |Q|^2$, $o^*(s_1, (d_1(t))_{t=0}^{k_1}) = o^*(s_1, (d_2(t))_{t=0}^{k_2})$ implies $o^*(s_2, (d_1(t))_{t=0}^{k_1}) = o^*(s_2, (d_2(t))_{t=0}^{k_2})$, that is $s_1 \succeq s_2$ by definition. □

Proof of Lemma C.1 . Given the mealy machine $(Q, D, \mathscr{T}, \Theta, o)$, we can construct a product mealy

machine $(Q\times Q, D, \mathscr{T}_{pd}, \Theta\times\Theta, o_{pd})$ where for all s_i, $s_j \in Q$ and $d \in D$

$$\mathscr{T}_{pd}((s_i,s_j),d) = (\mathscr{T}(s_i,d), \mathscr{T}(s_j,d)), \tag{C.3}$$

$$o_{pd}((s_i,s_j),d) = (o(s_i,d), o(s_j,d)). \tag{C.4}$$

Given θ_1 and θ_2 as two value assignments of Θ, we define the parametrized output function $o_{pd}((s_i,s_j),d;\theta_1,\theta_2) = (o(s_i,d;\theta_1), o(s_j,d;\theta_2))$.

Given s_1, $s_2 \in Q$ and θ_1 and θ_2, by construction, $o^*_{pd}((s_1,s_2),\cdot;\theta)$ is equal to $(o^*(s_1,\cdot;\theta_1), o^*(s_2,\cdot;\theta_2))$. Thus, $o^*(s_1,\cdot;\theta_1) \neq o^*(s_2,\cdot;\theta_2)$ if and only if there exists a $(d(t))_{t=0}^k$ such that $(s_1',s_2') = \mathscr{T}^*_{pd}((s_1,s_2),(d(t))_{t=0}^{k-1})$ and $o_{pd}((s_1',s_2'),d(k);\theta_1,\theta_2) = (u_1,u_2)$ for some u_1, $u_2 \in \Theta$, $\bar{u}_1 \neq \bar{u}_2$. Since there are only $|Q|^2$ states in the product mealy machine, if (s_1',s_2') can be visited from (s_1,s_2) under action sequence $(d(t))_{t=0}^{k-1}$, the smallest k we need is less than or equal to $|Q|^2$. Thus, if $o^*(s_1,\cdot;\theta_1) \neq o^*(s_2,\cdot;\theta_2)$, there must exists a $(d(t))_{t=0}^k$ with $k \leq |Q|^2$ such that $o^*(s_1,(d(t))_{t=0}^k;\theta_1) \neq o^*(s_2,(d(t))_{t=0}^k;\theta_2)$ □

Proof of Theorem 5.6. Denote $C_p = \pi_{[1,n]}(C')$. Suppose C' is an RCIS of Σ' in $S\times\mathbb{R}^m$. Let $x \in C_p$. We want to show that there exist u such that $(x,u) \in S$ and for all $d \in D$, $Ax+Bu+d \in C_p$.

By definition of C_p, there exists $u \in \mathbb{R}^m$ such that $(x,u) \in C' \subseteq S$. Furthermore, since C' is controlled invariant, there exists $v \in \mathbb{R}^m$ such that $(Ax+Bu+d, v) \in C'$ for all $d \in D$. Thus, $Ax+Bu+d \in C_p$ for all $d \in D$. Thus, we showed that C_p is an RCIS for the system Σ in S.

Next, suppose that C' is the maximal RCIS for Σ' in $S\times\mathbb{R}^m$. Also, suppose that C_{max} is the maximal RCIS for Σ in S. We want to show that $\pi_{[1,n]}(C') = C_{max}$. Note that $\pi_{[1,n]}(C') \subseteq C_{max}$ as $\pi_{[1,n]}(C')$ is controlled invariant for Σ in S. We need to show that $\pi_{[1,n]}(C') \supseteq C_{max}$, which is done in 3 steps.

First, define the set $C'_{max} = \{(x,u) \mid (x,u) \in S, Ax+Bu+d \in C_{max}, \forall d \in D\}$. We want to show that C'_{max} is controlled invariant for Σ' in $S\times\mathbb{R}^m$. Let $(x,u) \in C'_{max}$ and $d \in D$. By construction, $(x,u) \in S$ and $x^+ = Ax+Bu+d \in C_{max}$. Since C_{max} is controlled invariant for Σ, there exists $v \in \mathbb{R}^m$ such that $(x^+,v) \in S$ and $Ax^+ + Bu + d^+ \in C_{max}$ for all $d^+ \in D$. Thus, by definition of C'_{max}, $(x^+,v) = (Ax+Bu+d,v) \in C'_{max}$. Thus, C'_{max} is an RCIS for Σ' in $S\times\mathbb{R}^m$.

Second, as C' is the maximal RCIS for Σ' in $S\times\mathbb{R}^m$, $C' \geq C'_{max}$. Thus, $\pi_{[1,n]}(C') \supseteq \pi_{[1,n]}(C'_{max})$.

Finally, note that for all $x \in C_{max}$, there exists u such that $(x,u) \in S$ and $Ax+Bu+d \in C_{max}$ for all $d \in D$, namely that $(x,u) \in C'_{max}$. Hence, $C_{max} \subseteq \pi_{[1,n]}(C'_{max}) \subseteq \pi_{[1,n]}(C')$. That is, $\pi_{[1,n]}(C') = C_{max}$ is the maximal RCIS for Σ in S. □

APPENDIX D

Complementary Materials for Chapter 7

Proof of Theorem 7.2. First, we want to show the set $C_{max,p,co}$ in (7.11) is a CIS of the system $\mathscr{D}(\Sigma_p)$ within the safe set $S_{xu,p} \times D$. Let $(x(0), d_{1:p}(0))$ be any point in $C_{max,p,co}$. By (7.11) , there exists $u(0)$, ..., $u(p-1)$ such that $(x(t), u(t))$ is in S_{xu} for t from 0 to $p-1$ and $x(p)$ is in $C_{max,co}$. Since $C_{max,co}$ is controlled invariant for $\mathscr{D}(\Sigma)$, there exists $u(p)$ and $v(0) \in D$ such that $(x(p), u(p))$ is in S_{xu} and $x(p+1) = f(x(p), u(p), v(0))$ is in $C_{max,co}$. Thus, for control inputs $u(0)$ and $v(0)$, it is easy to check that

$$(x(0), d_{1:p}(0), u(0), v(0)) \in S_{xu,p} \times D$$

and the next state

$$(x(1), d_{2:p}(0), v(0)) \in C_{max,p,co},$$

where $x(1) = f(x(0), u(0), d_1(0))$. Thus, $C_{max,p,co}$ is a CIS of $\mathscr{D}(\Sigma_p)$ within $S_{xu,p,co}$.

Next, denote the maximal CIS of $\mathscr{D}(\Sigma_p)$ in $S_{xu,p} \times D$ by $\overline{C}_{max,p,co}$. Suppose that $C_{max,p,co}$ is strictly contained in $\overline{C}_{max,p,co}$. Then, there exists at least one point $(x(0), d_1(0), \cdots, d_p(0))$ in $\overline{C}_{max,p,co}$ but not in $C_{max,p,co}$. That implies for all $u(0)$, ..., $u(p-1)$ satisfying $(x(t), u(t)) \in S_{xu}$ for t from 0 to $p-1$, we have $x(p) \notin C_{max,co}$. As $C_{max,co}$ is the maximal CIS of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$, for all input sequence $(u(t))_{t=p}^{\infty}$ and $(v(t))_{t=p}^{\infty}$ with $v(t) \in D$, there always exists a finite time instant $T \geq p$ such that $(x(T), u(T)) \notin S_{xu}$. That implies that any state-input trajectory of $\mathscr{D}(\Sigma_p)$ from $(x(0), d_{1:p}(0))$ always leaves the safe set $S_{xu,p} \times D$ in finite time for any input sequence, contradicting to the assumption that $(x(0), d_{1:p}(0))$ is in the maximal CIS $\overline{C}_{max,p,co}$. Thus, $C_{max,p,co}$ must be the maximal RCIS $\overline{C}_{max,p,co}$ and thereby (7.11) is proven.

Finally, by the construction of $S_{xu,p}$, it is easy to check that $C_{max,p,co} \subseteq \mathbb{R}^n \times D^p$. Thus, to prove (7.12) is equivalent to prove that

$$\pi_{[1,n]}(C_{max,p,co}) \subseteq C_{max,co}. \tag{D.1}$$

Let $x(0)$ be any point in $\pi_{[1,n]}(C_{max,p,co})$. By (7.11), it is easy to check that

$$x(0) \in Pre^{p}_{\mathscr{D}(\Sigma)}(C_{max,co}, S_{xu} \times D). \tag{D.2}$$

Since $C_{max,co}$ is the maximal CIS of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$, we have

$$Pre^{p}_{\mathscr{D}(\Sigma)}(C_{max,co}, S_{xu} \times D) = C_{max,co}. \tag{D.3}$$

Thus, $x(0) \in C_{max,co}$. That is, (D.1) is proven. □

Proof of Lemma 7.3. We first show that $\pi_{[1,n]}(C_{max,p})$ is convex and compact. Note that Assumption 7.2 implies that the safe set $S_{xu,p}$ is convex and compact. Thus, the maximal RCIS $C_{max,p}$ is compact. Also, for linear systems, the convex hull of any RCIS is also a RCIS. Thus, the maximal RCIS of Σ_p in $S_{xu,p}$ must be convex (otherwise we can take the convex hull of the maximal RCIS and obtain a larger RCIS). Since $C_{max,p}$ is convex and compact, the projection $\pi_{[1,n]}(C_{max,p})$ is convex and compact.

Next, we show that $\pi_{[1,n]}(C_{max,p})$ is a CIS of $\mathscr{D}(\Sigma)$ within $S_{xu} \times D$ by definition. Let x be an arbitrary point in $\pi_{[1,n]}(C_{max,p})$. There exists $d_{1:p} \in D^p$ such that $(x, d_{1:p}) \in C_{max,p}$. As $C_{max,p}$ is an RCIS of Σ_p, there exists u_0 such that $(x, u_0) \in S_{xu}$ such that $(Ax+Bu_0+Ed_1, d_{2:p}, d) \in C_{max,p}$ for any $d \in D$. Thus, there exists u and u_d such that $(x, u, u_d) \in S_{xu} \times D$ and $Ax+Bu+Eu_d \in \pi_{[1,n]}(C_{max,p})$ (one feasible choice is $u = u_0$, $u_d = d_1$). Thus, by definition, $\pi_{[1,n]}(C_{max,p})$ is a CIS of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$.

Finally, [9, Proof of Theorem 3.3] shows[1] that for any nonempty convex compact CIS C of a linear system Σ in safe set S_{xu}, there always exists a forced equilibrium $(x_e, u_e) \in S_{xu}$ of the system such that $x_e \in C$. Thus, (i) implies (ii). □

Proof of Lemma 7.4. We first show the 1-step backward reachable set of the projection $\pi_{[1,n]}(C_{max,p})$ is contained in the projection $\pi_{[1,n]}(C_{max,p+1})$, that is

$$\begin{aligned} &Pre_{\mathscr{D}(\Sigma)}(PROJ_{[1,n]}(C_{max,p}), S_{xu} \times D) \\ &\subseteq \pi_{[1,n]}(C_{max,p+1}). \end{aligned} \tag{D.4}$$

[1] [9, Proof of Theorem 3.3] only proves the case without input constraints, which can be easily extended to the case with state-input constraints according to [9, Remark 1]. Also, though [9] mainly considers controllable systems, the part used to prove (ii) holds for any linear systems.

By Theorem 7.1, $C_{max,p} \times D \subseteq C_{max,p+1}$, which implies

$$\begin{aligned} &Pre_{\Sigma_{p+1}}(C_{max,p} \times D, S_{xu,p+1}) \\ &\subseteq Pre_{\Sigma_{p+1}}(C_{max,p+1}, S_{xu,p+1}) = C_{max,p+1}. \end{aligned} \tag{D.5}$$

Thus, to prove (D.4), it is sufficient to show

$$\begin{aligned} &Pre_{\mathscr{D}(\Sigma)}(\pi_{[1,n]}(C_{max,p}), S_{xu} \times D) \\ &\subseteq \pi_{[1,n]}(Pre_{\Sigma_{p+1}}(C_{max,p} \times D, S_{xu,p+1})). \end{aligned} \tag{D.6}$$

We select an arbitrary point x_0 such that

$$x_0 \in Pre_{\mathscr{D}(\Sigma)}(\pi_{[1,n]}(C_{max,p}), S_{xu} \times D).$$

By the definition of *Pre*, there exists u_0, d_0 such that the point (x_0, u_0, d_0) is in $S_{xu} \times D$ and the point $x_1 = f(x_0, u_0, d_0)$ is in $PROJ_{[1,n]}(C_{max,p})$. Also, there exists $d_{1:p} \in D^p$ such that $(x_1, d_{1;p}) \in C_{max,p}$. That is, $(f(x_0, u_0, d_0), d_{1:p}, D) \subseteq C_{max,p} \times D$. Thus, $(x_0, d_0, d_{1:p}) \in Pre_{\Sigma_{p+1}}(C_{max,p} \times D)$, which implies

$$x_0 \in \pi_{[1,n]}(Pre_{\Sigma_{p+1}}(C_{max,p} \times D)).$$

Thus, (D.4) is proven.

Next, by taking *Pre* on both sides of (D.4), we have

$$\begin{aligned} &Pre^2_{\mathscr{D}(\Sigma)}(PROJ_{[1,n]}(C_{max,p}), S_{xu} \times D) \\ &\subseteq Pre_{\mathscr{D}(\Sigma)}(\pi_{[1,n]}(C_{max,p+1}), S_{xu} \times D) \\ &\subseteq \pi_{[1,n]}(C_{max,p+2}), \end{aligned} \tag{D.7}$$

where the second inclusion is due to (D.4). Following the pattern in (D.4) and D.7, (7.18) can be easily proven by induction. □

To prove Theorem 7.5, we need the following lemma.

Lemma D.1. Under the same conditions of Theorem 7.5, for any $\xi \in (0, 1]$, the N-step backward reachable set of $\xi C_{max,co}$ satisfies that for $\xi \leq \lambda\gamma$,

$$Pre^N_{\mathscr{D}(\Sigma)}(\xi C_{max,co}, S_{xu} \times D) \supseteq \frac{\xi}{\lambda} C_{max,co}; \tag{D.8}$$

for $\xi > \lambda\gamma$,

$$\begin{aligned} &Pre^N_{\mathscr{D}(\Sigma)}(\xi C_{max,co}, S_{xu} \times D) \\ &\supseteq \left(1-(1-\xi)\left(\frac{1-\gamma}{1-\gamma\lambda}\right)\right) C_{max,co}, \end{aligned} \tag{D.9}$$

Proof. First, it can be easily checked that for any $\gamma' \leq \gamma$, $\gamma' C_{max,co}$ is an N-step λ-contractive CIS of $\mathscr{D}(\Sigma)$ in $\gamma'/\gamma(S_{xu} \times D)$, which is a subset of $S_{xu} \times D$.

Case 1: $\xi \leq \lambda\gamma$. Since $\xi/\lambda \leq \gamma$, $(\xi/\lambda)C_{max,co}$ is an N-step λ-contractive CIS of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$, which implies (D.8).

Case 2: $\xi > \lambda\gamma$. We first separate ξ into two parts, that is

$$\xi = \xi_1 + \lambda\xi_2 \tag{D.10}$$

where

$$\xi_1 = \frac{\xi - \lambda\gamma}{1-\lambda\gamma},\ \xi_2 = \gamma(\frac{1-\xi}{1-\lambda\gamma}). \tag{D.11}$$

Since $C_{max,co}$ is convex, we can separate $\xi C_{max,co}$ into two parts, that is

$$\xi C_{max,co} = \xi_1 C_{max,co} + \lambda\xi_2 C_{max,co}. \tag{D.12}$$

It can be shown that for any convex set C, and any a, $b \in [0,1]$,

$$\begin{aligned} &Pre^N_{\mathscr{D}(\Sigma)}(C, S_{xu} \times D) \supseteq Pre^N_{\mathscr{D}(\Sigma)}(aC, b(S_{xu} \times D)) + \\ &Pre^N_{\mathscr{D}(\Sigma)}((1-a)C, (1-b)(S_{xu} \times D)). \end{aligned} \tag{D.13}$$

By (D.10) and (D.13), we have

$$\begin{aligned} &Pre^N_{\mathscr{D}(\Sigma)}(\xi C_{max,co}, S_{xu} \times D) \supseteq \\ &Pre^N_{\mathscr{D}(\Sigma)}(\gamma_1 C_{max,co}, \gamma_1(S_{xu} \times D)) + \\ &Pre^N_{\mathscr{D}(\Sigma)}(\lambda\gamma_2 C_{max,co}, (1-\gamma_1)(S_{xu} \times D)). \end{aligned} \tag{D.14}$$

Since $\gamma_1 C_{max,co}$ is a CIS of $\mathscr{D}(\Sigma)$ in $\gamma_1(S_{xu} \times D)$, we have that

$$Pre^N_{\mathscr{D}(\Sigma)}(\gamma_1 C_{max,co}, \gamma_1(S_{xu} \times D)) \supseteq \gamma_1 C_{max,co}. \tag{D.15}$$

Note that $1-\gamma_1 = \gamma_2/\gamma$. Also, for a linear $\mathscr{D}(\Sigma)$, we have that for any set C and $a > 0$, we have

$$Pre^N_{\mathscr{D}(\Sigma)}(C, S_{xu} \times D) = \frac{1}{a} Pre^N_{\mathscr{D}(\Sigma)}(aC, a(S_{xu} \times D)). \tag{D.16}$$

Thus, by (D.16),

$$\begin{aligned}
&Pre^N_{\mathscr{D}(\Sigma)}(\lambda\gamma_2 C_{max,co}, (1-\gamma_1)(S_{xu} \times D)) \\
=&Pre^N_{\mathscr{D}(\Sigma)}(\lambda\gamma_2 C_{max,co}, \gamma_2/\gamma(S_{xu} \times D)) \\
=&(\gamma_2/\gamma) Pre^N_{\mathscr{D}(\Sigma)}(\lambda\gamma C_{max,co}, S_{xu} \times D) \\
\supseteq&(\gamma_2/\gamma)\gamma C_{max,co} = \gamma_2 C_{max,co}.
\end{aligned} \tag{D.17}$$

By (D.14), (D.15) and (D.17), we have

$$Pre^N_{\mathscr{D}(\Sigma)}(\xi C_{max,co}, S_{xu} \times D) \supseteq (\gamma_1 + \gamma_2) C_{max,co} \tag{D.18}$$

It is easy to check that

$$\gamma_1 + \gamma_2 = 1 - (1-\xi)\frac{1-\gamma}{1-\gamma\lambda},$$

which proves (D.9). □

Now we are ready to prove Theorem 7.5.

Proof of Theorem 7.5. We only prove the case of $\lambda_0 \leq \lambda\gamma$, since the case of $\lambda_0 > \lambda\gamma$ can be easily extended from this proof. Given $\lambda_0 \leq \lambda\gamma$, by (D.8), we have

$$Pre^N_{\mathscr{D}(\Sigma)}(\lambda_0 C_{max,co}, S_{xu} \times D) \supseteq \frac{\lambda_0}{\lambda} C_{max,co}. \tag{D.19}$$

If $\lambda_0/\lambda \leq \lambda\gamma$, we can compute the N-step backward reachable sets of both sides of (D.19) and apply (D.8) again, which leads to

$$\begin{aligned}
Pre^{2N}_{\mathscr{D}(\Sigma)}(\lambda_0 C_{max,co}, S_{xu} \times D) &\supseteq Pre^N_{\mathscr{D}(\Sigma)}(\frac{\lambda_0}{\lambda} C_{max,co}) \\
&\supseteq \frac{\lambda_0}{\lambda^2} C_{max,co}.
\end{aligned} \tag{D.20}$$

Following the pattern of (D.19) and (D.20), as long as $\lambda_0/\lambda^{k-1} \leq \lambda\gamma$, we can easily prove by induction that

$$Pre^{kN}_{\mathscr{D}(\Sigma)}(\lambda_0 C_{max,co}, S_{xu} \times D) \supseteq \frac{\lambda_0}{\lambda^k} C_{max,co}. \tag{D.21}$$

We denote the minimal k such that $\lambda_0/\lambda^k > \lambda\gamma$ by k_0. Then,

$$k_0 = \left\lceil \frac{\log\lambda_0 - \log\gamma}{\log\lambda} - 1 \right\rceil. \tag{D.22}$$

By (D.21) and (D.22), we have proven (7.20) for $k \leq k_0$. Note that when $k = k_0$, we have $\lambda_0/\lambda^{k_0} > \lambda\gamma$. Thus,

$$\begin{aligned} Pre^{k_0N}_{\mathscr{D}(\Sigma)}(\lambda_0 C_{max,co}, S_{xu} \times D) &\supseteq \frac{\lambda_0}{\lambda^{k_0}} C_{max,co} \\ &\supseteq \lambda\gamma C_{max,co}. \end{aligned} \tag{D.23}$$

We define a function $g : \mathbb{R} \to \mathbb{R}$ by

$$g(\xi) = 1 - (1-\xi)\left(\frac{1-\gamma}{1-\gamma\lambda}\right). \tag{D.24}$$

Then, we define $g^k(\xi)$ recursively by

$$g^1(\xi) = g(\xi), \tag{D.25}$$

$$g^k(\xi) = g(g^{k-1}(\xi)). \tag{D.26}$$

By taking N-step backward reachable sets on both sides of (D.23) and applying (D.9), we have

$$\begin{aligned} &Pre^{(k_0+1)N}_{\mathscr{D}(\Sigma)}(\lambda_0 C_{max,co}, S_{xu} \times D) \\ &\supseteq Pre^{N}_{\mathscr{D}(\Sigma)}\left(\frac{\lambda_0}{\lambda^{k_0}} C_{max,co}, S_{xu} \times D\right) \\ &\supseteq g(\lambda_0/\lambda^{k_0}) C_{max,co}. \end{aligned} \tag{D.27}$$

It can be checked that if $\xi > \lambda\gamma$, then $g^k(\xi) > \lambda\gamma$ for all $k \geq 1$. Thus, $g^k(\lambda_0/\lambda^{k_0}) > \lambda\gamma$ for all $k \geq 1$. Following the pattern of (D.27), we can prove by induction that

$$\begin{aligned} &Pre^{(k_0+1)N}_{\mathscr{D}(\Sigma)}(\lambda_0 C_{max,co}, S_{xu} \times D) \\ &\supseteq g(\lambda_0/\lambda^{k_0}) C_{max,co} \end{aligned} \tag{D.28}$$

Finally, it can be checked that

$$g^k(\xi) = 1 - (1-\xi)\left(\frac{1-\gamma}{1-\gamma\lambda}\right)^{k-k_0}. \tag{D.29}$$

By (D.28) and (D.29), we have proven (7.21) for $k \geq k_0$.

The proof for the case of $\lambda_0 > \lambda\gamma$ follows exactly the same arguments from (D.23) to (D.29), with $k_0 = 0$. □

Proof of Lemma 7.6. By the proof of Proposition 23 of [32], since the system $\mathscr{D}(\Sigma)$ is stabilizable and the safe set $S_{xu} \times D$ contains the origin in the interior, there exists a λ_a-contractive ellipsoidal CIS $\mathscr{E}$ within the safe set $S_{xu} \times D$ for some $\lambda_a \in [0,1)$. Clearly, $\mathscr{E} \subseteq C_{max,co}$, and thereby $C_{max,co}$ contains the origin in the interior.

Let λ_{in} be the maximal λ such that $\lambda C_{max,co} \subseteq \mathscr{E}$. As $\mathscr{E}$ contains the origin in the interior, $\lambda_{in} > 0$. Since $\mathscr{E}$ is λ-contractive, we have

$$\lambda_{in} C_{max,co} \subseteq \mathscr{E} \subseteq Pre^k_{\mathscr{D}(\Sigma)}(\lambda_a^k \mathscr{E}, S_{xu} \times D). \tag{D.30}$$

Let λ be an arbitrary number in $(0,1)$. Since $\lambda_a \in [0,1)$ and $\mathscr{E} \subseteq C_{max,co}$, there exists $N > 0$ such that $\lambda_a^N \mathscr{E} \subseteq \lambda\lambda_{in} C_{max,co}$. Thus,

$$\begin{aligned} \lambda_{in} C_{max,co} &\subseteq Pre^N_{\mathscr{D}(\Sigma)}(\lambda_a^N \mathscr{E}, S_{xu} \times D) && \text{(D.31)} \\ &\subseteq Pre^N_{\mathscr{D}(\Sigma)}(\lambda\lambda_{in} C_{max,co}, S_{xu} \times D). && \text{(D.32)} \end{aligned}$$

Thus, $\lambda_{in} C_{max,co}$ is an N-step λ-contractive CIS of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$. □

Proof of Theorem 7.8. By the definition of c_0, $\mathscr{E}(c_0)$ is a λ_a-contractive CIS of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$. Therefore, $\mathscr{E}(c_0)$ is a subset of $C_{max,co}$. Also, $\gamma C_{max,co} \subseteq \mathscr{E}(\gamma c_{out}) = \mathscr{E}(c_0)$. Thus, we have

$$\gamma C_{max,co} \subseteq \mathscr{E}(c_0) \subseteq C_{max,co}. \tag{D.33}$$

Since $\mathscr{E}(c_0)$ is λ_a-contractive, we have for any $k \geq 1$

$$Pre^k_{\mathscr{D}(\Sigma)}(\lambda_a^k \mathscr{E}(c_0), S_{xu} \times D) \supseteq \mathscr{E}(c_0) \supseteq \gamma C_{max,co}. \tag{D.34}$$

Since $C_{max,co} \supseteq \mathscr{E}(c_0)$,

$$\begin{aligned} &Pre^k_{\mathscr{D}(\Sigma)}(\lambda_a^k C_{max,co}, S_{xu} \times D) \\ &\supseteq Pre^k_{\mathscr{D}(\Sigma)}(\lambda_a^k \mathscr{E}(c_0), S_{xu} \times D) && \text{(D.35)} \end{aligned}$$

By (D.34) and (D.35),

$$Pre^k_{\mathscr{D}(\Sigma)}(\lambda_a^k C_{max,co}, S_{xu} \times D) \supseteq \gamma C_{max,co}. \tag{D.36}$$

By definition of N, $\lambda_a^N < \gamma$. Thus, $\lambda = \lambda_a^N/\gamma < 1$.

$$\begin{aligned} & Pre^N_{\mathscr{D}(\Sigma)}(\lambda\gamma C_{max,co}, S_{xu} \times D) \\ =& Pre^N_{\mathscr{D}(\Sigma)}(\lambda_a^N C_{max,co}, S_{xu} \times D) \supseteq \gamma C_{max,co}. \end{aligned} \tag{D.37}$$

That is, $\gamma C_{max,co}$ is N-step λ-contractive. □

Proof of Lemma 7.9. To prove the lemma, it suffices to show that $Pre^n_{\mathscr{D}(\Sigma)}(0, S_{xu} \times D)$ contains the origin in the interior. Since the system is controllable, there exists a feedback gain $K = [K_1\ K_2]$ such that the closed-loop system matrix $A_c = A + K_1 B + K_2 E$ has all eigenvalues equal to zero. That implies $A_c^n = 0$. In other words, under the control of $u = K_1 x$, $u_d = K_2 x$, the closed-loop system reaches the origin 0 at step n for all initial states $x_0 \in \mathbb{R}^n$. Now let us consider the unit ball $\mathscr{B}$ in $\mathbb{R}^n$. Since $S_{xu} \times D$ contains the origin in the interior, there exists a positive scalar $\xi > 0$ such that $\xi(A_c^k \mathscr{B} \times K A_c^k \mathscr{B})$ is contained in $S_{xu} \times D$ for k from 0 to $n-1$. That implies that $\xi\mathscr{B}$ is a subset of $Pre^n_{\mathscr{D}(\Sigma)}(\{0\}, S_{xu} \times D)$. Thus, $Pre^n_{\mathscr{D}(\Sigma)}(0, S_{xu} \times D)$ contains the origin in the interior.

Since $Pre^n_{\mathscr{D}(\Sigma)}(0, S_{xu} \times D)$ is a CIS of $\mathscr{D}(\Sigma)$ in $S_{xu} \times D$, the maximal CIS $C_{max,co}$ must contain $Pre^n_{\mathscr{D}(\Sigma)}(0, S_{xu} \times D)$ and thus contain the origin in the interior. Thus, there exists a positive scalar γ such that $\gamma C_{max,co}$ is contained in $Pre^n_{\mathscr{D}(\Sigma)}(0, S_{xu} \times D)$. □

APPENDIX E

Complementary Materials for Chapter 9

E.1 Proof of Theorem 9.1

In this section, we denote the distance $\gamma(\mathscr{P}(\mathbf{u},x),\Delta_{all})$ between $\mathscr{P}(\mathbf{u},x)$ and Δ_{all} by $r(x,\mathbf{u})$ for short.

Lemma E.1. For any given $(x,\alpha)\in\mathbb{R}^{n+1}$, the minimal distance $\inf_{\mathbf{u}} r(x,\mathbf{u})$ at x is less than or equal to $(1-\alpha^l)\mu(D)$ if and only if $x\in C_{max,\alpha}$. Furthermore, a controller $\mathbf{u}$ satisfies $r(x,\mathbf{u})\leq(1-\alpha^l)\mu(D)$ for all $x\in C_{max,\alpha}$ if and only if for all $x\in C_{max,\alpha}$,

$$\mathbf{u}(x)\in\mathscr{A}(x,C_{max,\alpha}), \tag{E.1}$$

where $\mathscr{A}(x,C_{max,\alpha})$ is the admissible input set of Σ_α.

Proof. We first show the "if" direction. Pick an arbitrary $x\in C_{max,\alpha}$. Since $C_{max,\alpha}$ is the maximal RCIS of Σ_α, there exist $\mathbf{u}:\mathbb{R}^n\to\mathbb{R}^m$ and $\mathbf{u}_d:\mathbb{R}^n\to(1-\alpha)D$ such that

$$\mathscr{R}^k_{\Sigma_\alpha}((x,\alpha),(\mathbf{u},\mathbf{u}_d),\Delta_\alpha)\subseteq S_{xu,\alpha},\ \forall k\geq 0. \tag{E.2}$$

Define the disturbance model $\Delta(\bar{x}):=\mathbf{u}_d(\bar{x})+\alpha D\in\mathscr{D}$ for all $\bar{x}\in\mathbb{R}^n$. By the construction of Σ_α and $S_{xu,\alpha}$, (E.2) implies that $\mathscr{R}^k_\Sigma(x,\mathbf{u},\Delta)\subseteq S_{xu}$ for all $k\geq 0$. That is, $\Delta\in\mathscr{P}(\mathbf{u},x)$. Hence,

$$\inf_{\overline{\mathbf{u}}} r(x,\overline{\mathbf{u}})\leq r(x,\mathbf{u})\leq\gamma(\Delta,\Delta_{all})=(1-\alpha^l)\mu(D), \tag{E.3}$$

where the last equality uses the property of Lebesgue measure $\mu(\alpha D)=\alpha^l\mu(D)$ (recall that $D\subseteq\mathbb{R}^l$). Also, by Section 9.1.1, we know that $\mathbf{u}$ satisfies (E.1). Hence, we proved the "if" direction of both statements in Lemma E.1.

Next, we pick an arbitrary (x,α) such that $\inf_{\overline{\mathbf{u}}} r(x,\overline{\mathbf{u}})\leq(1-\alpha^l)\mu(D)$. By the definition of $r(x,\mathbf{u})$, for any integer $i\geq 1/\alpha$, there exist $\mathbf{u}_i:\mathbb{R}^n\to\mathbb{R}^m$ and $\Delta_i:\mathbb{R}^n\to\mathscr{D}$ such that $\Delta_i\in\mathscr{P}(\mathbf{u}_i,x)$

and $\gamma(\Delta_i, \Delta_{all}) < (1-\alpha_i^l)\mu(D)$, with $\alpha_i := \alpha - 1/i > 0$. By the definition of $\mathscr{D}$ in (9.7), there exists $\mathbf{u}_d : \mathbb{R}^n \to (1-\alpha_i)D$ such that $\mathbf{u}_d(\bar{x}) + \alpha_i D \subseteq \Delta_i(\bar{x})$ for all $\bar{x} \in \mathbb{R}^n$. Hence, $\Delta_i \in \mathscr{P}(\mathbf{u}, x)$ implies that the disturbance model $\mathbf{u}_d + \alpha_i D$ is in $\mathscr{P}(\mathbf{u}, x)$ as well. That is, $\mathscr{R}_\Sigma^k(x, \mathbf{u}_i, \mathbf{u}_d + \alpha_i D) \subseteq S_{xu}$ for all $k \geq 0$, which is further equivalent to

$$\mathscr{R}_{\Sigma_{\alpha_i}}^k(x, (\mathbf{u}_i, \mathbf{u}_d), \Delta_{\alpha_i}) \subseteq S_{xu,\alpha_i},\ \forall k \geq 0. \tag{E.4}$$

By definition, (E.4) implies that $x \in C_{max,\alpha_i}$, that is, $(x, \alpha_i) \in C_{max,[0,1]}$. Since $C_{max,[0,1]}$ is closed, we know that $(x, \alpha) = \lim_{i\to\infty}(x, \alpha_i) \in C_{max,[0,1]}$ as well. This completes the proof for the first statement in Lemma E.1.

Now suppose that $\mathbf{u}$ is a controller satisfying $r(x, \mathbf{u}) \leq (1-\alpha^l)\mu(D)$ for all $x \in C_{max,\alpha}$. We pick an arbitrary $x \in C_{max,\alpha}$. Clearly, $(x, \mathbf{u}(x)) \in S_{xu}$. For all $i \geq 1/\alpha$, there exists $\Delta_i : \mathbb{R}^n \to \mathscr{D}$ such that $\Delta_i \in \mathscr{P}(\mathbf{u}, x)$ and $\gamma(\Delta_i, \Delta_{all}) < (1-\alpha_i^l)\mu(D)$. Thus, for all $d \in \Delta_i(x)$,

$$\mathscr{R}_\Sigma^k(Ax + B\mathbf{u}(x) + Ed, \mathbf{u}, \Delta_i) \subseteq S_{xu},\ \forall k \geq 0, \tag{E.5}$$

which implies that $r(Ax + B\mathbf{u}(x) + Ed, \mathbf{u}) \leq (1-\alpha_i^l)\mu(D)$ for all $d \in \Delta_i(x)$. Based on the first statement of Lemma E.1, we have $(Ax + B\mathbf{u}(x) + Ed, \alpha_i) \in C_{max,[0,1]}$ for all $d \in \Delta_i(x)$. Since $\gamma(\Delta_i, \Delta_{all}) < \mu(D) - \mu(\alpha_i D)$, there exists $u_{d,i} \in (1-\alpha_i)D$ such that $u_{d,i} + \alpha_i D \subseteq \Delta_i(x)$. Thus, we have $(Ax + B\mathbf{u}(x) + E(u_{d,i} + d), \alpha_i) \in C_{max,[0,1]}$ for all $d \in \alpha_i D$. Since D is compact, there exists a subsequence of $u_{d,i}$ that converges to a point $u_d \in (1-\alpha)D$. We abuse the notation a bit and denote this subsequence by $u_{d,i}$ again. Then, we have $Ax + B\mathbf{u}(x) + E(u_d + \alpha D) \subseteq C_{max,\alpha}$ and $(x, \mathbf{u}(x)) \in S_{xu}$, which implies $\mathbf{u}(x) \in \mathscr{A}(x, C_{max,\alpha})$. □

Proof of Theorem 9.1 . Point (i) of Problem 9.2 is trivially satisfied by u^* since (9.9) implies (9.5) with respect to $C_{max,1}$ for $x \in C_{max,1}$. For point (ii), according to (9.10), a controller $\mathbf{u}^*$ minimizes $r(x, \cdot)$ for all $x \in \mathbb{R}^n$ if and only if $r(x, \mathbf{u}^*) \leq (1-\alpha^*(x)^l)\mu(D)$ for all $x \in C_{max,0}$, which is equivalent to the condition in (E.1) due to Lemma E.1. Equation (9.10) is just a direct application of Lemma E.1. □

BIBLIOGRAPHY

[1] *Road Design Manual*. Michigan Department of Transportation, 2019.

[2] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008.

[3] A. A. Ahmadi and O. Gunluk. Robust-to-dynamics optimization. *arXiv preprint arXiv:1805.03682*, 2018.

[4] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pages 3420–3431. IEEE, 2019.

[5] 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, 2016.

[6] B. Aminof, G. De Giacomo, A. Lomuscio, A. Murano, S. Rubin, et al. Synthesizing strategies under expected and exceptional environment behaviors. In *IJCAI*, pages 1674–1680. ijcai. org, 2020.

[7] T. Anevlavis, Z. Liu, N. Ozay, and P. Tabuada. Controlled invariant sets: implicit closed-form representations and applications. *arXiv preprint arXiv:2107.08566*, 2021.

[8] T. Anevlavis, Z. Liu, N. Ozay, and P. Tabuada. An enhanced hierarchy for (robust) controlled invariance. In *2021 American Control Conference (ACC)*, pages 4860–4865. IEEE, 2021.

[9] T. Anevlavis and P. Tabuada. Computing controlled invariant sets in two moves. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6248–6254. IEEE, 2019.

[10] T. Anevlavis and P. Tabuada. A simple hierarchy for computing controlled invariant sets. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, pages 1–11, 2020.

[11] P. J. Antsaklis and A. N. Michel. *Linear systems*, pages 286–288. Springer Science & Business Media, 2006.

[12] Z. Artstein. Linear systems with delayed controls: A reduction. *IEEE Transactions on Automatic control*, 27(4):869–879, 1982.

[13] Z. Artstein and S. V. Raković. Set invariance under output feedback: a set-dynamics approach. *International Journal of Systems Science*, 42(4):539–555, 2011.

[14] M. Arwashan, T. Ge, Z. Liu, and N. Ozay. Driving with guardian: Blending user inputs with safety ensuring barriers. In *2020 IEEE Conference on Control Technology and Applications (CCTA)*, pages 326–333. IEEE, 2020.

[15] N. Athanasopoulos, K. Smpoukis, and R. M. Jungers. Safety and invariance for constrained switching systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6362–6367. IEEE, 2016.

[16] J.-P. Aubin. A survey of viability theory. *SIAM Journal on Control and Optimization*, 28(4):749–788, 1990.

[17] A. Bajcsy, S. Bansal, E. Bronstein, V. Tolani, and C. J. Tomlin. An efficient reachability-based framework for provably safe autonomous navigation in unknown environments. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 1758–1765. IEEE, 2019.

[18] H. Balim, A. Aspeel, Z. Liu, and N. Ozay. Koopman-inspired implicit backward reachable sets for unknown nonlinear systems. *IEEE Control Systems Letters*, 2023.

[19] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *CDC*, pages 2242–2253. IEEE, 2017.

[20] D. Bertsekas. Infinite time reachability of state-space regions by using feedback control. *IEEE Transactions on Automatic Control*, 17(5):604–613, 1972.

[21] N. Birla and A. Swarup. Optimal preview control: A review. *Optimal Control Applications and Methods*, 36(2):241–268, 2015.

[22] 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.

[23] F. Blanchini and S. Miani. *Set-theoretic methods in control*, volume 78. Springer, 2008.

[24] S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[25] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.

[26] P. Brunovský. A classification of linear controllable systems. *Kybernetika*, 6:173–188, 1970.

[27] P. Caravani and E. De Santis. Doubly invariant equilibria of linear discrete-time games. *Automatica*, 38(9):1531–1538, 2002.

[28] D. S. Carrasco and G. C. Goodwin. Feedforward model predictive control. *Annual Reviews in Control*, 35(2):199–206, 2011.

[29] G. Chou, Y. E. Sahin, L. Yang, K. J. Rutledge, P. Nilsson, and N. Ozay. Using control synthesis to generate corner cases: A case study on autonomous driving. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11):2906–2917, 2018.

[30] M. Cwikel and P.-O. Gutman. Convergence of an algorithm to find maximal state constraint sets for discrete-time linear dynamical systems with bounded controls and states. *IEEE Transactions on Automatic Control*, 31(5):457–459, 1986.

[31] M. S. Darup and M. Mönnigmann. On general relations between null-controllable and controlled invariant sets for linear constrained systems. In *53rd IEEE Conference on Decision and Control*, pages 6323–6328. IEEE, 2014.

[32] E. De Santis, M. D. Di Benedetto, and L. Berardi. Computation of maximal safe sets for switching systems. *IEEE Transactions on Automatic Control*, 49(2):184–195, 2004.

[33] Z. Du, Z. Liu, J. Weitze, and N. Ozay. Sample complexity analysis and self-regularization in identification of over-parameterized arx models. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 6026–6033. IEEE, 2022.

[34] J. Duncan Glover and F. C. Schweppe. Control of linear dynamic systems with set constrained disturbances. *Automatic Control, IEEE Transactions on*, 16:411 – 423, 11 1971.

[35] P. Falcone, M. Ali, and J. Sjoberg. Predictive threat assessment via reachability analysis and set invariance theory. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1352–1361, 2011.

[36] M. Fiacchini and M. Alamir. Computing control invariant sets is easy. *arXiv preprint arXiv:1708.04797*, 2017.

[37] B. Finkbeiner and N. Passing. Synthesizing dominant strategies for liveness. In *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[38] Y. Gao, K. H. Johansson, and L. Xie. Computing probabilistic controlled invariant sets. *IEEE Transactions on Automatic Control*, 66(7):3138–3151, 2020.

[39] Y. Gao, C. Liu, and K. H. Johansson. Robust risk-aware model predictive control of linear systems with bounded disturbances. In *CDC*, pages 1148–1155. IEEE, 2022.

[40] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[41] 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.

[42] J. Grzybowski and R. Urbański. Order cancellation law in the family of bounded convex sets. *Journal of Global Optimization*, 77(2):289–300, 2020.

[43] 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, 2018.

[44] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.

[45] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames. A scalable safety critical control framework for nonlinear systems. *IEEE Access*, 8:187249–187275, 2020.

[46] P.-O. Gutman and M. Cwikel. An algorithm to find maximal state constraint sets for discrete-time linear dynamical systems with bounded controls and states. *IEEE Transactions on Automatic Control*, 32(3):251–254, 1987.

[47] S. Herbert, J. J. Choi, S. Sanjeev, M. Gibson, K. Sreenath, and C. J. Tomlin. Scalable learning of safety guarantees for autonomous systems using hamilton-jacobi reachability. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5914–5920. IEEE, 2021.

[48] M. Herceg, M. Kvasnica, C. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland, July 17–19 2013. http://control.ee.ethz.ch/~mpt.

[49] M. Holtmann, Ł. Kaiser, and W. Thomas. Degrees of lookahead in regular infinite games. In *International Conference on Foundations of Software Science and Computational Structures*, pages 252–266. Springer, 2010.

[50] M. Jalalmaab, B. Fidan, S. Jeon, and P. Falcone. Guaranteeing persistent feasibility of model predictive motion planning for autonomous vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 843–848. IEEE, 2017.

[51] S. Jang, N. Ozay, and J. L. Mathieu. Large-scale invariant sets for safe coordination of thermostatic loads. In *2021 American Control Conference (ACC)*, pages 4163–4170. IEEE, 2021.

[52] M. Jankovic. Control barrier functions for constrained control of linear systems with input delay. In *2018 Annual American Control Conference (ACC)*, pages 3316–3321. IEEE, 2018.

[53] M. Jankovic. Robust control barrier functions for constrained stabilization of nonlinear systems. *Automatica*, 96:359–367, 2018.

[54] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pages 1620–1626. IEEE, 2003.

[55] T. Katayama, T. Ohki, T. Inoue, and T. Kato. Design of an optimal controller for a discrete-time system subject to previewable demand. *International Journal of Control*, 41(3):677–699, 1985.

[56] E. C. Kerrigan and J. M. Maciejowski. Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 5, pages 4951–4956. IEEE, 2000.

[57] R. Kianfar, P. Falcone, and J. Fredriksson. Safety verification of automated driving systems. *IEEE Intelligent Transportation Systems Magazine*, 5(4):73–86, 2013.

[58] F. Klein and M. Zimmermann. How much lookahead is needed to win infinite games? In *International Colloquium on Automata, Languages, and Programming*, pages 452–463. Springer, 2015.

[59] I. Kolmanovsky, E. Garone, and S. Di Cairano. Reference and command governors: A tutorial on their theory and automotive applications. In *2014 American Control Conference*, pages 226–241. IEEE, 2014.

[60] M. Korda and I. Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

[61] M. Korda and I. Mezić. Optimal construction of koopman eigenfunctions for prediction and control. *IEEE Transactions on Automatic Control*, 65(12):5114–5129, 2020.

[62] O. Kupferman, D. Sadigh, and S. A. Seshia. Synthesis with clairvoyance. In *Haifa Verification Conference*, pages 5–19. Springer, 2011.

[63] J. Laks, L. Pao, E. Simley, A. Wright, N. Kelley, and B. Jonkman. Model predictive control using preview measurements from lidar. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 813, 2011.

[64] A. J. Laub. *Matrix analysis for scientists and engineers*, volume 91. Siam, 2005.

[65] M. Lazar and V. Spinu. Finite-step terminal ingredients for stabilizing model predictive control. *IFAC-PapersOnLine*, 48(23):9–15, 2015. 5th IFAC Conference on Nonlinear Model Predictive Control NMPC 2015.

[66] J. Lee, J. Kim, and A. D. Ames. Hierarchical relaxation of safety-critical controllers: Mitigating contradictory safety conditions with application to quadruped robots. *arXiv preprint arXiv:2305.03929*, 2023.

[67] B. Legat, P. Tabuada, and R. M. Jungers. Computing controlled invariant sets for hybrid systems with applications to model-predictive control. *IFAC-PapersOnLine*, 51(16):193–198, 2018.

[68] D. Li, M. L. Shehab, Z. Liu, N. Aréchiga, J. DeCastro, and N. Ozay. Outlier-robust inverse reinforcement learning and reward-based detection of anomalous driving behaviors. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 4175–4182. IEEE, 2022.

[69] N. Li, Y. Li, and I. Kolmanovsky. A unified safety protection and extension governor. *arXiv preprint arXiv:2304.07984*, 2023.

[70] Y. Li, X. Chen, and N. Li. Online optimal control with linear dynamics and predictions: Algorithms and regret analysis. *Advances in Neural Information Processing Systems*, 32, 2019.

[71] Y. Li and J. Ibanez-Guzman. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61, 2020.

[72] Y. Li and J. Liu. Computing maximal invariant sets for switched nonlinear systems. In *2016 IEEE conference on computer aided control system design (CACSD)*, pages 862–867. IEEE, 2016.

[73] Y. Li, Z. Sun, and J. Liu. Rocs 2.0: An integrated temporal logic control synthesis tool for nonlinear dynamical systems. *IFAC-PapersOnLine*, 54(5):31–36, 2021.

[74] D. Limon, T. Alamo, and E. F. Camacho. Enlarging the domain of attraction of mpc controllers. *Automatica*, 41(4):629–635, 2005.

[75] Z. Liu, T. Anevlavis, N. Ozay, and P. Tabuada. Automaton-based implicit controlled invariant set computation for discrete-time linear systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 1112–1119. IEEE, 2021.

[76] Z. Liu, H. Chen, Y. Gao, and N. Ozay. Opportunistic safety outside the maximal controlled invariant set. 2023. submitted for journal publication, under review.

[77] Z. Liu and N. Ozay. Safety control with preview automaton. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 1557–1564. IEEE, 2019.

[78] Z. Liu and N. Ozay. On the value of preview information for safety control. In *2021 American Control Conference (ACC)*, pages 2348–2354. IEEE, 2021.

[79] Z. Liu and N. Ozay. Safe online planning in unknown nonconvex environments with implicit controlled invariant sets. *IFAC-PapersOnLine*, 54(5):163–168, 2021.

[80] Z. Liu and N. Ozay. On the convergence of the backward reachable sets of robust controlled invariant sets for discrete-time linear systems. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 4270–4275. IEEE, 2022.

[81] Z. Liu and N. Ozay. Quantifying the value of preview information for safety control. *arXiv preprint arXiv:2303.10660*, 2023.

[82] Z. Liu, N. Ozay, and E. D. Sontag. On the non-existence of immersions for systems with multiple omega-limit sets. In *IFAC world congress*, pages 81–85, 2023.

[83] Z. Liu, L. Yang, and N. Ozay. Scalable computation of controlled invariant sets for discrete-time linear systems with input delays. In *2020 American Control Conference (ACC)*, pages 4722–4728. IEEE, 2020.

[84] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[85] W. Lombardi, S. Olaru, G. Bitsoris, and S.-I. Niculescu. Cyclic invariance for discrete time-delay systems. *Automatica*, 48(10):2730–2733, 2012.

[86] W. Lombardi, S. Olaru, M. Lazar, and S.-I. Niculescu. On positive invariance for delay difference equations. In *Proceedings of the 2011 American Control Conference*, pages 3674–3679. IEEE, 2011.

[87] G. Mamakoukas, S. Di Cairano, and A. P. Vinod. Robust model predictive control with data-driven koopman operators. In *2022 American Control Conference (ACC)*, pages 3885–3892. IEEE, 2022.

[88] D. Mayne, M. Seron, and S. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[89] E. J. N. Menezes, A. M. Araújo, and N. S. B. Da Silva. A review on wind turbine control and its associated methods. *Journal of cleaner production*, 174:945–953, 2018.

[90] L. P. Nilsson. *Correct-by-construction control synthesis for high-dimensional systems*. PhD thesis, 2017.

[91] P. Nilsson, O. Hussien, A. Balkan, Y. Chen, A. D. Ames, J. W. Grizzle, N. Ozay, H. Peng, and P. Tabuada. Correct-by-construction adaptive cruise control: Two approaches. *IEEE Transactions on Control Systems Technology*, 24(4):1294–1307, 2015.

[92] P. Nilsson and N. Ozay. Provably-correct compositional synthesis of vehicle safety systems. In *Safe, Autonomous and Intelligent Vehicles*, pages 97–122. Springer, 2019.

[93] P. Nilsson, N. Özay, U. Topcu, and R. M. Murray. Temporal logic control of switched affine systems with an application in fuel balancing. In *2012 American Control Conference (ACC)*, pages 5302–5309. IEEE, 2012.

[94] S. Olaru and S.-I. Niculescu. Predictive control for linear systems with delayed input subject to constraints. *IFAC Proceedings Volumes*, 41(2):11208–11213, 2008.

[95] G. Orosz and A. D. Ames. Safety functionals for time delay systems. In *2019 American Control Conference (ACC)*, pages 4374–4379. IEEE, 2019.

[96] H. Parwana, R. Wang, and D. Panagou. Algorithms for finding compatible constraints in receding-horizon control of dynamical systems, 2023.

[97] H. Peng and M. Tomizuka. Preview control for vehicle lateral guidance in highway automation. *Journal of Dynamic Systems, Measurement, and Control*, 115(4):679–686, 1993.

[98] S. V. Rakovic and M. Baric. Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks. *IEEE Transactions on Automatic Control*, 55(7):1599–1614, 2010.

[99] S. V. Rakovic, 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), 2005.

[100] 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.

[101] S. Resnick. *A probability path*. Springer, 2019.

[102] M. Rungger, M. Mazo Jr, and P. Tabuada. Specification-guided controller synthesis for linear systems and safe linear-time temporal logic. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 333–342, 2013.

[103] M. Rungger and P. Tabuada. Computing robust controlled invariant sets of linear systems. *IEEE Transactions on Automatic Control*, 62(7):3665–3670, 2017.

[104] S. Sadraddini and R. Tedrake. Linear encodings for polytope containment problems. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4367–4372. IEEE, 2019.

[105] Y. E. Sahin, Z. Liu, K. Rutledge, D. Panagou, S. Z. Yong, and N. Ozay. Intention-aware supervisory control with driving safety applications. In *2019 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1–8. IEEE, 2019.

[106] S. Samuel, K. Mallik, A.-K. Schmuck, and D. Neider. Resilient abstraction-based controller design. In *CDC*, pages 2123–2129. IEEE, 2020.

[107] A. Scholbrock, P. Fleming, D. Schlipf, A. Wright, K. Johnson, and N. Wang. Lidar-enhanced wind turbine control: Past, present, and future. In *2016 American Control Conference (ACC)*, pages 1399–1406. IEEE, 2016.

[108] T. B. Sheridan. Three models of preview control. *IEEE Transactions on Human Factors in Electronics*, (2):91–102, 1966.

[109] S. Shin and V. M. Zavala. Controllability and observability imply exponential decay of sensitivity in dynamic optimization. *IFAC-PapersOnLine*, 54(6):179–184, 2021.

[110] A. Singletary, P. Nilsson, T. Gurriet, and A. D. Ames. Online active safety for robotic manipulators. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 173–178. IEEE, 2019.

[111] M. Sinner, V. Petrović, A. Langidis, L. Neuhaus, M. Hölling, M. Kühn, and L. Y. Pao. Experimental testing of a preview-enabled model predictive controller for blade pitch control of wind turbines. *IEEE Transactions on Control Systems Technology*, 30(2):583–597, 2021.

[112] S. W. Smith, P. Nilsson, and N. Ozay. Interdependence quantification for compositional control synthesis with an application in vehicle safety systems. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 5700–5707. IEEE, 2016.

[113] 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, 2014.

[114] M. Tomizuka and D. Whitney. Optimal discrete finite preview problems (why and how is future information important?). *Journal of Dynamic Systems, Measurement, and Control*, 97(4):319–325, 1975.

[115] S. Tonkens, A. Toofanian, Z. Qin, S. Gao, and S. Herbert. Patching neural barrier functions using hamilton-jacobi reachability. *arXiv preprint arXiv:2304.09850*, 2023.

[116] J. Tumova, G. C. Hall, S. Karaman, E. Frazzoli, and D. Rus. Least-violating control strategy synthesis with safety rules. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 1–10, 2013.

[117] K. P. Wabersich and M. N. Zeilinger. Linear model predictive safety certification for learning-based control. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 7130–7135. IEEE, 2018.

[118] P.-B. Wieber. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 137–142. IEEE, 2006.

[119] A. Wintenberg and N. Ozay. Implicit invariant sets for high-dimensional switched affine systems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3291–3297. IEEE, 2020.

[120] S. Xu and H. Peng. Design, analysis, and experiments of preview path tracking control for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):48–58, 2019.

[121] L. Yang, D. Rizzo, M. Castanier, and N. Ozay. Parameter sensitivity analysis of controlled invariant sets via value iteration. In *ACC*, pages 4737–4744. IEEE, 2020.

[122] C. Yu, G. Shi, S.-J. Chung, Y. Yue, and A. Wierman. The power of predictions in online control. *Advances in Neural Information Processing Systems*, 33:1994–2004, 2020.

[123] M. Zimmermann. Finite-state strategies in delay games. *arXiv preprint arXiv:1709.03539*, 2017.