THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Control of Constrained Dynamical Systems with Performance Guarantees

With Application to Vehicle motion Control

Ankit Gupta

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2021

Control of Constrained Dynamical Systems with Performance Guarantees

With Application to Vehicle motion Control

ANKIT GUPTA

ISBN 978-91-7905-437-3

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 4904
ISSN 0346-718X
This thesis has been prepared using LaTeX.

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
Email: ankit.gupta@chalmers.se; ankitgupta47@gmail.com

Cover:
An illustration by the author of a dynamical system operating in constrained environment.

Printed by Chalmers Reproservice
Gothenburg, Sweden, January 2021

To my Family, Pallavi and Friends

Abstract

In control engineering, models of the system are commonly used for controller design. A standard control design problem consists of steering the given system output (or states) towards a predefined reference. Such a problem can be solved by employing feedback control strategies. By utilizing the knowledge of the model, these strategies compute the control inputs that shrink the error between the system outputs and their desired references over time. Usually, the control inputs must be computed such that the system output signals are kept in a desired region, possibly due to design or safety requirements. Also, the input signals should be within the physical limits of the actuators. Depending on the constraints, their violation might result in unacceptable system failures (e.g. deadly injury in the worst case). Thus, in safety-critical applications, a controller must be robust towards the modelling uncertainties and provide a priori guarantees for constraint satisfaction.

A fundamental tool in constrained control application is the robust control invariant sets (RCI). For a controlled dynamical system, if initial states belong to RCI set, control inputs always exist that keep the future state trajectories restricted within the set. Hence, RCI sets can characterize a system that never violates constraints. These sets are the primary ingredient in the synthesis of the well-known constraint control strategies like model predictive control (MPC) and interpolation-based controller (IBC). Consequently, a large body of research has been devoted to the computation of these sets. In the thesis, we will focus on the computation of RCI sets and the method to generate control inputs that keep the system trajectories within RCI set. We specifically focus on the systems which have time-varying dynamics and polytopic constraints. Depending upon the nature of the time-varying element in the system description (i.e., if they are observable or not), we propose different sets of algorithms.

The first group of algorithms apply to the system with time-varying, bounded uncertainties. To systematically handle the uncertainties and reduce conservatism, we exploit various tools from the robust control literature to derive novel conditions for invariance. The obtained conditions are then combined with a newly developed method for volume maximization and minimization in a convex optimization problem to compute desirably large and small RCI sets. In addition to ensuring invariance, it is also possible to guarantee desired closed-loop performance within the RCI set. Furthermore, developed algo-

rithms can generate RCI sets with a predefined number of hyper-planes. This feature allows us to adjust the computational complexity of MPC and IBC controller when the sets are utilized in controller synthesis. Using numerical examples, we show that the proposed algorithms can outperform (volume-wise) many state-of-the-art methods when computing RCI sets.

In the other case, we assume the time-varying parameters in system description to be observable. The developed algorithm has many similar characteristics as the earlier case, but now to utilize the parameter information, the control law and the RCI set are allowed to be parameter-dependent. We have numerically shown that the presented algorithm can generate invariant sets which are larger than the maximal RCI sets computed without exploiting parameter information.

Lastly, we demonstrate how we can utilize some of these algorithms to construct a computationally efficient IBC controller for the vehicle motion control. The devised IBC controller guarantees to meet safety requirements mentioned in ISO 26262 and the ride comfort requirement by design.

Keywords: Invariant set, Linear parameter varying system, Linear matrix inequalities, Semi-definite program, Linear fractional transformation, Robust control.

List of Publications

This thesis is based on the following publications:

[A] **A. Gupta**, H. Köroğlu and P. Falcone, "Restricted-complexity characterization of control-invariant domains with application to lateral vehicle dynamics control". Published in CDC 2017.

[B] **A. Gupta**, H. Köroğlu and P. Falcone, "Computation of low-complexity control-invariant sets for systems with uncertain parameter dependence". Published in Automatica 2019.

[C] **A. Gupta** and P. Falcone, "Full-Complexity Characterization of Control-Invariant Domains for Systems With Uncertain Parameter Dependence". IEEE Control Systems Letters 2019.

[D] **A. Gupta**, Hakan Köroğlu and P. Falcone, "Computation of Robust Control Invariant Sets with Predefined Complexity for Uncertain Systems". International Journal of Robust and Nonlinear Control 2020 (accepted).

[E] **A. Gupta**, M. Mejari, P. Falcone, D. Piga, "Computation of Parameter Dependent Invariant Sets for LPV Systems with Guaranteed Performance". Automatica 2020 (submitted).

Other publications by the author, not included in this thesis, are:

[F] L. Ni, **A. Gupta**, P. Falcone, L. Johannesson, "Vehicle Lateral Motion Control with Performance and Safety Guarantees". *IFAC Symposium on Advances in Automotive Control (AAC)*, Volume 49, Issue 11, 2016, Pages 285-290.

[G] **A. Gupta** and P. Falcone, " Low-Complexity Explicit MPC Controller for Vehicle Lateral Motion Control". *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2839-2844.

[H] E. Klintberg, M. Nilsson, L. J. Mårdh and **A. Gupta**, "A Primal Active-Set Minimal-Representation Algorithm for Polytopes with Application to Invariant-Set Calculations". *IEEE Conference on Decision and Control (CDC)*, 2018, pp. 6862-6867.

[I] **A. Gupta**, M. Nilsson, P. Falcone, E. Klintberg and L. J. Mårdh, "A Framework for Vehicle Lateral Motion Control With Guaranteed Tracking and Performance". *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3607-3612.

[J] E. Klintberg, M. Nilsson, **A. Gupta**, L. J. Mårdh and P. Falcone, "Tree-Structured Polyhedral Invariant Set Calculations". *IEEE Control Systems Letters*, vol. 4, no. 2, pp. 426-431, April 2020.

Acknowledgments

With my PhD studies coming to an end, I will like to take a moment to acknowledge all the people who made this possible and helped me to the end of this journey. First and foremost, I express my sincere gratitude to my supervisors Prof. Paolo Falcone and Dr. Lars Johannesson Mårdh for giving me the opportunity to pursue my PhD studies at Chalmers. I am thankful to them for proposing an exciting research problem and providing me with needed guidance. I am also grateful to Prof. Jonas Sjöberg for accepting me in the Mechatronics research group and the Vinnova FFI program for financial support.

I will always be grateful to Prof. Hakan Köroğlu from whom I learned a lot during the PhD studies. I always enjoyed the collaboration with you, and all the technical and general discussions. I also thank Dr. Emil Klintberg and Dr. Magnus Nilsson for all the great ideas and knowledge they shared with me during the research, which helped me improve my understanding. A big thank to all in Zenseact's lateral control group for allowing me to be part of their group and always being welcoming.

I especially thank my good friend Ramin with whom I started my PhD journey and shared my office. Thanks for all the funny moments we had and many interesting discussions. A huge thank to Ivo and Muddassar for being great friends, chats with you were always delightful. I am deeply grateful to my friends Robert, Giuseppe, Simon, Fredrik, Elena, Karim, Masoud, Angelos, Stefan, Remi, Sten, Ahad and rest of the past and present Syscon group members. I will always miss the interesting conversation we had during lunch and Fika. You guys truly made my time at Chalmers memorable.

I want to thank my Masters Supervisors Prof. N. M. Singh and Prof. Faruk Kazi from VJTI Mumbai, to motivate me to do PhD studies and provide the necessary support during those days. Great thanks to my friends from VJTI days Aniket, Manas, Mandar and Vishwanath. You guys have always been fantastic and looking forward to having more crazy trips and experiences with all of you.

I was lucky to meet many lovely people who became my family in Sweden. I will always be thankful to Mamta and Nikul for providing me with all the needed help and support after moving to Sweden, in one way or other you have contributed to my achievement. A special thanks to Gurpreet, Harveen, Ganesh, Poonam, Nazmun, Debayani, Nalin, Monika, Vivek and Mamta for

all the awesome dinners, chai talks and the time we spent together.

Finally, the biggest recognition goes to my Family and Pallavi. Words cannot express my love and appreciation for them, without their endless support, encouragement and patience, this journey would not have been possible.

Ankit Gupta,
Göteborg, January 2021.

Acronyms

CI:	Control Invariant
RCI:	Robust Control Invariant
MRCI:	Maximal Robust Control Invariant
mRCI:	minimal Robust Control Invariant
LC-RCI:	Low-Complexity RCI
PI:	Positive Invariant
RPI:	Robust Positive Invariant
MPC:	Model Predicitve Control
IBC:	Interpolation Based Control
SDP:	Semi-Definite Program
QCQP:	Quadratically Constrained Quadratic Program
QP:	Quadratic Program
LP:	Linear Program
LMI:	Linear Matrix Inequality
LTI:	Linear Time Invariant
LPV:	Linear Parameter Varying
LFT:	Linear Fractional Transform

Contents

Part I

Background

CHAPTER 1

Overview

1.1 Introduction

Dynamical models can describe a wide variety of engineering systems and processes. These models describe the dynamic relations between the system input and output signals. One could use these models to study various system properties, e.g., controllability, observability and stability, as well as they can be used for the controller synthesis. However, it may not always be possible to model every system detail precisely. This is the case, for instance, when physical parameters in the model are unknown. Or when the model is too complex to be used in a control design framework. Thus, there might be a mismatch between the output of a model and the real system. Usually, these mismatches can be treated as bounded static or time-varying uncertainties, and then added to the system description to model the worst-case system behaviour. Overall, uncertainties are an inherent part of the system description, and a reasonable assumption should be made about their properties to be able to synthesize robust controllers.

A standard controller design problem involves stabilizing the given dynamical system towards a predefined reference using feedback control strategies.

By utilizing the available knowledge of the model, these strategies compute input that shrinks the error between the system outputs and their desired references over time. Usually, the control inputs must be computed such that the system output signals are kept in a desired region, possibly due to design or safety requirements. Also, the input signals should be within the physical limits of the actuators. Depending on the constraints, their violation might result in unacceptable system failures (e.g. deadly injury in the worst case). Thus, in safety-critical applications, a controller must be robust towards the modelling uncertainties and provide a priori guarantees for constraint satisfaction.

A fundamental tool in constrained control application is the robust control invariant sets (RCI). For a controlled dynamical system, if initial states belong to RCI set, control inputs always exist that keep the future state trajectories restricted within the set. Hence, RCI sets can characterize a system that never violates constraints. In the thesis, we will focus on the computation of RCI sets and the method to generate control inputs that keep the system trajectories within RCI set. In the next sections, we will learn about RCI sets in more detail.

1.2 Invariant Sets

Consider a control problem in which we want to control a discrete time nonlinear system described by the model

$$x(k+1) = f(x(k), u(k), w(k)), \tag{1.1}$$

where $x(k)$, $u(k)$ and $w(k)$ are the current state vector, control input and the unmeasured disturbance (or uncertainty) vectors, and $x(k+1)$ is the successor state vector, subject to the constraints $x(k) \in \boldsymbol{\mathcal{X}}$ and $u(k) \in \boldsymbol{\mathcal{U}}$ for all $w(k) \in \boldsymbol{\mathcal{W}}$ and for all $k \geq 0$.

We can now formally define RCI sets as [1]

Definition 1: *A set* $\boldsymbol{\Omega} \subseteq \boldsymbol{\mathcal{X}}$ *is an RCI set for the system* (1.1) *if*

$$x(k) \in \boldsymbol{\Omega} \Rightarrow \exists u(k) \in \boldsymbol{\mathcal{U}} : x(k+1) \in \boldsymbol{\Omega}, \forall w(k) \in \boldsymbol{\mathcal{W}}. \tag{1.2}$$

Roughly speaking, for the states in $\boldsymbol{\Omega}$, an admissible control input can be found such that the system evolves within $\boldsymbol{\Omega}$.

An another type of set, which is closely related to the RCI set is the λ-Contractive set. For some $\lambda \in (0, 1]$, we say a set $\mathbf{\Omega}_\lambda$ is λ-Contractive if

$$x(k) \in \mathbf{\Omega}_\lambda \Rightarrow \exists u(k) \in \boldsymbol{\mathcal{U}} : x(k+1) \in \lambda\mathbf{\Omega}_\lambda, \forall w(k) \in \boldsymbol{\mathcal{W}}. \tag{1.3}$$

For $\lambda = 1$, $\mathbf{\Omega}_1 = \mathbf{\Omega}$ i.e., $\mathbf{\Omega}_1$ is RCI. However, for $0 < \lambda < 1$, λ-Contractiveness becomes a stronger condition than invariance since it requires the system trajectories to move into the interior of the set $\mathbf{\Omega}_\lambda$ at the subsequent time step. Note that similar sets can be defined for the autonomous systems by removing the existence condition on the input. Nevertheless, the autonomous systems are not the focus of the thesis.

RCI and λ-Contractive sets are widely used in various control applications. Mostly, we are interested in the largest (maximal) and smallest (minimal) RCI set for a given system due to their various properties, some of which are listed below

1.2.1 Applications of RCI sets

- **Stability Analysis**:The maximal RCI set contains all the initial states $x(0)$ whose future trajectories can be restricted within the constraint set and possibly converge towards the equilibrium point. Further, λ-Contractive sets (with $0 < \lambda < 1$) characterize all the initial states for which convergence towards the equilibrium point is guaranteed [1].

  Apart from characterizing initial states from where the system can be stabilized, these sets can also be utilized to construct Lyapunov functions. Lyapunov functions are widely used to analyze the stability properties of a system, [2]–[4]. However, it is not always easy to construct Lyapunov functions itself for a given system. An early link between the Lyapunov functions and the invariance sets was investigated by J.P. LaSalle in his famously known result, LaSalle's invariance principle [5]. From [1], it is known that the level sets of Lyapunov functions are in fact invariant set. Thus, if we can compute RCI set for the system, we can also construct Lyapunov functions (see, [1], [6]–[8]).

  In the presence of additive disturbance in the system dynamics, the system trajectories may never stay at the equilibrium point. Thus, analogous to asymptotic stability, instead of a unique point, the system trajectories converge to a set, which is also known as the minimal RCI set or ultimate bounded set [9]–[11]. For the controlled system, it is desirable to have this set as small as possible.

- **Controller Synthesis**: In the control schemes like model predictive control (MPC), RCI sets are used as a terminal constraint to guarantee the persistent feasibility of the underlying optimization problem [12], [13]. These sets directly impact the various properties of the MPC controller. For example, the size of the feasibility set of the controller is directly proportional to the volume of the RCI set. Also, since these sets are used as a constraint in MPC optimization problem, they directly affect the computational complexity of the MPC controller [14].

  The RCI set with the minimum volume also has an essential role in robust control. For example, in [15]–[17], it is used to generate an invariant tube around a nominal trajectory. In such an application, the goal is to keep the actual trajectory of the system within a tube centered at the nominal trajectory, and the cross-section of the tube is error-invariant. Since it is desirable to have actual trajectories of the system closer to the nominal trajectory, the error-invariant set should have desirably small cross-section.

  An another control strategy, which uses RCI sets for constraint control are the interpolation-based control (IBC) strategies [18]–[22]. As the name suggests, an IBC controller interpolates between two (or more) local controllers, to guarantee constraint satisfaction and specific performance. Since RCI sets are used as a constraint within the optimization problem to compute control input, the computational complexity of the IBC controller depends on the representational complexity of the RCI set.

- **Other Applications**: RCI sets are also employed to predict constraint violation in reference governor systems for constrained tracking [23], [24]. Recently, [25] showed that RCI sets could be used to solve the measurement scheduling problem for dynamical systems. In the measurement scheduling problem, RCI sets are used to understand how long the system can evolve without measurements while satisfying state and input constraints. Furthermore, RCI can be used as a supervisory level to guarantee safety with minimal intervention on top of an existing controller [26].

Due to their properties and vast applicability, RCI sets have been an area of interest within the controls community for the past six decades [1], [6], [9], [27]. Consequently, a lot of research has been devoted to the computation of the maximal and the minimal RCI sets, a comprehensive survey on the existing approaches can be found in [9], [27]. We next briefly survey the most recent methods and classify them based on the system type they are applicable.

1.2.2 Survey of Existing Methods

Linear Systems

The algorithms to compute an RCI set for a linear system with polytopic constraints are well established in the literature [1], [9], [28], [29]. A widely used algorithm to compute an RCI set for linear systems belongs to the class of the so-called geometric approach, in which a recursive method based on the calculation of one-step backward reachable set is employed until some termination condition is matched. At each iteration, set operations like Minkowski sum, projection and finding minimal set representation are performed on polytopes, which can be computationally very demanding [1], [30], [31]. Depending upon the choice of the initial set, the result of such a recursive method is the arbitrarily close outer/inner approximation of the maximal RCI set [30], [32]. For polytopic linear systems, the overall computational complexity of the geometric approach grows exponentially with each additional system vertex and dimension [1], [33], [34]. Although effective, the geometric approach does not guarantee finite time termination. Also, the obtained set may have a very high representational complexity (i.e., large number of intersecting hyperplanes) [9], [28], [30], [32], which may be not suitable for the controller synthesis.

Hence approaches have been proposed aiming at reducing the computational complexity of the algorithms that are used to calculate the maximal/minimal RCI set and finding an approximation of these sets with a relatively less complex representation; see [11], [35], [36] and the references therein. A commonly used candidate RCI set descriptions that would facilitate computation in a finite number of steps are the ellipsoidal sets. The main advantage of using ellipsoidal sets is that only one quadratic inequality is required for the set representation, and it can be computed by solving a semi-definite program (SDP) [37]–[39].

Closely related work proposes computation of semi-ellipsoidal RCI sets [40], [41]. These sets are represented by the intersection of a polytopic and an ellipsoidal set. In [7], a new approach is proposed for the computation of the RCI sets described by some higher-order polynomial, which is applicable to LTI systems. For linear systems with polytopic constraints, it is well known from the literature that the maximal RCI set is also polytopic [1]. Therefore, using a polytopic candidate RCI set would qualify to be an obvious choice.

However, due to the quadratic nature of the constraints, an ellipsoidal or quasi-ellipsoidal RCI set would not be convenient for use in online control strategies like MPC [42], [43]. Indeed, the overall optimization problem would become a quadratically constrained quadratic problem (QCQP), which is computationally more demanding when compared with a linear program (LP) or quadratic program (QP). Thus, various recent methods shifted their focus towards the computation of polytopic RCI set represented by a number of inequalities defined beforehand.

In [44], [45], approaches are proposed to calculate a control-invariant set for linear systems of desired complexity in terms of the number of vertices. Since vertices are involved, a main drawback of the approach is the exponential growth in the computational complexity of the algorithm with the system dimension. Hence, a recent body of works deal with polytopic low-complexity RCI (LC-RCI) sets identified by their edges [42], [46], [47]. In all of these works, the considered LC-RCI set is symmetric around the origin, and the number of the associated affine inequalities is equal to twice the state dimension. In [48], [49], an improved algorithm is proposed, which allows the complexity of the RCI set to be predefined.

Nonlinear Systems

Computing RCI sets for general nonlinear systems is still an open problem, i.e., even if the RCI sets for a system exist, it may not be possible to compute them using existing approaches. Over the years, there have been some methods which calculate the RCI sets for the nonlinear systems. For example, [50] proposes an approach which computes polytopic RCI sets and applies to systems which can be approximated by so-called DC functions. Methods which can be applied to a system with sectored nonlinearity can be found in [39]. We can use the algorithm proposed in [51], [52] to compute invariant sets for polynomial systems. Because of the relaxation technique used, the approach in [52] can even handle polynomial constraints and can generate nonconvex RCI sets.

Nevertheless, these approaches are computationally demanding and unable to generate sets with predefined complexity and lack systematic handling of modelling uncertainties. Thus, a common practice is to approximate nonlinear systems with an uncertain linear system and use linear systems tools. The resulting RCI set obtained would be conservative or in the worst case could be empty due to the additional uncertainties added to system description while making linear approximations. [46].

Based on the above survey, we next summarize some of the main challenges faced when computing the RCI sets for a given system. Further, we explain how some of these challenges can be addressed using the various approach proposed in the thesis.

1.3 Challenges and Contributions

1. **Finite Time Termination Guarantees:** This is a well-known drawback of the geometric approach, especially if the linear system's eigenvalues corresponding to uncontrollable states are close to the unitary circle [28], [53]. In our work [54]–[58], we present iterative algorithms based on the solution of an optimization problem, more specifically, a semi-definite problem (SDP) to compute RCI sets. These algorithms were formulated in such a way that, every iteration, a new RCI set is computed of monotonically increasing volume until convergence. Hence, the procedure can be terminated at any iteration without needing a termination condition to be satisfied to obtain a desirably large RCI set eventually.

2. **Representational Complexity:** For the synthesis of controllers like MPC and IBC, it is desirable to have RCI sets that are polytopic and flexible representational complexity. We attack this problem in [54]–[58],where we propose a method to calculate centrally symmetric RCI sets. Specifically, the approach in [54], [56], computes RCI sets of fixed complexity, i.e., the same number of hyperplanes as twice the dimension of the state vector, and the one in [55], [57], [58] computes RCI sets with predefined number of hyperplanes.

3. **Systematic Handling of Parametric Uncertainties:** As discussed earlier, modelling uncertainties are unavoidable when constructing a model for real engineering systems and processes. These uncertainties can be partially known or completely unknown. For computation of RCI sets, we typically approximate the uncertain systems with an uncertain polytopic system, which overbounds the trajectories of the original system. Such an approximation disregards how the uncertainties enter the system dynamics and thus the obtained RCI sets would be conservative. To reduce the conservatism, we propose algorithms which are directly applicable to rationally parameter-dependent systems in [55]–[57]. Note that, we can represent a wider class of uncertain systems using rational parameter-dependent systems, and polytopic or affine system descriptions are just a special case of it [59].

   If modelling uncertainties are large then RCI set for it can be empty. Thus, in such a case, we may need to estimate these uncertainties. We can then use the approach presented in [58], to compute RCI sets for such systems.

4. **Nonlinear Systems:** From [60], it is well known that a wide variety of nonlinear systems can be approximated by LPV systems, with the nonlinearity embedded in the time-varying scheduling parameters. Thus to compute RCI sets for nonlinear systems, we can use the approach proposed in [58], which is directly applicable to linear parameter varying (LPV) systems.

1.4 Thesis Outline

The remainder of thesis is structured as follows

Chapter 2:

We will discuss the mathematical tools needed to derive the algorithms presented in the thesis. Most of the discussed results are well-know within the robust control domain.

Chapter 3:

This chapter proposes two algorithms to find a low-complexity robust control invariant (LC-RCI) set along with a state-feedback gain. These algorithms apply to rational parameter-dependent systems, which are subject to polytopic state and input constraints. The candidate RCI set is a polytope with fixed complexity of twice the system dimension and symmetric around the origin. The effectiveness of the proposed algorithm is illustrated by comparing it with many existing methods a number of examples.

Chapter 4:

The results proposed in the Chapter 3 are extended in this chapter to compute RCI sets with a arbitrary complexity, and thus not restricted to twice the system dimension. Moreover, we derive performance constraints, which can be included in the algorithm. By including the performance constraints, the algorithm will generate an RCI set which guarantees certain quadratic performance level for all the initial states within the RCI set. As before, we include examples to demonstrate the applicability of the proposed approach.

Chapter 5:

The algorithms in the Chapter 3-4 compute the RCI along with a static linear controller, which can be prohibitive. Thus, we propose an algorithm which directly computes the RCI sets without imposing any restrictive state feedback controller. The outcomes of the proposed algorithm can be thus used to construct a piecewise-affine controller based on offline computations. The algorithm relies upon novel LMI feasibility conditions for invariance and newly developed method for volume maximization.

Chapter 6:

In this chapter, we deviate from the rational parameter-dependent uncertain system and focus on LPV systems. We assume that the real-time measurements of the time-varying parameters are available; thus, we allow the RCI set description along with the invariance-inducing controller to be scheduling parameter dependent. This enables computing the RCI sets which can be possibly larger than the maximal RCI sets computed without exploiting parameter information. Numerical examples are included to illustrate the advantages of the proposed method.

Chapter 7:

We demonstrate how we can utilize the algorithms presented in Chapters 4 and 5 to construct a controller for a vehicle motion control application. For high-level autonomous vehicles, the controller has to guarantee the satisfaction of hard constraint on the deviation from the desired path. For the purpose, we propose a constrained control strategy which ensures safety by design.

Chapter 8:

We conclude the thesis with some reflection on the presented work and explain possible future directions of research.

CHAPTER 2

Preliminaries

In this chapter, we recall some known mathematical tools and results, which will support the development of the algorithms proposed in the thesis. Specifically, we will focus on the LMIs and various related result. For detailed discussion and proofs of the results, one can refer to the monographs [59], [61], [62] and references therein.

2.1 Semi-Definite Program

SDPs have vast application in control theory as well as other fields such as combinatorial and robust optimization. An SDP problem is a convex optimization problem and is formally expressed as

$$\begin{aligned} \min_{y} \quad & c^T y \\ \text{subject to:} \quad & F(y) \succ 0 \end{aligned} \tag{2.1}$$

where $c, y \in \mathbb{R}^m$, $y = [y_1, \ldots, y_m]^T$, $F(y) = F_0 + y_1 F_1 + \cdots + y_m F_m$ and $F_0, F_i \in \mathbb{R}^{n \times n}$ are symmetric matrices. The inequality $F(y) \succ 0$ involves matrix variables, and thus in the literature, such inequalities are commonly known as linear matrix inequality (LMI). There are various algorithms to solve the problem (2.1) efficiently, for instance, using interior-point methods [61], [63]. Furthermore, many freely or commercially available solvers such as SeDuMi [64], SDPT3 [65] or MOSEK [66] can be used together with the Yalmip interface [67] or the CVX interface [68] to solve SDP problems.

2.2 Linear Matrix Inequalities

A LMI is an inequality condition of the form

$$F(y) = F_0 + y_1 F_1 + \cdots + y_m F_m \succ 0, \tag{2.2}$$

where, $y = [y_1, y_2, \ldots, y_m]$ are called the decision variables and $y \in \mathbb{R}^m$. Further, $F_0, F_i \in \mathbb{R}^{n\times n}$ are symmetric matrix i.e., $F_i = F_i^T$. In general, a feasible solution to an inequality of the form (2.2), is the vector y which makes $F(y)$ positive definite or $u^T F(y) u > 0$, $\forall u \neq 0$.

Using LMIs, we can represent a variety of convex sets $\mathcal{S} = \{y | F(y) \succ 0\}$. For example, the set formed intersection of various convex constraints like linear inequalities, quadratic inequalities and matrix norm inequalities, which are commonly encountered in control theory, can be represented by LMIs [61]. The earliest use of LMI can be traced back to more than 100 years, when Lyapunov showed that for the stability of the given linear system $\dot{x} = Ax(t)$, a necessary and sufficient condition is the existence of a matrix $P \succ 0$ satisfying

$$A^T P + PA \prec 0. \tag{2.3}$$

Note that (2.3) is an LMI and we can always rewrite it in the standard form (2.2).

Example: if $A = \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix}$ and $P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}$, we can express (2.3) as

$$p_1 \underbrace{\begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix}}_{F_1} + p_2 \underbrace{\begin{bmatrix} 0 & 3 \\ 3 & -2 \end{bmatrix}}_{F_2} + p_3 \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix}}_{F_3} \succ 0, \tag{2.4}$$

Then by selecting $F_0 = 0$ and $y = [p_1, p_2, p_3]$, we can rewrite (2.4) in the form (2.2).

Moreover, if we have multiple LMI constraints $F^1(y) \succ 0, \ldots, F^q(y) \succ 0$ on an unknown variable y then they can be an equivalently expressed as single LMI constraint $\mathsf{Diag}(F^1(y), \ldots, F^q(y)) \succ 0$. Though the stability condition (2.3) is a LMI from start, in reality, we rarely arrive at LMI conditions when working on control-theoretic problems. Most of the derived conditions are nonlinear, which are practically intractable for solving. Thus, we next discuss a few known results applicable to matrix inequality conditions, using which we may recast them as LMIs.

2.2.1 Known Results in LMIs

Congruence Transform

If A is a square matrix and M is non-singular, then the product $M^T AM$ is called a congruence transformation of A. A known property of congruence transform of a symmetric matrix is that it does not change the number of positive and negative eigenvalues. Thus,

$$A \succ 0 \ \Leftrightarrow M^T AM \succ 0 \tag{2.5}$$

Schur Complement Lemma

Many nonlinear inequalities are converted to LMI form using the Schur complement. The basic idea is to re-express the matrix inequality condition in one of the equivalent ways given in the following lemma.

Lemma 1 (Schur Complement[61])**:** *Let us consider the matrices $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$ and $S \in \mathbb{R}^{n\times m}$. Then, the following statements are equivalent*

i. $\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \succ 0.$

ii. $Q \succ 0$ *and* $R - S^T Q^{-1} S \succ 0.$

iii. $R \succ 0$ *and* $Q - SR^{-1} S^T \succ 0.$

Finsler's Lemma

This lemma states equivalent ways of expressing the positive definiteness of a quadratic form subject to a linear constraint. Apart from linearization, this lemma is widely used to dilate LMI condition with additional variables (see, e.g., [69]), which can provide extra degrees of freedom when executing numerical procedures.

Lemma 2 (Finsler's Lemma[61], [70])**:** *Let the matrix $Q \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times n}$, then following statements are equivalent:*

i. $y^T Q y \succ 0,\ \forall By = 0,\ y \neq 0.$

ii. $(B^{\perp})^T Q B^{\perp} \succ 0$, *where* $BB^{\perp} = 0$.

iii. $\exists X \in \mathbb{R}^{n\times m}$ *such that* $Q + XB + B^T X^T \succ 0$.

iv. $\exists \mu \in \mathbb{R}$ *such that* $Q - \mu B^T B \succ 0$.

Often a condition $\mathrm{rank}(B) < n$ is also included, however [71] showed that this condition is not necessary.

The S-Procedure

We often encounter mathematical problems where a quadratic function $f_0(x)$ should be positive over a domain of x defined by another set of quadratic inequalities $f_i(x) \geq 0$, for $i = 1, \ldots, q$. Then using S-procedure, we can design LMI conditions for such problems. The obtained LMI condition can be a conservative, but often it is a useful approximation of the constraint. This lemma has found great application in many problem areas within control theory, and we widely use them in the thesis. The S-procedure can formally be defined as follows

Lemma 3 (S-Procedure[72])**:** *Let $f_0, \ldots, f_q$ be quadratic functions in $x \in \mathbb{R}^n$, where*

$$f_i(x) \triangleq x^T Q_i x + 2u_i^T x + v_i,\ i = 0, \ldots, q. \tag{2.6}$$

where $Q_i = Q_i^T$. We consider the following condition on $f_0, \ldots, f_q$

$$f_0(x) \geq 0 \text{ for all } x \text{ satisfying } f_i(x) \geq 0, \text{for } i = 1, \ldots, q. \tag{2.7}$$

holds if there exist $\tau_i \geq 0,\ i = 1, \ldots, q$ such that

$$f_0(x) - \sum_{i=1}^{q} \tau_i f_i(x) \geq 0,\ \forall x \tag{2.8}$$

Note (2.8) *can be equivalently expressed in a matrix inequality form as*

$$\begin{bmatrix} Q_0 & u_0 \\ u_0^T & v_0 \end{bmatrix} - \sum_{i=1}^{q} \tau_i \begin{bmatrix} Q_i & u_i \\ u_i^T & v_i \end{bmatrix} \succeq 0 \tag{2.9}$$

The converse statement in the **Lemma** 3 is not true unless $q = 1$, provided that there is some x_0 such that $f_1(x_0) > 0$.

Young's Inequality

In the thesis, we will face matrix inequality condition of the form $L^T M^{-1} L \succ 0$, where both the matrices L and M are variable matrices. Such inequality can be linearized with the help of the following lemma.

Lemma 4 ([56])**:** *Let $L \in \mathbb{R}^{m\times n}$ be any arbitrary matrix and $M \in \mathbb{R}^{m\times m}$ be a positive definite matrix then following relation always holds for any arbitrary matrix $Y \in \mathbb{R}^{m\times n}$*

$$L^T M^{-1} L \succeq L^T Y + Y^T L - Y^T M Y \tag{2.10}$$

Proof.

$$\begin{aligned} L^T M^{-1} L &= (L - MY)^T M^{-1}(L - MY) + L^T Y + Y^T L - Y^T M Y \\ &\succeq L^T Y + Y^T L - Y^T M Y. \end{aligned}$$

□

Linearization of the term on the l.h.s of (2.10) can be achieved by replacing it with its lower bound on the r.h.s. The resulting condition will be linear for a fixed Y. An ideal choice would be $Y = M^{-1}L$ since this makes the residual term $(L - MY)^T M^{-1}(L - MY) = 0$. Nevertheless, the lower bound in (2.10) would then be unchanged.

Remark 1: Though Y can be any arbitrary matrix of compatible dimension, to reduce conservatism in iterative linearization procedures (as explained in [56]), an appropriate choice would be to select $Y = M_0^{-1} L_0$, where M_0, L_0 are the values of M, L from the previous iteration. Thus the nonlinearity can be resolved by using successive linearization technique in which the residual term shrinks at each iteration until convergence (see, [56] for details). Notice that (2.10) holds with equality if $L = L_0$ and $M = M_0$. This will be a key property which will be exploited in this thesis to derive various iterative schemes.

2.3 Parameter-Dependent LMIs

Robust stability analysis or controller synthesis problems commonly result in SDP problems where the data matrices are a function of some real or complex parameter δ. The actual value of the parameter δ is not available but known to be contained in some set $\boldsymbol{\delta}$. Such a problem can be stated as

$$\begin{array}{cc} \min\limits_{y} & c^T y \\ \text{subject to:} & F(y,\delta) \succ 0,\ \forall \delta \in \boldsymbol{\delta}, \end{array} \tag{2.11}$$

where, $F(y,\delta) \triangleq F_0(\delta) + y_1 F_1(\delta) + \cdots + y_m F_m(\delta)$. The constraint $F(y,\delta) \succ 0$ is an LMI for a given value of δ, hence for each $\delta \in \boldsymbol{\delta}$, the inequality in (2.11) represents infinite LMI constraints. The optimization problem (2.11) is also referred to as a *semi-infinite* problem, which is known to be intractable.

Example: For an uncertain system $\dot{x} = A(\delta)x$, where $\delta \in \boldsymbol{\delta}$, the sufficient Lyapunov stability condition is the existence of a matrix $P \succ 0$ satisfying

$$A(\delta)^T P + PA(\delta) \prec 0,\ \forall \delta \in \boldsymbol{\delta} \tag{2.12}$$

We can express the inequality in (2.12) in the form $F(y,\delta) \succ 0$. The interested readers is referred to [59], [61], [62] for further details about LMIs involving uncertainties.

It is straightforward to see that (2.12) can be equivalently replaced by finitely many LMIs if $F(y,\delta)$ is affine in δ, and the set $\boldsymbol{\delta}$ is polytopic. This is because when the set $\boldsymbol{\delta}$ is polytopic, it can be represented by finitely many generators $\boldsymbol{\delta}^v = \{\delta^1, \ldots, \delta^r\}$, and thus the equivalent condition for $F(y,\delta) \succ 0$ is given by

$$F(y,\delta^i) \succ 0,\ \forall \delta^i \in \boldsymbol{\delta}^v. \tag{2.13}$$

Unfortunately, in the robust control design problems often result in matrix inequalities conditions which are nonlinear in the parameter δ. Furthermore, the parameter set $\boldsymbol{\delta}$ can be non-polytopic. Thus, we cannot equivalently replace the robust LMI with (2.13). In many cases, we can perform an affine approximation of these nonlinear parameter-dependent LMIs. For instance, we can over-bound the non-polytopic uncertainty sets with polytopic sets to use (2.13). Obviously, such an approximation would result in conservative conditions, and possibly the resultant feasibility set could be empty.

Depending upon the type of nonlinearity and the parameter set $\boldsymbol{\delta}$, we can use various relaxation methods in the literature. A careful choice of the relaxation methods can help obtain the least conservative sufficient (possibly necessary) LMI conditions for the given parameter-dependent matrix inequality. A non-exhaustive list of various relaxation methods can be found in [59], [62], [73]. We will next discuss some of these methods which will be used in this thesis.

2.3.1 Pólya's Relaxation Theorem

The Pólya's relaxation method is one of the important tools available to deal with the homogeneous polynomial parameter-dependent LMIs where the parameters belong to a set $\boldsymbol{\delta}$ described by unit-simplex. The relaxation gives tractable sufficient LMI conditions (independent of parameters) for the given parameter-dependent matrix inequality.

To state the result, we first need to introduce some additional notation. Let $r \in \mathbb{Z}_+$, we define $\mathcal{J}(d)$ as the lexically ordered set of r-tuples obtained as all possible combinations of $\beta = \beta_1\beta_2\cdots\beta_r$, $\beta_i \in \mathbb{Z}_+$, $i = 1, \dots, r$ such that $\beta_1 + \beta_2 + \cdots + \beta_r = d$. $\mathcal{J}_q(d)$ is the q-th element of $\mathcal{J}(d)$, $q = 1, \dots, \mathfrak{L}(d, r)$, where $\mathfrak{L}(d, r) = (r + d - 1)!/(d!(r - 1)!)$. For a given $\mathcal{J}_q(d)$, the associated standard multinomial coefficient is $\mathfrak{X}_q(d) = d!/(\beta_1!\beta_2!\cdots\beta_r!)$, where $\beta_1\beta_2\cdots\beta_r = \mathcal{J}_q(d)$.

For clarity of notations, we use an example in which $d = 3$ and $r = 3$. This gives $\mathfrak{L}(d, r) = 10$ and $\mathcal{J}(d) = \{003, 012, 021, 030, 102, 111, 120, 201, 210, 300\}$. For some $q = 2$, we have $\mathcal{J}_q(d) = 012$ and $\mathfrak{X}_q(d) = d!/(\beta_1!\beta_2!\beta_3!) = 3!/(0!1!2!) = 3$. Now we are ready to state the Pólya's Relaxation theorem.

Theorem 1 (Pólya's Relaxation [74], [75])**:** *Let $\mathcal{J}_q(g) = \beta_1 \cdots \beta_r$, and*

$$\mathcal{R}(\delta) = \sum_{q=1}^{\mathfrak{L}(g,r)} \mathfrak{X}_q(g) R_q \delta_1^{\beta_1} \delta_2^{\beta_2} \cdots \delta_r^{\beta_r}, \tag{2.14}$$

be a homogeneous matrix-valued polynomial function of degree g which is pos-

itive on the simplex,

$$\boldsymbol{\delta} = \left\{ \delta \in \mathbb{R}^r : \sum_{k=1}^{r} \delta_k = 1,\ \delta_k \geq 0 \right\}. \tag{2.15}$$

then all coefficients of the extended homogeneous matrix valued polynomial

$$\mathcal{R}_d(\delta) = \mathcal{R}(\delta). \left(\sum_{k=1}^{r} \delta_k \right)^d \tag{2.16}$$

are positive for a sufficiently large Pólya degree $d \in \mathbb{R}_+$.

Broadly speaking, the Pólya relaxation gives a sufficient LMI condition (free of parameters) for the positive definiteness of the matrix-valued polynomial $\mathcal{R}(\delta),\ \forall \delta \in \boldsymbol{\delta}$. The sufficient condition require all the matrix coefficients of the extended polynomial $\mathcal{R}_d(\delta)$ in (2.16) to be positive definite for some $d \geq 0$. With increased $d \in \{0, 1, 2, \ldots\}$ choices, we could obtain potentially less conservative conditions at the cost of a combinatorial increase in the number of LMIs. The relaxed conditions are asymptotically exact (i.e. as $d \to \infty$), provided that $\mathcal{R}(\delta) \succ 0$ is strictly feasible for all $\delta \in \boldsymbol{\delta}$; see [73]. Note that in the original form, the Pólya relaxation theorem was proposed for a scalar-valued homogeneous polynomial and can be found in [76].

We next consider a scalar-valued homogeneous polynomial to get some intuitive idea behind the relaxation.

Example: Consider a scalar valued polynomial $\mathcal{R}(\delta) = a\delta_1^2 + b\delta_1\delta_2 + a\delta_2^2$, where $\delta = [\delta_1,\ \delta_2]^T$ belong to a unit simplex set $\boldsymbol{\delta}$, i.e., $\sum_{k=1}^{2} \delta_k = 1$ and $\delta_1, \delta_2 \geq 0$. The task here is to find feasible values of the scalar variables a, b for which $\mathcal{R}(\delta) > 0,\ \forall \delta \in \boldsymbol{\delta}$.

By looking at the expression of the polynomial $\mathcal{R}(\delta)$, it is obvious that for any values of a, b in the first quadrant of the (a, b)-space, $\mathcal{R}(\delta) > 0, \forall \delta \in \boldsymbol{\delta}$, since $\delta_1, \delta_2 \geq 0$. This is the same condition obtained when we use Polya's relaxation order $d = 0$. However, it would be more interesting to see if we can find some values of a, b, which can be negative but still $\mathcal{R}(\delta) > 0$ holds for all $\boldsymbol{\delta}$. For this, we next use higher-order Polya's relaxation of order $d = 1$. The extended polynomial (2.16) is then given by $\mathcal{R}_1(\delta) = a\delta_1^3 + (a+b)\delta_1^2\delta_2 + (a+b)\delta_1\delta_2^2 + a\delta_2^2$. In this case, the sufficient conditions are $a > 0$ and $a + b > 0$, which includes all the values of a and b in the first quadrant and half of the second quadrant (where $b \leq 0$). Thus by increasing order of the relaxation, a larger and larger feasible set of solutions is obtained until convergence to a final set. This also implies, the relaxation becomes less conservative by increasing the order of the relaxation.

2.3.2 Full-Block S-Procedure

Another form of parametric nonlinearity we will see in this thesis is when $F(y,\delta)$ in (2.11) has a rational dependence on the uncertain/varying parameter δ. We handle the rational parameter-dependent LMIs using Full-Block S-procedure proposed originally in [77].

For understanding the full-block S-procedure, we first introduce a function Δ, which is linearly dependent on the parameter δ and an operator '$\star$' to define an LFT

$$\Delta \star \underbrace{\left[\begin{array}{c|c} \mathcal{Y}_{11} & \mathcal{Y}_{12} \\ \hline \mathcal{Y}_{21} & \mathcal{Y}_{22} \end{array}\right]}_{\mathcal{Y}} \triangleq \mathcal{Y}_{22} + \mathcal{Y}_{21}\Delta(I - \mathcal{Y}_{11}\Delta)^{-1}\mathcal{Y}_{12}. \tag{2.17}$$

The LFT (2.17) is said to be well-posed when $I - \mathcal{Y}_{11}\Delta$ is non-singular $\forall \Delta \in \mathbf{\Delta}$. If $F(y,\delta)$ is rationally dependent on the parameter δ and is symmetric then it can be re-written as

$$F(y,\delta) = \Delta \star \mathcal{Y} + (\Delta \star \mathcal{Y})^T \succ 0 \tag{2.18}$$

We next present a lemma, which gives sufficient conditions (possibly necessary as well, if the set $\mathbf{\Delta}$ is compact) for (2.18). The lemma was derived in [56] and is a corollary of the more general full-block S-procedure in [77].

Lemma 5: *The LFT $\Delta \star \mathcal{Y}$ is well-posed and*

$$\mathsf{He}\{\Delta \star \mathcal{Y}\} \triangleq \Delta \star \mathcal{Y} + (\Delta \star \mathcal{Y})^T \succ 0,\ \forall \Delta \in \mathbf{\Delta} \tag{2.19}$$

holds if there exists a matrix M that satisfies

$$\begin{bmatrix} \Delta^T \\ I \end{bmatrix}^T \underbrace{\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}}_{M} \begin{bmatrix} \Delta^T \\ I \end{bmatrix} \preccurlyeq 0,\ \ \forall \Delta \in \mathbf{\Delta}, \tag{2.20}$$

and

$$\begin{bmatrix} \mathcal{Y}_{21} R \mathcal{Y}_{21}^T + \mathsf{He}\{\mathcal{Y}_{22}\} & \mathcal{Y}_{21} R \mathcal{Y}_{11}^T + \mathcal{Y}_{21} S^T + \mathcal{Y}_{12}^T \\ * & Q + \mathcal{Y}_{11} R \mathcal{Y}_{11}^T + \mathsf{He}\{\mathcal{Y}_{11} S^T\} \end{bmatrix} \succ 0. \tag{2.21}$$

Proof. We only provide sufficiency proof as this is enough for our purposes in the thesis. By multiplying (2.21) from the left with $\Psi = \begin{bmatrix} I & \mathcal{Y}_{21}\Delta\left(I - \mathcal{Y}_{11}\Delta\right)^{-1} \end{bmatrix}$ and from the right with Ψ^T, we obtain

$$\mathcal{Y}_{21}\left(I - \Delta\mathcal{Y}_{11}\right)^{-1} \underbrace{\begin{bmatrix}\Delta & I\end{bmatrix} M \begin{bmatrix}\Delta & I\end{bmatrix}^T}_{\preccurlyeq 0} \left(I - \Delta\mathcal{Y}_{11}\right)^{-T} \mathcal{Y}_{21}^T + \mathsf{He}\{\Delta \star \mathcal{Y}\} \succ 0. \tag{2.22}$$

This clearly implies (2.19) since M is required to satisfy (2.20). To establish well-posedness, we assume that there exists a nonzero vector ρ and a specific $\Delta_\circ \in \boldsymbol{\Delta}$ for which $\rho^T(I - \mathcal{Y}_{11}\Delta_\circ) = 0$, i.e. $I - \mathcal{Y}_{11}\Delta_\circ$ is singular, and show that this leads to a contradiction. With $\rho^T = \rho^T\mathcal{Y}_{11}\Delta_\circ$, let us now multiply (2.20) the from the left with $\rho^T\mathcal{Y}_{11}$ and from the right with $\mathcal{Y}_{11}^T\rho$ to infer

$$\begin{aligned}\rho^T\mathcal{Y}_{11}\begin{bmatrix}\Delta_\circ & I\end{bmatrix} M \begin{bmatrix}\Delta_\circ & I\end{bmatrix}^T \mathcal{Y}_{11}^T\rho &= \rho^T\left(Q + \mathcal{Y}_{11}R\mathcal{Y}_{11}^T + \mathsf{He}\{\mathcal{Y}_{11}S^T\}\right)\rho, \\ &\leq 0.\end{aligned}$$

This contradicts the fact that the $(2,2)$ block of the matrix in (2.21) is necessarily positive-definite. □

For a compact set $\boldsymbol{\Delta}$, we can even prove necessity by adapting the proof given in [77] for the more general case. Lemma 5 allows us to replace the matrix inequality (2.19) that has rational parameter dependence with conditions consisting of two LMIs which are easier to deal with. Indeed (2.21) is a single (i.e. parameter-independent) matrix inequality, while (2.20) has only quadratic parameter dependence. Nevertheless, (2.20) has to be satisfied for all possible values of $\Delta \in \boldsymbol{\Delta}$, which results in infinitely many LMI conditions. Hence, in order to obtain tractable sufficient conditions for (2.19), we need to employ some relaxation methods to derive finitely many LMI conditions that ensure (2.20). Various relaxation methods in the literature serve this purpose depending upon the type of the uncertainty region. For polytopic regions, one can consider the so-called convex-hull relaxation or Pólya's method, while for regions described by polynomial inequalities one might use the sum-of-squares (SOS) approach (see e.g. [59], [73], [78] and the references therein).

In the thesis, we will limit our interest in an uncertain/varying parameter, which belongs to a polytopic regions defined as

$$\mathbf{\Delta} = \mathsf{conv}(\mathbf{\Delta}_c) \triangleq \left\{ \sum_{j=1}^{\kappa} \alpha_j \Delta^j : \sum_{j=1}^{\kappa} \alpha_j = 1, \alpha_j \geq 0 \right\}. \tag{2.23}$$

Thus, next we present relaxation of the condition (2.20) using Polya's relaxation theorem, since the convex-hull relaxation are known to be relatively conservative.

Relaxation based on Pólya's Method

These conditions are adapted from [78]. Let the set of all legitimate multipliers represented by $\boldsymbol{\mathcal{M}}$ be

$$\boldsymbol{\mathcal{M}} \triangleq \left\{ M : (2.20) \right\}. \tag{2.24}$$

Our goal is to characterize an inner approximation $\boldsymbol{\mathcal{M}}_{pol} \subseteq \boldsymbol{\mathcal{M}}$ based on finitely many LMIs. For the sake of concise expressions, we first introduce

$$\Omega_{ij}(M) \triangleq \begin{bmatrix} (\Delta^i)^T \\ I \end{bmatrix}^T M \begin{bmatrix} (\Delta^j)^T \\ I \end{bmatrix}, i, j = 1, \cdots, \kappa. \tag{2.25}$$

As explained in **Theorem** 1, the key idea behind the Pólya approach is to express the left-hand side of (2.20) as a polynomial matrix that is homogeneous in α. In the simplest case, we can achieve this with a polynomial order of two:

$$\sum_{j=1}^{\kappa} \alpha_j^2 \Omega_{jj} + \sum_{j=1}^{\kappa} \sum_{i=j+1}^{\kappa} \alpha_j \alpha_i \left(\Omega_{ji} + \Omega_{ji}^T \right) \preccurlyeq 0. \tag{2.26}$$

With $\alpha_j \in [0, 1]$, we can now state a set of sufficient LMI conditions for (2.20) by using zeroth order Polya's relaxation as follows:

$$\left. \begin{array}{rcll} \Omega_{jj}(M) & \preccurlyeq & 0, & j = 1, ..., \kappa \\ \mathsf{He}\{\Omega_{ji}(M)\} & \preccurlyeq & 0, & j = 1, ..., \kappa;\ i = j+1, ..., \kappa \end{array} \right\} \tag{2.27}$$

We have thus obtained an inner approximation to the original set of multipliers $\boldsymbol{\mathcal{M}}$ as follows:

$$\boldsymbol{\mathcal{M}}_{pol} \triangleq \{M : (2.27)\} \subseteq \boldsymbol{\mathcal{M}}. \tag{2.28}$$

As explained in in **Theorem** 1, it is possible to obtain potentially less conservative conditions by using higher order Polya's relaxation ie., by first multiplying (2.26) with $(\sum_{j=1}^{\kappa} \alpha_j)^d = 1$ and then enforcing that all coefficient matrices are non-negative definite. Note that with increased $d \in \{0, 1, 2, \dots\}$ choices, we could potentially reduce conservatism at the cost of a combinatorial increase in the number of LMIs.

Next, we discuss an important but naive relaxation technique known as gridding. Such a relaxation technique can be useful if parameter-dependent LMIs are not necessarily required to be satisfied for all parameter values.

2.3.3 Relaxation of Parameter-Dependent LMIs via Gridding

Gridding approach is the simplest and the most intuitive relaxation approach. The approach simply proposes to grid the uncertainty region $\boldsymbol{\delta}$ in (2.11) into a sufficiently dense set of points $\boldsymbol{\delta}_g$ and verify the positivity of the LMI in (2.11) for all $\delta \in \boldsymbol{\delta}_g$. The main advantage of the approach is that for the grid points if there is no feasible solution found, then we can conclude that there is no feasible solution for the original problem. Since the feasibility of the LMI is only verified for the finite set of points $\boldsymbol{\delta}_g$, the feasibility is then only a necessary condition for the feasibility of the original problem and not sufficient. The computational complexity of the gridding problem can be very high and suffer from the curse of dimensionality. For instance, if the number of parameters is N_p, and we take N number of samples for each parameter, then the number of LMIs to consider is equal to N^{N_p}.

2.3.4 Finsler's Lemma (parameter-dependent)

Lastly, we would also like state the parameter-dependent version of the Finsler's Lemma, which in its original version, was already discussed in **Lemma** 2.

Lemma 6 (Finsler's Lemma[71])**:** *Let $\delta \in \boldsymbol{\delta} \subseteq \mathbb{R}^{N_\delta}$, $Q : \boldsymbol{\delta} \to \mathbb{R}^{n\times n}$ and $B : \boldsymbol{\delta} \to \mathbb{R}^{m\times n}$. Then the following statements are equivalent:*

i. For each $\delta \in \boldsymbol{\delta}$, $y^T Q(\delta) y \succ 0$, $\forall B(\delta)y = 0$, $y \neq 0$.

ii. $(B(\delta)^\perp)^T Q B(\delta)^\perp \succ 0$, where $B(\delta)B(\delta)^\perp = 0$.

iii. For each $\delta \in \boldsymbol{\delta}$, $\exists X(\delta) \in \mathbb{R}^{n \times m}$ *such that*
$$Q(\delta) + X(\delta)B(\delta) + B(\delta)^T X(\delta)^T \succ 0.$$

iv. For each $\delta \in \boldsymbol{\delta}$, $\exists \mu(\delta) \in \mathbb{R}$ *such that* $Q(\delta) - \mu(\delta)B(\delta)^T B(\delta) \succ 0.$

Note: In this chapter, though we present all the result with just strict inequality, most of the results can be applied to non-strict LMIs.

Part II

Invariant set Computation

CHAPTER 3

Low-Complexity Control-Invariant Sets for Uncertain Systems

3.1 Introduction

In this chapter, we will present a set of algorithms to compute low-complexity RCI (LC-RCI) sets along with associated static linear state-feedback laws for uncertain systems. The RCI set is assumed to be symmetric around the origin and described by the same number of affine inequalities as twice the dimension of the state vector. These algorithms will form the base for some of the algorithms proposed later in the thesis. The results presented in this chapter have been presented in [54], [56].

3.2 Brief Literature Survey and Comparison

A recent body of works deal with polytopic LC-RCI sets identified by their edges [42], [46], [47]. In all of these works, the considered LC-RCI set is symmetric around the origin, and the number of the associated affine inequalities is equal to twice the state dimension. In [42], [46], an iterative approach is proposed to find an LC-RCI set for an uncertain parameter-dependent system, with the parameter vector assumed to lie in a known polytopic region. An SDP is solved to find an RCI set with a desirably large volume together with an associated state-feedback gain. Due to the polytopic uncertainty description, the number of LMIs for the invariance condition could grow exponentially. In [47], a system with norm-bounded uncertainty is considered. The parameter-dependent conditions resulting from the state and input constraints, and invariance condition were relaxed using the D-G scales in the way commonly employed within the robust control literature. The block-diagonal, norm-bounded uncertainty characterization is somewhat restrictive as it disregards any possible correlation between different parameters. Moreover, relaxation with block-diagonal D-G scales is potentially conservative for multiple blocks if compared to full-block scaling matrices. Another limitation of [47] is that the derivations are restricted to systems with affine parameter dependence.

In this chapter, we consider the LC-RCI set computation problem for a broader class of uncertain systems, compared to [42], [46], [47]. The supposed system depends rationally on the uncertain parameters which are allowed to vary arbitrarily within a known polytopic region. Our derivations are based on a state transformation which facilitates an equivalent expression of state and control input constraints as simple scalar inequalities. On the other hand, the invariance condition is relaxed into two alternative sets of (standard and dilated) matrix inequality conditions with rational parameter dependence. Sufficient conditions are then derived in the form of finitely many LMIs by an application of the full-block S-procedure **Lemma** 5. Tractable formulations of the system constraints and invariance condition are then used to develop one-step as well as iterative algorithms for the computation of LC-RCI sets of desirably large/small volumes. The proposed algorithms rely on novel methods for maximization (or minimization) of the determinant of a non-symmetric matrix under LMI conditions.

In the next section, we will summarize the exact problem we solve.

3.3 Problem Formulation

3.3.1 System and Constraints

Consider a discrete-time uncertain system whose dynamics is described by

$$x(k+1) = \mathcal{A}(\Delta(k))x(k)+\mathcal{B}(\Delta(k))u(k)+\mathcal{E}(\Delta(k))w(k), \tag{3.1}$$

where x, u and w the are the state, control input and disturbance vectors respectively. The system matrices are assumed to depend rationally on an uncertain (possibly) time-varying parameter matrix Δ that satisfies

$$\Delta(k) \in \mathbf{\Delta}, \forall k \geq 0, \tag{3.2}$$

where $\mathbf{\Delta}$ is a known compact uncertainty set. We restrict our interest to a polytopic uncertainty region that is expressed as the convex-hull of a given set of finitely many matrices $\mathbf{\Delta}_c = \left\{\Delta^1, \ldots, \Delta^\kappa\right\}$:

$$\mathbf{\Delta} = \mathsf{conv}(\mathbf{\Delta}_c) \triangleq \left\{\sum_{j=1}^{\kappa} \alpha_j \Delta^j : \sum_{j=1}^{\kappa} \alpha_j = 1, \alpha_j \geq 0\right\}. \tag{3.3}$$

The dependence of the system matrices on the parameter matrix is expressed compactly with an LFT of the form

$$\left[\, \mathcal{A}(\Delta) \,\middle|\, \mathcal{B}(\Delta) \,\middle|\, \mathcal{E}(\Delta) \,\right] = \left[\, A \,\middle|\, B \,\middle|\, E \,\right] + B_p\Delta(I - D_p\Delta)^{-1} \left[\, A_d \,\middle|\, B_d \,\middle|\, E_d \,\right], \tag{3.4}$$

where A, B, E, A_d, B_d, E_d, B_p and D_p are known system matrices of appropriate dimensions. We assume $(I - D_p\Delta)^{-1}$ is well-posed (i.e. $I - D_p\Delta$ must be invertible for all $\Delta \in \mathbf{\Delta}$) for a meaningful problem formulation. Note that affine parameter dependence, as considered in e.g. [47], is just a special case that corresponds to $D_p = 0$ in (3.4). There are systematic methods for obtaining an LFT as in (3.4) (see e.g. [59]; Chapter 6), which usually lead to structured (block-diagonal) Δ matrices. We can also use some freely available tools like [79] to compute the LFT representation with diagonal Δ matrices. Nevertheless, in our formulation, we do not require matrix Δ to be in a particular form since we employ the full-block S-procedure **Lemma** 5.

The system (3.4) is subjected to a number of state as well as control input constraints and the disturbances are assumed to be bounded. These are expressed concisely as

$$\begin{array}{rcl} \boldsymbol{\mathcal{X}} & = & \{x \in \mathbb{R}^n : Hx \leq \mathbf{1}\}, \\ \boldsymbol{\mathcal{U}} & = & \{u \in \mathbb{R}^m : Gu \leq \mathbf{1}\}, \\ \boldsymbol{\mathcal{W}} & = & \left\{w \in \mathbb{R}^d : |Dw| \leq \mathbf{1}\right\}, \end{array} \tag{3.5}$$

where $H \in \mathbb{R}^{n_x \times n}$, $G \in \mathbb{R}^{n_u \times m}$, $D \in \mathbb{R}^{n_w \times d}$ are given matrices and $\mathbf{1}$ represents the vector of ones. The inequalities in (3.5) are to be interpreted as element-wise.

3.3.2 Candidate RCI set and Invariance Inducing Controller

The synthesis goal in this chapter is to ensure invariance in an LC-RCI set via a static state feedback controller as

$$u(k) = Kx(k), \tag{3.6}$$

where $K \in \mathbb{R}^{m \times n}$ is a gain matrix to be found. The resulting closed-loop dynamics can be expressed as

$$x^+ = (\mathcal{A}(\Delta) + \mathcal{B}(\Delta)K)\, x + \mathcal{E}(\Delta)w, \tag{3.7}$$

where the k dependence is dropped and $x(k+1)$ is represented by x^+ for simplicity. Along the lines of the recent works [42], [46], [47], [54], we assume the LC-RCI set to be symmetric with respect to origin and to be described as

$$\boldsymbol{\mathcal{C}} = \left\{x \in \mathbb{R}^n : -\mathbf{1} \leq W^{-1}x \leq \mathbf{1}\right\}, \tag{3.8}$$

where $W \in \mathbb{R}^{n \times n}$ is a square matrix to be found. For now, we assume W is invertible, which would in fact be guaranteed by the LMI conditions for invariance. Note that, once the set is constrained to be symmetric with respect to the origin, the normalization of the lower/upper bounds to ∓ 1 in (3.8) does not lead to a loss of generality.

Since the state and control input constraints are to be satisfied by the elements of $\mathcal{C}$, this implies $\mathcal{C} \subseteq \mathcal{X}$ and $K\mathcal{C} \subseteq \mathcal{U}$, which can be expressed as

$$x \in \mathcal{C} \;\Rightarrow\; x \in \mathcal{X}, \tag{3.9}$$

$$x \in \mathcal{C} \;\Rightarrow\; u = Kx \in \mathcal{U}. \tag{3.10}$$

Let us now recall the definition of robust invariance and express the corresponding condition in a similar way. A subset $\mathcal{C}$ of the state-space is referred to as *robustly invariant* for the system (3.7), if

$$x \in \mathcal{C} \;\Rightarrow\; x^+ \in \mathcal{C},\; \forall w \in \mathcal{W},\; \forall \Delta \in \boldsymbol{\Delta}. \tag{3.11}$$

The problem of finding an RCI set together with a static state-feedback gain can now be formulated as follows:

Problem 1: Given the discrete-time system in (3.1) subject to the constraints (3.5) for the polytopic uncertainty set described by (3.3), find (W, K) such that:

1. all elements of the set $\mathcal{C}$ in (3.3.2) satisfy (3.9) and (3.10);

2. the controlled system in (3.7) satisfies (3.11).

We opted for a formulation in the form of a feasibility problem solely for the convenience of our presentation. In fact, our ultimate goal is to develop iterative algorithms for maximizing the volume of the computed LC-RCI set.

3.4 Tractable Formulations of System Constraints and Invariance Conditions

In this section, we derive tractable feasibility conditions for the solvability of **Problem 1**. We first derive scalar inequality conditions for the system constraints (3.9) and (3.10). This is followed by the derivation of two alternative sets of matrix inequality conditions for invariance (3.11).

3.4.1 System Constraints

To derive conditions for the system constraints of (3.9) and (3.10), we first introduce a state transformation as

$$\theta = W^{-1}x \Leftrightarrow x = W\theta. \tag{3.12}$$

According to (3.12) the LC-RCI set in (3.8) can be expressed as

$$\boldsymbol{\mathcal{C}} = \{W\theta \in \mathbb{R}^n : \theta \in \boldsymbol{\Theta}\}, \tag{3.13}$$

where $\boldsymbol{\Theta}$ is a hyper-rectangular region defined as follows:

$$\boldsymbol{\Theta} \triangleq \{\theta \in \mathbb{R}^n : -\mathbf{1} \leq \theta \leq \mathbf{1}\}. \tag{3.14}$$

The set $\boldsymbol{\Theta}$ can also be expressed as the convex hull of the vertices, θ^j, of a hyper-rectangular region:

$$\boldsymbol{\Theta} = \mathsf{conv}\left(\left\{\theta^1, \ldots, \theta^{2^n}\right\}\right). \tag{3.15}$$

Since the state and input constraints in (3.5) are affine in $x = W\theta$, their satisfaction at θ^j imply that they are satisfied over the whole set $\boldsymbol{\Theta}$. Due to the symmetry of $\boldsymbol{\Theta}$, we can always order θ^j to have

$$\theta^{j+2^{n-1}} = -\theta^j, \; j = 1, \ldots, 2^{n-1}. \tag{3.16}$$

This allows us to express (3.9) in terms of W as follows:

$$HW\theta \leq \mathbf{1}, \forall\theta \in \boldsymbol{\Theta} \Leftrightarrow -\mathbf{1} \leq HW\theta^j \leq \mathbf{1}, \; j = 1, \ldots, 2^{n-1}. \tag{3.17}$$

In order to express the control input constraints in a similar way, we first introduce a new matrix variable as

$$N \triangleq KW \quad \Leftrightarrow \quad K = NW^{-1}. \tag{3.18}$$

The control input constraints in (3.10) then read as

$$GN\theta \leq \mathbf{1}, \forall\theta \in \boldsymbol{\Theta} \; \Leftrightarrow -\mathbf{1} \leq GN\theta^j \leq \mathbf{1}, \; j = 1, \ldots, 2^{n-1}. \tag{3.19}$$

We stress that both (3.17) and (3.19) read as (respectively $n_x \times 2^n$ and $n_u \times 2^n$) scalar inequality conditions that are equivalent to (3.9) and (3.10), respectively.

3.4.2 Standard LMI Conditions for Invariance

In this section, we derive *standard LMI conditions* for invariance. We first use (3.12) and (3.18) to express the closed-loop system dynamics in (3.7) as follows:

$$x^+ = \underbrace{\left(\mathcal{A}(\Delta)W + \mathcal{B}(\Delta)N\right)}_{\mathcal{F}(\Delta,W,N)}\theta + \mathcal{E}(\Delta)w. \tag{3.20}$$

Note that the term $\mathcal{F}(\Delta, W, N)$ is introduced in (3.20). In the sequel, to simplify the notation, we suppress the arguments in $\mathcal{F}$ and $\mathcal{E}$.

Let us express the system dynamics in the *θ-state-space* as follows:

$$W\theta^+ = \mathcal{F}\theta + \mathcal{E}w. \tag{3.21}$$

From (3.11) and (3.13), each element $\theta_i^+, i = 1, \ldots, n$ of the vector θ^+ must satisfy the following to ensure invariance:

$$1 - (\theta_i^+)^2 = 1 - (e_i^T\theta^+)^2 \geq 0, \forall\theta \in \boldsymbol{\Theta},\ i = 1, \ldots, n. \tag{3.22}$$

Here, e_i represents the i'th column of the identity matrix of size $n \times n$. A sufficient condition for (3.22) can be obtained by first multiplying this inequality with a positive scalar variable ϕ_i and then imposing this product to be greater than or equal to an expression that is known to be non-negative within the region $\boldsymbol{\Theta}$ and for a disturbance input that satisfies $w \in \boldsymbol{\mathcal{W}}$. In this fashion, we obtain a sufficient condition for (3.22) as follows:

$$\begin{aligned}\phi_i\left(1 - (e_i^T\theta^+)^2\right) \geq 2(\theta^+)^T \underbrace{\left(\mathcal{F}\theta + \mathcal{E}w - W\theta^+\right)}_{0} + \underbrace{(\mathbf{1} + \theta)^T\Lambda_i(\mathbf{1} - \theta)}_{\geq 0} \\ + \underbrace{(\mathbf{1} + Dw)^T\Gamma_i(\mathbf{1} - Dw)}_{\geq 0},\end{aligned} \tag{3.23}$$

where,

$$\phi_i \in \mathbb{R}_+,\ \Lambda_i = \mathsf{diag}(\lambda_{i1}, .., \lambda_{in}) \succcurlyeq 0, \Gamma_i = \mathsf{diag}(\gamma_{i1}, .., \gamma_{in}) \succcurlyeq 0. \tag{3.24}$$

It is straightforward to verify, based on (3.5) and (3.14), that the last two terms in the right hand side of the inequality (3.23) are both non-negative. On the other hand, the first term in the right hand side of (3.23) is identically zero because of (3.21). Via standard manipulations, we compactly express (3.23) as

$$\underbrace{\begin{bmatrix}\theta\\ w\\ -\theta^+\end{bmatrix}^T}_{\varkappa^T}\begin{bmatrix}\Lambda_i & 0 & *\\ 0 & D^T\Gamma_i D & *\\ \mathcal{F} & \mathcal{E} & W+W^T-\phi_i e_i e_i^T\end{bmatrix}\underbrace{\begin{bmatrix}\theta\\ w\\ -\theta^+\end{bmatrix}}_{\varkappa}+\phi_i \geq \mathbf{1}^T\Lambda_i\mathbf{1}+\mathbf{1}^T\Gamma_i\mathbf{1}, \tag{3.25}$$

where $*$'s represent entries that can be identified from symmetry. In order to ensure invariance, (3.25) needs to be satisfied for any $\varkappa$. With $\varkappa = 0$, we must then have

$$\phi_i \geq \mathbf{1}^T\Lambda_i\mathbf{1} + \mathbf{1}^T\Gamma_i\mathbf{1},\ i = 1, ..., n. \tag{3.26}$$

To ensure (3.25) for any $\varkappa$, it is then necessary and sufficient to impose (3.26) together with

$$\begin{bmatrix}\Lambda_i & 0 & *\\ 0 & D^T\Gamma_i D & *\\ \mathcal{F}(\Delta) & \mathcal{E}(\Delta) & W+W^T-\phi_i e_i e_i^T\end{bmatrix} \succcurlyeq 0, \forall\Delta\in\mathbf{\Delta}, \tag{3.27}$$

where parameter dependence of $\mathcal{F}$ and $\mathcal{E}$ is recalled for emphasis.

We have thus arrived at a parameter-dependent LMI condition as in (3.27), for which tractable (i.e. finitely many and sufficient) conditions need to be derived. To this end, we employ **Lemma** 5 in the Chapter 2 to arrive at a sufficient condition for (3.27). As the key ingredient in this procedure, we first recall (2.20), in which the *scaling* (or *multiplier*) matrix M is introduced and is required to satisfy

$$\begin{bmatrix}\Delta^T\\ I\end{bmatrix}^T\underbrace{\begin{bmatrix}Q & S\\ S^T & R\end{bmatrix}}_{M}\begin{bmatrix}\Delta^T\\ I\end{bmatrix} \preccurlyeq 0,\ \forall\Delta\in\mathbf{\Delta}. \tag{3.28}$$

The set of all valid scaling matrices is then identified as

$$\boldsymbol{\mathcal{M}} \triangleq \left\{M : (3.28)\right\}. \tag{3.29}$$

It is possible to obtain inner approximations of this set as $\boldsymbol{\mathcal{M}}_{pol} \subseteq \boldsymbol{\mathcal{M}}$ based on finitely many LMI conditions, presented in Chapter 2 (in Section 2.3.2) as (2.27) for the polytopic uncertainty region of (2.23). An application of the full-block S-procedure to (3.27) then leads us to an LMI-based solution to Problem 1 as follows:

Theorem 2: *Problem 1 is feasible, if there exist $W \in \mathbb{R}^{n\times n}$, $N \in \mathbb{R}^{m\times n}$ and $\phi_i \in \mathbb{R}_+$, $\Lambda_i \in \mathbb{R}^{n\times n}$, $\Gamma_i \in \mathbb{R}^{n_w\times n_w}$ as in (3.24), $M_i \in \boldsymbol{\mathcal{M}}_{pol}$, for $i = 1, ..., n$ satisfying (3.17), (3.19), (3.26) and (3.30). A state-feedback gain that solves the problem can be constructed as in (3.18).*

$$\begin{bmatrix} \Lambda_i & 0 & * & * \\ 0 & D^T\Gamma_i D & * & * \\ AW{+}BN & E & W{+}W^T{-}\phi_i e_i e_i^T + B_p R_i B_p^T & * \\ A_d W{+}B_d N & E_d & S_i B_p^T + D_p R_i B_p^T & \Phi_i \end{bmatrix} \succ 0. \tag{3.30}$$

where, $\Phi_i = Q_i + D_p S_i^T + S_i D_p^T + D_p R_i D_p^T$ and $i = 1, ..., n$

Proof. We first express the strict inequality version of (3.27) as in (2.19) with $\mathcal{Y}$ and its sub-blocks chosen as

$$\mathcal{Y} = \left[\begin{array}{c|ccc} D_p & A_d W{+}B_d N & E_d & 0 \\ \hline 0 & \frac{1}{2}\Lambda_i & 0 & 0 \\ 0 & 0 & \frac{1}{2}D^T\Gamma_i D & 0 \\ B_p & AW{+}BN & E & W - \frac{1}{2}\phi_i e_i e_i^T \end{array}\right]. \tag{3.31}$$

By now applying the full-block S-procedure as in Lemma 5, with a scaling matrix M_i associated with the i'th-invariance condition, we get the LMI condition as in (3.30). Thus a (W, N) couple satisfying (3.17), (3.19), (3.26) and (3.30) together with the matrices $M_i \in \boldsymbol{\mathcal{M}}_{pol}$ serve as a solution to Problem 1. It follows from the strict version of (3.27) that $W + W^T \succ 0$. This implies that W is invertible, since we would otherwise have $\varphi^T(W + W^T)\varphi = 0$ for a vector φ from the null-space of W. □

At this point, it is worth commenting on the conditions for state and input constraints satisfaction (3.17) and (3.19), respectively, and the invariance condition (3.26), (3.30) in reference to the recent works [47], [80]. Starting from the system constraints (3.5), we obtained equivalent conditions (3.17) and (3.19) as simple scalar inequalities in terms of N and W. On the other hand, [47], [80] provide alternative conditions that are expressed in terms of additional variables introduced via an application of the S-procedure. In our LMI conditions for invariance, the parameter-dependent LMIs (3.27) are relaxed by using full-block scaling matrices (by favor of Lemma 5 and Pólya's method) as opposed to D-G scales used in [47]. The D-G scales are constructed as block-diagonal multipliers of commuting structure for block-diagonal Δ's with norm-bounded sub-blocks. We can note two drawbacks in connection with the use of D-G scales: (i) restriction to hyper-rectangular uncertainty regions, which disregards possible correlations among the parameters; (ii) restricting the choices of P_i's in Theorem 2 to satisfy $R_i = -Q_i \preccurlyeq 0$ and $S_i = -S_i^T$ in addition to the block-diagonal structure. In conclusion our result not only offers potential reduction in conservatism but also facilitates the consideration of possible correlation among the uncertain parameters. Our approach also allows using higher order Pólya relaxations as discussed in Section 2.3.2 for possible further reduction of conservatism. However, with two consecutive applications of the S-procedure, it does not seem possible to make a generic exactness statement about **Theorem** 2.

3.4.3 Dilated Conditions for Invariance

In this section, we present alternative conditions for invariance in the form of *dilated matrix inequalities*. As the first step, we associate the i'th invariance condition with a new matrix variable $V_i \in \mathbb{R}^{n\times n}$ and new vector $\xi_i \triangleq V_i^{-1}x^+$ for $i = 1, ..., n$. With $x^+ = V_i\xi_i$, it then follows from the system dynamics (3.20) that

$$\mathcal{F}\theta + \mathcal{E}w - V_i\xi_i = 0. \tag{3.32}$$

We can then express (3.23) equivalently as

$$\phi_i \left(1 - (e_i^T W^{-1} V_i \xi_i)^2\right) \geq 2\xi_i^T \underbrace{(\mathcal{F}\theta + \mathcal{E}w - V_i\xi_i)}_{0} + \underbrace{(\mathbf{1}+\theta)^T \Lambda_i (\mathbf{1}-\theta)}_{\geq 0} + \underbrace{(\mathbf{1}+Dw)^T \Gamma_i (\mathbf{1}-Dw)}_{\geq 0}. \tag{3.33}$$

Along similar lines to the previous subsection, we can obtain the i'th matrix inequality condition for invariance that accompanies (3.26) as follows:

$$\begin{bmatrix} \Lambda_i & 0 & * \\ 0 & D^T\Gamma_i D & * \\ \mathcal{F} & \mathcal{E} & V_i + V_i^T - \phi_i V_i^T W^{-T} e_i e_i^T W^{-1} V_i \end{bmatrix} \succcurlyeq 0. \tag{3.34}$$

Note that, if we select $V_i = W$, $\forall i = 1, \cdots, n$ in (3.34), we get (3.27). This shows that (3.34) is potentially less conservative than (3.27), meaning that when (3.27) is feasible, (3.34) is necessarily feasible as well. This motivates us to state an alternative solution to Problem 1 as follows:

Theorem 3: *Problem 1 is feasible, if there exist $W \in \mathbb{R}^{n\times n}$, $N \in \mathbb{R}^{m\times n}$ and $\phi_i \in \mathbb{R}_+$, $\Lambda_i \in \mathbb{R}^{n\times n}$, $\Gamma_i \in \mathbb{R}^{n_w\times n_w}$ as in (3.24), $X_i = X_i^T \in \mathbb{R}^{n\times n}$, $V_i \in \mathbb{R}^{n\times n}$, $P_i \in \boldsymbol{\mathcal{P}}_{pol}$, for $i = 1, ..., n$ satisfying (3.17), (3.19), (3.26), (3.35) and (3.36). A state-feedback gain that solves the problem can be constructed as in (3.18).*

$$\begin{bmatrix} W^T X_i^{-1} W & \phi_i e_i \\ * & \phi_i \end{bmatrix} \succ 0, \ i = 1, ..., n, \tag{3.35}$$

$$\begin{bmatrix} \Lambda_i & 0 & * & * & * \\ 0 & D^T\Gamma_i D & * & * & * \\ AW + BN & E & V_i + V_i^T + B_P R_i B_p^T & * & * \\ 0 & 0 & V_i & X_i & * \\ A_d W + B_d N & E_d & S_i B_p^T + D_p R_i B_p^T & 0 & \Phi_i \end{bmatrix} \succ 0, \tag{3.36}$$

where, $\Phi_i = Q_i + D_p S_i^T + S_i D_p^T + D_p R_i D_p^T$ and $i = 1, ..., n$.

Proof. With the intention to resolve the nonlinear variable dependence in the $(3,3)$ block of (3.34), we introduce a positive-definite matrix variable X_i that satisfies

$$X_i^{-1} - \phi_i W^{-T} e_i e_i^T W^{-1} \succ 0. \tag{3.37}$$

Then by applying the Schur complement to this inequality followed by a congruence transformation, we obtain the equivalent condition (3.35). With X_i satisfying (3.37), a condition that implies (3.34) can be formulated as

$$\begin{bmatrix} \Lambda_i & 0 & * \\ 0 & D^T\Gamma_i D & * \\ \mathcal{F} & \mathcal{E} & V_i + V_i^T - V_i^T X_i^{-1} V_i \end{bmatrix} \succ 0. \tag{3.38}$$

Again, by a standard application of the Schur complement, (3.38) can be expressed equivalently as

$$\begin{bmatrix} \Lambda_i & 0 & * & * \\ 0 & D^T\Gamma_i D & * & * \\ \mathcal{F}(\Delta) & \mathcal{E}(\Delta) & V_i + V_i^T & * \\ 0 & 0 & V_i & X_i \end{bmatrix} \succ 0, \forall \Delta \in \boldsymbol{\Delta}. \tag{3.39}$$

Lemma 5 can be applied to the inequality (3.39) with the choice of $\mathcal{Y}$ and its sub-blocks as follows:

$$\mathcal{Y} = \left[\begin{array}{c|ccccc} D_p & A_dW{+}B_dN & E_d & 0 & 0 & \\ \hline 0 & \frac{1}{2}\Lambda_i & 0 & 0 & 0 & \\ 0 & 0 & \frac{1}{2}D^T\Gamma_i D & 0 & 0 & \\ B_p & AW{+}BN & E & V_i & 0 & \\ 0 & 0 & 0 & V_i & \frac{1}{2}X_i & \end{array}\right]. \tag{3.40}$$

This application leads to the condition in (3.36). □

Note that **Theorem** 3 is still not an LMI feasibility problem since the $(1,1)$ block of (3.35) depends nonlinearly on (W, X_i). In Section *3.5.1*, we propose a successive linearization approach to handle this nonlinearity, where it will be integrated into an iterative scheme.

3.5 Iterative Algorithms for the Computation of LC-RCI Sets

We develop two iterative algorithms to obtain LC-RCI sets of desirably large volumes. Our presentation is built on the dilated invariance conditions (3.26), (3.35) and (3.36). We hence start by linearizing (3.35) within an iterative scheme. A lemma is then formulated as the basis of the iterative volume maximization algorithms presented at the end of the section. In order to clearly present our iterative algorithms, we add a superscript ^{k} to the solutions obtained from the k'th iteration, e.g. W is the generic matrix variable, W^k is its value obtained from the k'th iteration.

3.5.1 Linearization

To linearize the $(1,1)$ block of (3.35), we make use of **Lemma 4**, which gives

$$W^T X_i^{-1} W \succcurlyeq \underbrace{Y_i^T W + W^T Y_i - Y_i^T X_i Y_i}_{\mathcal{N}_i}, \tag{3.41}$$

where Y_i is an arbitrary matrix of compatible dimension. This inequality allows us to obtain a sufficient condition for (3.35) by replacing its $(1,1)$ block with $\mathcal{N}_i$. The sufficient condition that we obtain in this fashion will be an LMI only if Y_i is a fixed matrix. Based on Remark (1), we propose choosing Y_i at the $k+1$'st iteration based on the solutions at the k'th iteration as follows:

$$Y_i = Y_i^k \triangleq \left(X_i^k\right)^{-1} W^k. \tag{3.42}$$

Hence, a sufficient LMI condition for (3.35) can be used at the $k+1$'st iteration of an iterative scheme as follows:

$$\begin{bmatrix} (Y_i^k)^T W + W^T Y_i^k - (Y_i^k)^T X_i Y_i^k & e_i \phi_i \\ \phi_i e_i^T & \phi_i \end{bmatrix} \succ 0, i = 1, .., n. \tag{3.43}$$

As explained in **Lemma 4**, such an iterative scheme will alleviates the conservatism introduced by (3.41) iteratively. Note that $(W, X_i) = (W^k, X_i^k)$ is a feasible solution for (3.43), this is a crucial property and will used next to prove the recursive feasibility of the iterative schemes. Lastly, we also note that (3.41) and (3.43) together imply that W is invertible.

3.5.2 Iterative Procedure for Volume Maximization

We now formulate a lemma that will form the basis of the proposed iterative volume maximization algorithms. For a concise and general formulation, we express the system constraints and invariance condition at the $k+1$'st iteration (via block-diagonal concatenation) as a single LMI

$$\mathcal{L}(W, L, Y^k) \succ 0, \tag{3.44}$$

where L represents all the variables other than W and $Y^k = \mathsf{diag}(Y_1^k, \cdots, Y_n^k)$ introduced in (3.42). We emphasize that

$$\mathcal{L}(W^k, L^k, Y^k) \succ 0, \tag{3.45}$$

for the system constraints and invariance conditions. This follows directly from the fact that $(W, X_i) = (W^k, X_i^k)$ is a feasible solution for (3.43).

The volume of $\boldsymbol{\mathcal{C}}$ is proportional to $|\mathsf{det}(W)|$ as noted in [42]. In order to obtain an RCI set with a desirably large volume, we hence need to solve a determinant maximization problem under LMI conditions. Nevertheless, such a problem can be easily solved only when W is imposed to be symmetric; [81]. Enforcing symmetry on W would typically introduce conservatism and is hence undesirable.

The main motivation behind an iterative scheme is to avoid the requirement that W is symmetric. For this, we formulate our determinant maximization problem in such a way that the consecutive solutions fulfill the condition

$$|\mathsf{det}(W^{k+1})| \geq |\mathsf{det}(W^k)|. \tag{3.46}$$

To this end, we introduce a new matrix variable $\overline{Z}$ required to satisfy

$$W^T W \succcurlyeq \overline{Z} \succ 0. \tag{3.47}$$

It follows from Minkowski determinant inequality that

$$\mathsf{det}(W^T W) = |\mathsf{det}(W)|^2 \geq \mathsf{det}(\overline{Z}). \tag{3.48}$$

Since (3.47) is not an LMI, we consider replacing it with

$$W^T W^k + (W^k)^T W - (W^k)^T W^k \succcurlyeq \overline{Z} \succ 0. \tag{3.49}$$

By showing that (3.49) is a sufficient condition for (3.47), we can provide an iterative solution to the determinant maximization problem under LMI constraints as follows:

Lemma 7: *Consider an optimization problem formulated at the $k+1$'st* *iteration of an iterative scheme as follows:*

$$\begin{array}{cl} \underset{W, L, \overline{Z}}{\text{maximize}} & \log\det(\overline{Z}) \\ \text{subject to :} & (3.44) \text{ and } (3.49) \end{array}$$

The solutions of this optimization problem satisfy the determinant condition in (3.46) provided that (3.45) is true.

Proof. From

$$\begin{aligned} W^T W = {} & \overbrace{(W - W^k)^T (W - W^k)}^{\succcurlyeq 0} \\ & + \underbrace{(W - W^k)^T W^k + (W^k)^T (W - W^k) + (W^k)^T W^k}_{W^T W^k + (W^k)^T W - (W^k)^T W^k}, \end{aligned} \qquad (3.50)$$

it directly follows that (3.49) implies (3.47). From (3.48), the solution of the $k+1$'st iteration satisfies

$$|\det(W^{k+1})|^2 \geq \det(\overline{Z}^{k+1}). \qquad (3.51)$$

On the other hand, as (3.45) is known to be true, $(W, L, \overline{Z}) = (W^k, L^k, (W^k)^T W^k)$ will be a feasible solution of the $k+1$'st optimization problem since (3.49) would then also hold true with equality. With the problem formulated as the maximization of $\det(T)$, this implies that

$$\det(\overline{Z}^{k+1}) \geq \det((W^k)^T W^k) = |\det(W^k)|^2. \qquad (3.52)$$

Condition (3.46) is clearly ensured by (3.51) and (3.52). $\square$

3.5.3 Algorithms with Dilated and Standard LMIs

In order to formulate two iterative volume maximization problems, we simply apply Lemma 7 by specifying the conditions summarized in (3.44). The lemma is applicable with dilated as well as standard LMI conditions, since (3.45) is valid in both cases (and trivially so with standard LMIs in which Y^k is absent).

Algorithm 1 (based on dilated LMIs)

Optimization at $k+1$'st iteration:

$$\begin{array}{ll} \underset{W,N,\overline{Z},X_i,V_i,P_i\in\mathcal{P}_{pol},\Lambda_i,\Gamma_i\succcurlyeq 0,\phi_i}{\text{maximize}} & \mathsf{log\,det}(\overline{Z}) \\ \text{subject to:} & (3.17),(3.19),(3.26),(3.36),(3.43),(3.49) \end{array}$$

Initial Optimization to Compute W^0: Condition (3.49) is removed and $\mathsf{log\,det}(\overline{Z})$ is changed to $\mathsf{log\,det}(W)$ or $\mathsf{log\,det}(W+W^T)$ (to avoid conservatism due to symmetric W); (3.43) is imposed with $Y_i^k \to I$.

Algorithm 2 (based on standard LMIs)

Optimization at $k+1$'st iteration:

$$\begin{array}{ll} \underset{W,N,\overline{Z},P_i\in\mathcal{P}_{pol},\Lambda_i,\Gamma_i\succcurlyeq 0,\phi_i}{\text{maximize}} & \mathsf{log\,det}(\overline{Z}) \\ \text{subject to:} & (3.17),(3.19),(3.26),(3.30),(3.49) \end{array}$$

Initial Optimization to Compute W^0: Condition (3.49) is removed and $\mathsf{log\,det}(\overline{Z})$ is changed to $\mathsf{log\,det}(W)$ or $\mathsf{log\,det}(W+W^T)$ (to avoid conservatism due to symmetric W).

Termination: $|\mathsf{det}(W^{k+1})|/|\mathsf{det}(W^k)| \leq 1+\epsilon$ with a desirably small $\epsilon > 0$ or when $k > \eta_{ite}$.

Remark 2: At each iteration of our algorithms, we need to solve a generalized SDP, which can be solved efficiently; [81]. Lemma 7 guarantees recursive feasibility (i.e. existence of a solution at each iteration) and iterative volume increase. Hence both algorithms would converge to stationary points, which need not be a global (or even local) optimum of the volume maximization problem under system and invariance constraints. In practice, the termination criterion needs to be adjusted in a way to avoid numerical problems as well as an early termination.

Remark 3: We can modify **Algorithm 1** and **Algorithm 2** to compute desirably small RCI sets. This can be achieved by minimizing $|\mathsf{det}(W)|$ iteratively in a similar fashion as done in the algorithm above. However determinant minimization problems are concave and are not tractable for solving. Thus, as proposed in [47], we instead minimize $\mathsf{trace}(\underline{Z})$ at each iteration and replace (3.49) by $\begin{bmatrix} \underline{Z} & W^T \\ W & I \end{bmatrix} \succcurlyeq 0$ in the algorithms. For further details, we refer the reader to [47].

3.6 Illustrative Example

3.6.1 $4-D$ Vehicle Lateral Dynamics

In this section, the proposed algorithms are demonstrated with a standard bicycle model for vehicle lateral dynamics; see [82]. The continuous-time model is described by

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 & V_x & 0 \\ 0 & \frac{-171.2893}{V_x} & 0 & \frac{85.2523-V_x^2}{V_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{42.1875}{V_x} & 0 & \frac{-199.6488}{V_x} \end{bmatrix}}_{A_c(V_x)} \underbrace{\begin{bmatrix} e_y \\ \dot{y} \\ e_\psi \\ \dot{\psi} \end{bmatrix}}_{x}$$
$$+ \underbrace{\begin{bmatrix} 0 & 0 \\ 65.8919 & 0 \\ 0 & 0 \\ 43.6411 & 0.2287 \end{bmatrix}}_{B_c} \underbrace{\begin{bmatrix} \delta \\ \mu_b \end{bmatrix}}_{u} + \underbrace{\begin{bmatrix} 0 \\ 0.0018 \\ 0 \\ -0.0022 \end{bmatrix}}_{E_c} \underbrace{V_w^2}_{w}, \quad (3.53)$$

where $e_y\,[rad]$ is the lateral error wrt a reference path, $\dot{y}\,[m/s]$ is the lateral velocity, $e_\psi\,[rad]$ is the orientation error, $\dot{\psi}\,[rad/s]$ is the yaw rate, $\delta\,[rad]$ is the steering angle, $\mu_b = 10^{-3} \cdot M_b$ where $M_b\,[Nm]$ is the braking yaw moment and $V_w\,[m/s]$ is the wind velocity. The longitudinal velocity is assumed to vary around the nominal value $V_0 = 60/3.6\,[m/s]$ as $V_x = V_0 + \delta_1$ with $|\delta_1| \leq 10/3.6\,[m/s]$. Notice that the matrix A_c has rational dependence on V_x and thereby on the uncertain parameter δ_1. The state and control input constraints are expressed as

$$\begin{array}{ccc} |e_y| \leq 0.4\ m, & |\dot{y}| \leq 3\ m/s, & |e_\psi| \leq \frac{10\cdot\pi}{180}\ rad, \\ |\delta| \leq \frac{5\cdot\pi}{180}\ rad, & |M_b| \leq 10^3\ Nm, & |V_w| \leq 10\ m/s. \end{array} \quad (3.54)$$

We first discretize the system in (3.53) based on the Euler approach with a sampling time of $\tau_s = 1/40\,s$. A discrete-time LFT representation is then obtained as in (3.4) with the uncertainty block $\Delta = \mathsf{diag}(\delta_1, \delta_1, \delta_1, \delta_1)$ and the following system matrices:

$$B = \tau_s B_c,\ E = \tau_s E_c,\ B_d = 0,\ E_d = 0, \quad (3.55)$$

$$
\left[\begin{array}{c|c} A & B_p \\ \hline A_d & D_p \end{array}\right] =
$$

$$
\left[\begin{array}{cccc|cccc}
1 & 0.0250 & 0.4167 & 0 & 0.025 & 0 & 0 & 0 \\
0 & 0.7431 & 0 & -0.2888 & 0 & 0.0015 & 0 & 0 \\
0 & 0 & 1 & 0.0250 & 0 & 0 & 0 & 0 \\
0 & 0.0633 & 0 & 0.7005 & 0 & 0 & 0.0015 & 0 \\
\hline
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 10.2774 & 0 & -21.7818 & 0 & -0.060 & 0 & 1 \\
0 & -2.5313 & 0 & 11.9789 & 0 & 0 & -0.060 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0
\end{array}\right]. \tag{3.56}
$$

By applying **Algorithm 1** on the system description (3.55), (3.56) with constraints (3.54), the matrix couple (W, K) is computed after $\eta_{ite} = 73$ iterations as follows:

$$
\left[\begin{array}{c} W \\ \hline K \end{array}\right] = \left[\begin{array}{cccc}
0.3455 & -0.0182 & -0.0202 & 0.0161 \\
-0.2084 & 0.5180 & -0.0890 & -0.0217 \\
-0.0354 & -0.0065 & 0.0280 & -0.0084 \\
0.1540 & 0.0909 & -0.0596 & 0.2195 \\
\hline
-0.0808 & -0.0504 & -1.3707 & -0.2017 \\
1.9478 & 0.7785 & -6.2921 & -4.6698
\end{array}\right]. \tag{3.57}
$$

The algorithm was implemented in CVX [68] using SeDuMi solver on a 3.1 GHz Intel Core i7-555U computer with 8 GB RAM, running macOS. Each iteration of **Algorithm 1** had 21 LMIs and 188 scalar inequalities in 306 variables and was solved in 14 seconds on average. The plots in Fig. 3.1-(a), (b) and (c) show the projections of admissible states $\boldsymbol{\mathcal{X}}$ in (3.5) and maximum volume set $\mathcal{C}$ obtained from **Algorithm 1**, respectively. To demonstrate the invariance of $\mathcal{C}$, the state trajectories are plotted in Fig. 3.1-(a), (b) and (c) (in black) by randomly varying the input disturbance and the uncertain parameters within the chosen bounds. **Algorithm 2** failed to generate any set in this example, which indicates that dilation indeed leads to a reduction in conservatism. We should, nevertheless, mention that **Algorithm 2** delivers useful results with significantly reduced computational complexity in other examples.

In order to be able make comparisons with the existing algorithm by [47], we first introduce a new parameter as $a_0+\delta_2 = 1/(V_0+\delta_1)$, where, $a_0 = 0.0617$ and $|\delta_2| \leq 0.0103$ and thus view (3.53) as a system that has affine dependence

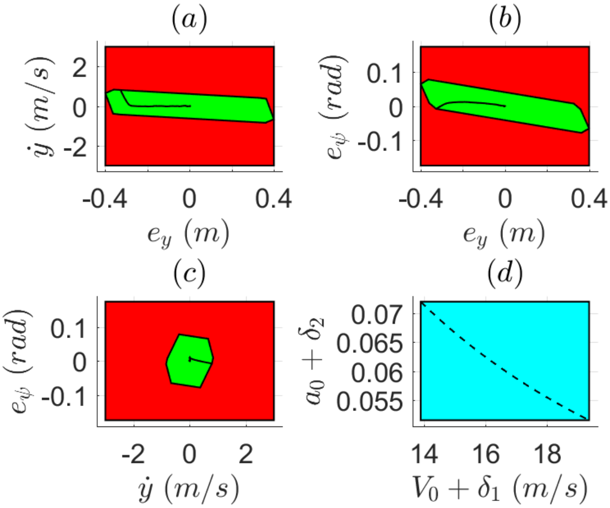

Figure 3.1: $(a), (b), (c)$-Projection of admissible set $\boldsymbol{\mathcal{X}}$ (red) and the RCI set $\mathcal{C}$ from **Algorithm 1** with (3.55), (3.56) (green) on shown axis and (d)-Uncertainty set in (δ_1, δ_2) domain (cyan).

on (δ_1, δ_2). In the new uncertainty description, one typically needs to use a region that covers the original uncertainty set, i.e. the dashed line in Fig. 3.1-(d). The uncertainty region is formed in this representation as rectangular in order to be able to use the algorithm from [47]. This approach would clearly be quite conservative, since the cyan region in Fig. 3.1-(d) is added to the uncertainty set unnecessarily. In fact, we were not able to generate any set using **Algorithm 1**, **Algorithm 2** or the method of [47] with this description. This demonstrates the benefit of the proposed approach, which facilitates a direct handling of rational parameter dependence.

3.6.2 Academic Example

For further comparison with the other methods, we investigate an example extracted from the [83]. We compare the set obtained from the proposed algorithm with the one presented in [83], and also with the method generating ellipsoidal RCI sets [37].

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -7 & -7 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u. \tag{3.58}$$

The system (3.58) is assumed to be under following constraints

$$|x_i| \leq 10,\ i = 1, 2 \text{ and } |u| \leq 10, \tag{3.59}$$

Considering a sampling time of 0.01 units and using the **Algorithm 1**, we were able to find a RCI set $\mathcal{C}$ (see Figure 3.2) defined by

$$W = \begin{bmatrix} 9.3148 & -0.6851 \\ 0.0000 & 10.0000 \end{bmatrix},\ K = \begin{bmatrix} -0.2898 & -0.7496 \end{bmatrix} \tag{3.60}$$

Though we sought for linear state feedback and a low complexity RCI set, we still obtain a larger set (**Figure 3.2**) in terms of volume to the one found in [83]. The approach in [83] does not restrict the control input to any particular form, which is hoped to eliminate conservatism significantly. But evidently, conservatism is introduced at various steps of derivations or due to the non-convexity of the resulting problem. In the **Figure 3.2**, we can see that the sets obtained from the proposed algorithm are also larger than the ellipsoidal set computed using approach from [37]. This demonstrates the advantage

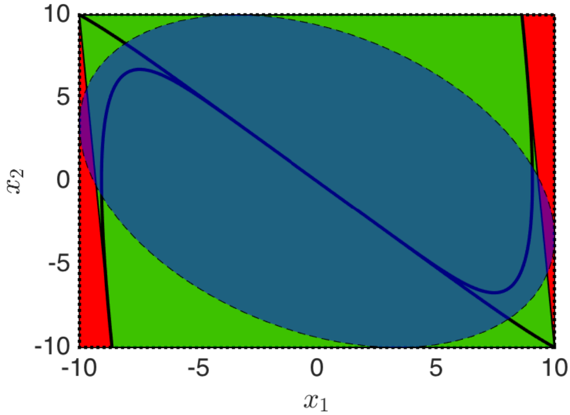

Figure 3.2: RCI set $\mathcal{C}$ using proposed **Algorithm 1** (green), admissible set $\mathcal{X}$ (red) and a closed-loop trajectory (black).

of using polytopic candidate set over the ellipsoidal. Of course, one should consider making comparisons in more examples to claim the benefit of using low complexity set representation over ellipsoidal, which in general may not be valid.

3.6.3 Oscillator

Consider an uncertain linear system defined by

$$x(k+1) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} \delta(k) \\ 1 \end{bmatrix} u(k), \tag{3.61}$$

where $|\delta| \leq c$ is an unknown time-varying parameter. It is known that the system (3.61) does not admit a quadratic control Lyapunov function for $c > 1$ but it admits one for $c \leq 1$, see [84]. We investigated the LC-RCI set computation problem for the system by imposing constraints as $|x_i| \leq 2$, $\forall i = 1, 2$ and $|u| \leq 1$. Using proposed **Algorithm 1**, we obtain RCI set for $0 \leq c \leq 1$ (see Figure 3.3 and Figure 3.4).

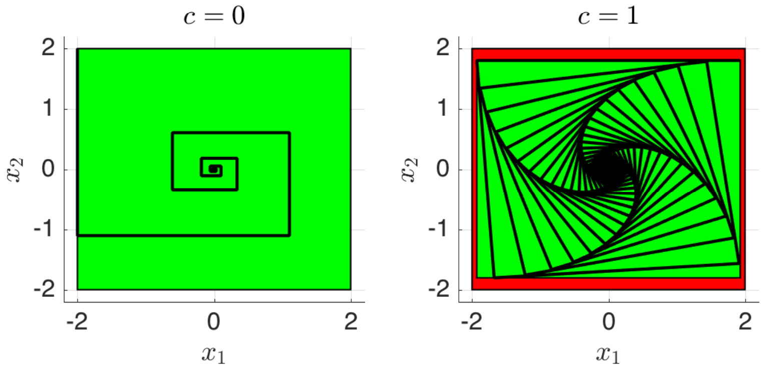

Figure 3.3: RCI set $\mathcal{C}$ using proposed **Algorithm 1** (green), admissible set $\mathcal{X}$ (red) and a closed-loop trajectory (black) for the indicated values of c.

$$\left[\frac{W}{K}\right]_{c=0} = \left[\begin{array}{cc} 2.0000 & 0.0000 \\ 0.0000 & 2.0000 \\ \hline -0.4481 & 0.0000 \end{array}\right], \quad \left[\frac{W}{K}\right]_{c=1} = \left[\begin{array}{cc} 1.9304 & 0.0000 \\ 0.0000 & 1.8032 \\ \hline -0.0659 & 0.0000 \end{array}\right]$$

3.6.4 Minimal RCI sets

We now compare minimal RCI sets obtained using the proposed approach with an recent algorithm [48]. For comparison, we consider an example from [48].

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u(k) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} w(k) \tag{3.62}$$

We assume following system constraints

$$|u(k)| \leq 1, \; |w(k)| \leq [0.1, \, 0.1]^T \tag{3.63}$$

Using the proposed **Algorithm 1** with the modifications suggested in Remark 3, we obtain the RCI set $\mathcal{C}$ shown in **Figure** 3.5 and defined by

$$\left[\frac{W}{K}\right] = \left[\begin{array}{cc} 0.1458 & -0.1264 \\ 0.0084 & 0.2528 \\ \hline -0.9999 & -1.5000 \end{array}\right] \tag{3.64}$$

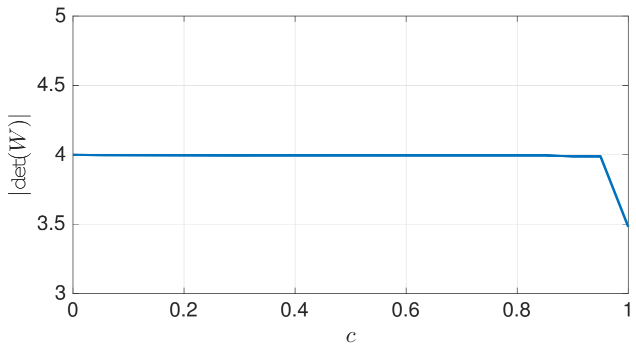

Figure 3.4: $|\det(W)|$ by varying c

Since the method in [48] is only applicable to the autonomous system, we obtain the closed-loop description by using the LQR controller with tuning matrices $Q = I$ and $R = 1$ (as done in [48]). Then applying the method, we compute minimal RCI sets $\boldsymbol{\mathcal{S}}_i$ (shown in **Figure** 3.5) with different representational complexity $i = 6, 10, 20$. This example demonstrates the benefit of our strategy in which we optimize the volume of the RCI set over controller gain K.

3.6.5 Nonlinear System

Lastly, we consider a bilinear continuous-time system from [37], [46] to illustrate the advantage of the proposed algorithm's ability to handle more general system description. The system is described as

$$\dot{x}_1 = x_2 + u(\mu + (1-\mu)x_1), \tag{3.65}$$

$$\dot{x}_2 = x_1 + u(\mu - 4(1-\mu)x_2), \tag{3.66}$$

where the coupling coefficient $\mu = 0.9$. The system is discretized with sampling time 0.1 and is subject to the constraints $|x| \leq [4,\ 4]^T$ and $|u| \leq 2$.

To use [46], we first find the polytopic representation of the system wrt to some inclusion polytope to compute the RCI sets for a nonlinear system. The choice of the inclusion polytope has a significant effect on the size of the RCI set. Nevertheless, the optimal inclusion polytope cannot, in general, be

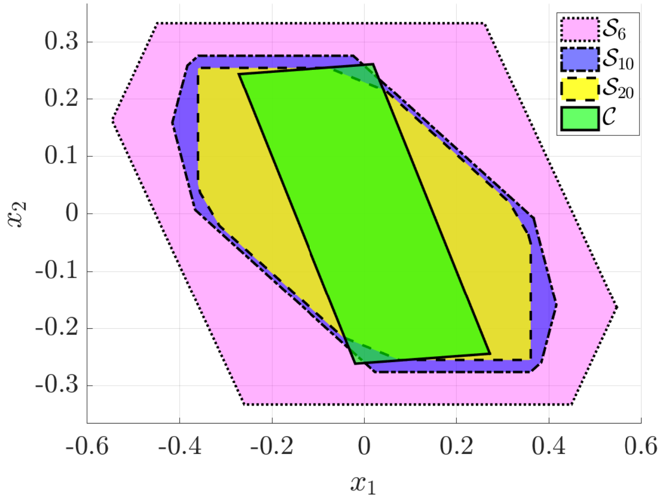

Figure 3.5: Minimal RCI set $\boldsymbol{\mathcal{S}}_i$, $i = 6, 10, 20$ obtained using [48], and the minimal RCI set $\boldsymbol{\mathcal{C}}$ using **Algorithm 1**

computed systematically. Thus [46] proposes an approach to compute RCI sets in which the difference between the inclusion polytope and the RCI set is arbitrarily small. In **Figure** 3.6, we show LC-RCI set $\boldsymbol{\mathcal{S}}$ with $\mathsf{Volume}(\boldsymbol{\mathcal{S}}) = 11.8408$ obtained using [46] and ellipsoidal RCI set $\boldsymbol{\mathcal{E}}$ with $\mathsf{Volume}(\boldsymbol{\mathcal{E}}) = 9.2208$ obtained using [37].

For comparison with the proposed algorithm, we represent (3.65) in the form (3.1). Since we use LFT representation of the system, we do not have to look for an optimal inclusion polytope. Using the proposed **Algorithm 1**, we then compute the relatively larger RCI set $\boldsymbol{\mathcal{C}}$ with $\mathsf{Volume}(\boldsymbol{\mathcal{C}}) = 14.5653$.

$$\left[\begin{array}{c} W \\ \hline K \end{array}\right] = \left[\begin{array}{rr} 1.9582 & -1.6350 \\ -0.0768 & 1.9237 \\ \hline -1.0566 & -0.8980 \end{array}\right] \tag{3.67}$$

Lastly, to demonstrate invariance, we also show the trajectories of the discretized nonlinear system in closed-loop with the obtained controller.

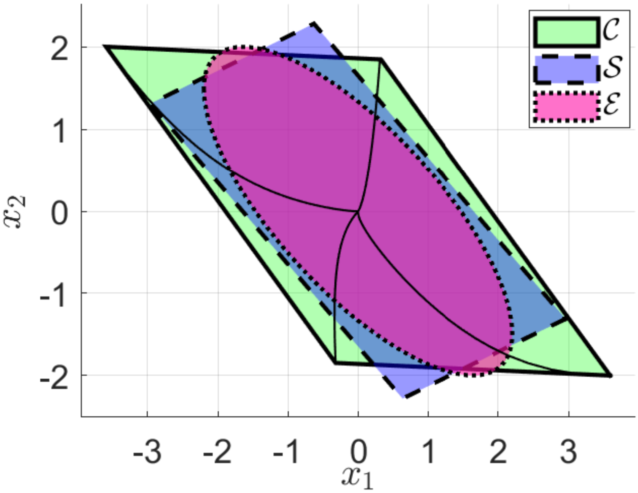

Figure 3.6: Maximal RCI set $\boldsymbol{\mathcal{S}}$ obtained using [46], Maximal RCI set $\boldsymbol{\mathcal{E}}$ obtained using [37] and Maximal RCI set $\boldsymbol{\mathcal{C}}$ obtained using proposed algorithm.

CHAPTER 4

Full-Complexity of Control-Invariant Sets for Uncertain Systems

4.1 Introduction

In the previous chapter, the complexity of the RCI set was fixed and equal to twice the number of states. It is evident that an RCI set with fixed complexity may not be the most convenient choice to approximate the maximal or minimal RCI set. Such observation motivates us to develop an algorithm that gives the flexibility to choose the RCI set's complexity. This chapter proposes an iterative algorithm to find an RCI set of desired complexity along with the associated linear state feedback controller. Furthermore, for controller syntheses like MPC and IBC, it is desirable to have an RCI set where certain performance can be guaranteed. Thus, in the proposed algorithm, one can optionally impose desired quadratic performance level as additional constraints. The proposed result builds upon the previous chapter and [55].

4.2 Brief Literature Survey and Comparison

In literature, there have been many methods to compute RCI sets with a predefined complexity. For instance, [45] proposed a method to calculate an RCI set with a desired number of vertices. This method is only applicable to LTI systems, and its computational complexity exponentially grows with the system dimension. The procedures in [80], [85] can also be used to compute RCI sets with a restricted or predefined number of hyperplanes, as the earlier method, their application is limited to LTI systems. An approach to compute minimal RCI sets with predefined complexity for autonomous LTI system was also proposed in [48]. The authors in [49] extend the results presented in [47], to compute RCI sets with a predefined number of hyperplanes for affine parameter-dependent systems. The authors use the same relaxation tools to derive tractable LMI conditions for invariance.

Like the previous chapter, we first introduce a state transformation in which the candidate RCI set is mapped into a known polytope. The system constraints (state and input constraints) in the transformed space are found to be affine inequalities. We consider a more general input disturbance set compared to [54], [56]. Using numerical examples, we show that by increasing representational complexity, the proposed algorithm computes larger RCI sets, and feasible solutions are obtained for larger bounds on uncertainty and disturbance. Hence, representational complexity can be used as an additional tuning parameter to get an RCI with a larger volume.

4.3 Problem Formulation

4.3.1 System and Constraints

We consider a discrete-time uncertain system described by

$$x(k+1) = \mathcal{A}(\Delta(k))x(k) + \mathcal{B}(\Delta(k))u(k) + \mathcal{E}(\Delta(k))w(k), \tag{4.1}$$

$$y(k) = Cx(k) + Du(k), \tag{4.2}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $w \in \mathbb{R}^{n_w}$ and $y \in \mathbb{R}^{n_y}$ are the state, control input, disturbance and the performance output vectors respectively. The system matrices in (4.1) depend rationally on an uncertain (possibly) time-varying parameter matrix Δ that satisfies (3.2) and (3.3). Thus, we can always re-express (4.1) compactly with an LFT of the form (3.4).

The system is subject to the following polytopic state and input constraints, respectively:

$$\begin{aligned} \boldsymbol{\mathcal{X}} &= \{x : Hx \leq \mathbf{1}\}, \\ \boldsymbol{\mathcal{U}} &= \{u : Gu \leq \mathbf{1}\}, \end{aligned} \tag{4.3}$$

where $H \in \mathbb{R}^{n_h \times n_x}$ and $G \in \mathbb{R}^{n_g \times n_u}$ are given matrices and $\mathbf{1}$ represents the vector of ones of compatible dimension. We assume $w \in \boldsymbol{\mathcal{W}}$, with $\boldsymbol{\mathcal{W}}$ a compact (i.e. closed and bounded) polytopic set containing the origin and represented by the convex-hull of a set of finitely many known vertices $\boldsymbol{\mathcal{W}}_c = \{w^1, w^2, \cdots, w^\rho\}$:

$$\boldsymbol{\mathcal{W}} = \mathsf{conv}(\boldsymbol{\mathcal{W}}_c) \triangleq \left\{ \sum_{j=1}^{\rho} \beta_j w^j : \sum_{j=1}^{\rho} \beta_j = 1, \beta_j \in [0,1] \right\}. \tag{4.4}$$

Note that $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{W}}$ are allowed to be non-symmetric unlike [47], [54], [56], [80] where some or all the constraints are assumed to be symmetric.

4.3.2 Candidate RCI set and Invariance Inducing Controller

We consider the static state feedback control law

$$u(k) = Kx(k), \tag{4.5}$$

where $K \in \mathbb{R}^{n_u \times n_x}$ is a feedback gain matrix and the resulting closed loop dynamics (using (4.1) and (4.5)) are

$$x^+ = (\mathcal{A}(\Delta) + \mathcal{B}(\Delta)K)x + \mathcal{E}(\Delta)w, \tag{4.6}$$

The set $\boldsymbol{\mathcal{C}}$ is referred to as *robustly invariant* for the system (4.6), if the following condition is satisfied:

$$x \in \boldsymbol{\mathcal{C}} \;\Rightarrow\; x^+ \in \boldsymbol{\mathcal{C}},\ \forall w \in \boldsymbol{\mathcal{W}},\ \forall \Delta \in \boldsymbol{\Delta}. \tag{4.7}$$

The set $\mathcal{C}$ has to satisfy the state and input constraints, this implies $\mathcal{C} \subseteq \mathcal{X}$ and $K\mathcal{C} \subseteq \mathcal{U}$, which can be further expressed as

$$x \in \mathcal{C} \;\Rightarrow\; x \in \mathcal{X}, \tag{4.8}$$

$$x \in \mathcal{C} \;\Rightarrow\; u = Kx \in \mathcal{U}. \tag{4.9}$$

We assume $\mathcal{C}$ to be symmetric with respect to origin and described as

$$\mathcal{C} = \left\{ x \in \mathbb{R}^n : -\mathbf{1} \leq PW^{-1}x \leq \mathbf{1} \right\}, \tag{4.10}$$

where $P \in \mathbb{R}^{n_p \times n_x}$ and $W \in \mathbb{R}^{n_x \times n_x}$. The invertibility of the matrix W would be later guaranteed by the LMI conditions for invariance.

4.3.3 Performance Constraint

In this chapter, along with invariance, we consider quadratic performance constraints defined as

$$\sum_{k=0}^{\infty} y(k)^T y(k) \leq \gamma, \; \forall x(k) \in \mathcal{C}, \tag{4.11}$$

where $0 \leq \gamma < \infty$. The performance constraint (4.11) will be only feasible if $w(k) = 0, \forall k \geq 0$ or eventually becomes zero after a finite time. Hence, for meaningful problem formulation, we assume $w(k) = 0$ whenever (4.11) is considered, and thus we can ignore matrix $\mathcal{E}$ in this case.

The problem considered in this chapter can now be stated as follows:

Problem 2: For a given matrix P, scalar γ and the discrete-time system (4.1) subject to constraints (4.3) and (4.4), find (W, K) such that:

1. The invariance condition (4.7) holds;

2. All elements of the set $\mathcal{C}$ in (4.10) satisfy the state and input constraints (4.8) and (4.9), respectively.

3. For any initial $x(k) \in \mathcal{C}$, performance constraints (4.11) is satisfied.

In this chapter, we are interested in computing RCI sets with representational complexity $2n_p$, which can be achieved by choosing the matrix P appropriately. If we select $P = I$ and the performance constraints (4.11) is ignored then the Problem 2 is equivalent to the one solved in the previous chapter. Similar to the last chapter, we search for the largest and the smallest set $\boldsymbol{\mathcal{C}}$ solving Problem 2.

4.4 Tractable Formulation of the Constraints and Invariance Conditions

In this section, we rewrite (4.7), (4.8), (4.9) and (4.11) in a way such that Problem 2 can be solved efficiently. We proceed along the lines in [54], [56] by introducing a convenient coordinate transformation such that state and control input constraints are expressed as affine inequalities. The invariance condition and the performance constraints are expressed as a set of LMIs using Lemma 5. We thus provide a solution to Problem 2 in the form of an LMI feasibility problem.

4.4.1 System Constraints

We now introduce a state transformation as

$$\theta \triangleq W^{-1}x \Leftrightarrow x = W\theta. \tag{4.12}$$

This allows us to express the set $\boldsymbol{\mathcal{C}}$ as

$$\boldsymbol{\mathcal{C}} = \{W\theta \in \mathbb{R}^n : \theta \in \boldsymbol{\Theta}\}, \tag{4.13}$$

where $\boldsymbol{\Theta}$ is a symmetric set defined as follows:

$$\boldsymbol{\Theta} \triangleq \{\theta \in \mathbb{R}^n : -\mathbf{1} \leq P\theta \leq \mathbf{1}\}. \tag{4.14}$$

We have thus introduced a θ-state-space in which the candidate invariant set is a known symmetric set around the origin. The corresponding polytopic set in the x-state-space will be completely determined by the choice of W.

Since P is a known matrix, the symmetric set $\boldsymbol{\Theta}$ can be always expressed as the convex hull of the finitely many vertices given as:

$$\boldsymbol{\Theta} = \mathsf{conv}\left(\left\{\theta^1, \ldots, \theta^{2\sigma}\right\}\right). \tag{4.15}$$

where σ is some known positive integer determined by the choice of P. To derive conditions for the state and input constraints to hold, we first express (4.3) in the θ-state-space using (4.12). Satisfying these inequalities at the vertices θ^j ensures that they are satisfied over the whole set $\boldsymbol{\Theta}$ as well. Therefore, we can express (4.8) in terms of W as follows:

$$HW\theta \leq \mathbf{1}, \forall\theta \in \boldsymbol{\Theta} \;\Leftrightarrow\; HW\theta^j \leq \mathbf{1},\; j = 1, \ldots, 2\sigma. \tag{4.16}$$

Similarly to express the control input constraints, we first introduce a new matrix variable as

$$N \triangleq KW \quad \Leftrightarrow \quad K = NW^{-1}. \tag{4.17}$$

The control input constraints in (4.9) then read as

$$GN\theta \leq \mathbf{1}, \forall\theta \in \boldsymbol{\Theta} \;\Leftrightarrow\; GN\theta^j \leq \mathbf{1},\; j = 1, \ldots, 2\sigma. \tag{4.18}$$

The system constraints (4.16) and (4.18) are affine and are identified by $n_h \times 2\sigma$ and $n_g \times 2\sigma$ scalar inequalities. Note that, if the sets $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{U}}$ are symmetric, half of the constraints in (4.16) and (4.18) are redundant and hence removable. This is a consequence of the symmetry of the set $\boldsymbol{\Theta}$, which allows arranging θ^j's in a way to have $\theta^{j+\sigma} = -\theta^j$ for $j = 1, \ldots, \sigma$.

4.4.2 Matrix Inequality Conditions for Performance

Towards the tractable formulation of the constraints (4.11), we first introduce a matrix $U \succ 0$ and $U \in \mathbb{R}^{n_x \times n_x}$. Then (4.11) is satisfied if

$$x(k)^T U^{-1} x(k) \leq \gamma,\; \forall x(k) \in \boldsymbol{\mathcal{C}} \tag{4.19}$$

$$x(k+1)^T U^{-1} x(k+1) - x(k)^T U^{-1} x(k) \leq -y(k)^T y(k). \tag{4.20}$$

By summing both the sides of (4.20) from $k = 0$ to $k = \infty$, it can be easily verified that (4.11) holds if (4.19) and (4.20) hold.

In the following lemma, we will present sufficient matrix inequality conditions for (4.19) and (4.20). To this aim it is then convenient to recall a Full-block S-Procedure (**Lemma** 5) about parameter-dependent matrix inequalities. The *scaling* (or *multiplier*) matrix M_y is introduced and is required to satisfy

$$\begin{bmatrix}\Delta^T\\ I\end{bmatrix}^T \underbrace{\begin{bmatrix}Q_y & S_y\\ S_y^T & R_y\end{bmatrix}}_{M_y} \begin{bmatrix}\Delta^T\\ I\end{bmatrix} \preccurlyeq 0,\ \forall\Delta\in\boldsymbol{\Delta}. \tag{4.21}$$

We also recall $\boldsymbol{\mathcal{M}} \triangleq \{M : (2.20)\}$ and $\boldsymbol{\mathcal{M}}_{pol} \triangleq \{M : (2.27)\}$ and from [56] it is known that $\boldsymbol{\mathcal{M}}_{pol} \subseteq \boldsymbol{\mathcal{M}}$.

Lemma 8: *System* (4.6) *satisfies performance constraints* (4.11) *if there exist* $W \in \mathbb{R}^{n_x\times n_x}$, $N \in \mathbb{R}^{n_u\times n_x}$, $U \in \mathbb{R}^{n_x\times n_x}$, $\Pi = \mathsf{diag}(\pi_1,\dots,\pi_{n_p}) \succ 0$, $\pi_i \in \mathbb{R}$ *and* $M_y \in \boldsymbol{\mathcal{M}}_{pol}$ *satisfying*

$$\begin{bmatrix}\gamma-\mathbf{1}^T\Pi\mathbf{1} & 0 & 0\\ 0 & P^T\Pi P & *\\ 0 & W & U\end{bmatrix}\succeq 0 \tag{4.22}$$

$$\begin{bmatrix}W^TU^{-1}W & * & * & *\\ AW+BN & U+B_pR_yB_p^T & 0 & *\\ CW+DN & 0 & I & 0\\ A_dW+B_dN & S_yB_P^T+D_pR_yB_P^T & 0 & \Phi_y\end{bmatrix}\succeq 0 \tag{4.23}$$

where, $\Phi_y = Q_y + S_yD_p^T + D_pS_y^T + D_pR_yD_p^T$

Proof. Using (4.12), (4.17) and (4.2), we can rewrite (4.20) as

$$(\theta^+)^TW^TU^{-1}W\theta^+ - \theta^TW^TU^{-1}W\theta \le -\theta^T(CW+DN)^T(CW+DN)\theta \tag{4.24}$$

Substituting $W\theta^+ = (\mathcal{A}(\Delta)W + \mathcal{B}(\Delta)N)\theta$, and using schur complement

$$\begin{bmatrix}W^TU^{-1}W & * & *\\ \mathcal{A}(\Delta)W+\mathcal{B}(\Delta)N & U & 0\\ CW+DN & 0 & I\end{bmatrix}\succeq 0 \tag{4.25}$$

Now by employing Lemma 5 with the multiplier matrix satisfying (4.21), we obtain (4.23).

Next, we want (4.19) to be satisfied for all $x(k)\in\boldsymbol{\mathcal{C}}$. Using S-Procedure, a sufficient condition can be expressed as

$$\gamma - \theta^TW^TU^{-1}W\theta \ge (1-P\theta)^T\Pi(1+P\theta) \tag{4.26}$$

By writing (4.26) in quadratic form and applying schur complement, we obtain (4.22). $\square$

4.4.3 Matrix Inequality Condition for Invariance

To derive the conditions for invariance, we follows the steps similar to the Section 3.4.2. From (4.13), the invariance condition in (4.7) can be written as

$$1 - (e_i^T P\theta^+)^2 \geq 0,\ i = 1, \ldots, n_p, \tag{4.27}$$

$\forall \theta \in \mathbf{\Theta}$, $\forall w \in \boldsymbol{\mathcal{W}}$, $\forall \Delta \in \mathbf{\Delta}$, where e_i represents the i'th column of the identity matrix of size $n_p \times n_p$. We now multiply (4.27) by a positive scalar variable ϕ_i and lower bound the lhs by a term that is known to be non-negative within the region $\mathbf{\Theta}$ (S-procedure). In this fashion, we obtain a sufficient condition for invariance as follows:

$$\phi_i(1 - (e_i^T PW^{-1}V_i\xi_i)^2) \geq 2\xi_i^T \underbrace{(\mathcal{F}\theta + \mathcal{E}w - V_i\xi_i)}_{0} + \underbrace{(\mathbf{1} + P\theta)^T \Lambda_i (\mathbf{1} - P\theta)}_{\geq 0}, \tag{4.28}$$

with

$$\phi_i > 0,\ \Lambda_i = \mathsf{diag}(\lambda_{i1}, \ldots, \lambda_{in_p}),\ \text{with } \lambda_{ij} \geq 0. \tag{4.29}$$

It is straightforward to verify based on (4.14),(3.32) and (4.29) that the right hand side of (4.28) is nonnegative.

A sufficient condition for invariance can now be obtained by first rearranging (4.28) into the form

$$\varkappa^T \mathcal{P}_i(W, N, V_i, \Lambda_i, \Gamma_i, \phi_i)\varkappa \succcurlyeq 0,\ \forall \varkappa, \forall (w, \Delta) \in (\mathcal{W}, \mathbf{\Delta}), \tag{4.30}$$

where $\varkappa^T = \begin{bmatrix} 1 & \theta^T & -\xi_i^T \end{bmatrix}$ and $\mathcal{P}_i$ is a symmetric matrix. The condition (4.30) thus holds if $\mathcal{P}_i \succcurlyeq 0$, that is

$$\begin{bmatrix} \phi_i - \mathbf{1}^T \Lambda_i \mathbf{1} & 0 & * \\ 0 & P^T \Lambda_i P & * \\ \mathcal{E}w & \mathcal{F} & V_i + V_i^T - V_i^T \mathcal{L}_i V_i \end{bmatrix} \succcurlyeq 0, \tag{4.31}$$

where $*$'s represent entries that are uniquely identifiable from symmetry and

$\mathcal{L}_i \triangleq \phi_i W^{-T} P^T e_i e_i^T P W^{-1}$. Note that the block $(3,3)$ clearly has a nonlinear dependence on ϕ_i, V_i and W. Moreover, the (3.32) shows that $\mathcal{F}$ and $\mathcal{E}$ are parameter dependent matrices, which need to be further manipulated in order to derive tractable LMI conditions. The main result can now be stated in the following theorem:

Theorem 4: *Problem 2 is feasible if there exist* $W \in \mathbb{R}^{n_x \times n_x}$,$Q \in \mathbb{R}^{n_x \times n_x}$, $N \in \mathbb{R}^{n_u \times n_x}$, $X_i = X_i^T \in \mathbb{R}^{n_x \times n_x}$, $V_i \in \mathbb{R}^{n_x \times n_x}$, $\Pi, \Lambda_i \in \mathbb{R}^{n_p \times n_p}$, $\phi_i \in \mathbb{R}$ *as in (4.29) and* $M_y, M_i \in \boldsymbol{\mathcal{M}}_{pol}$ *that satisfy (4.16), (4.18),* (4.22), (4.23), *(4.32) and (4.33), for* $i = 1, \ldots, n_p$, $\forall\, w^j \in \boldsymbol{\mathcal{W}}_c$. *A robust control invariant set can then be obtained as in (4.10) and the state feedback gain* $K = NW^{-1}$.

$$\begin{bmatrix} W^T X_i^{-1} W & \phi_i P^T e_i \\ * & \phi_i \end{bmatrix} \succ 0. \tag{4.32}$$

$$\begin{bmatrix} \phi_i - \mathbf{1}^T \Lambda_i \mathbf{1} & 0 & * & * & * \\ 0 & P^T \Lambda_i P & * & * & * \\ E w^j & AW + BN & V_i + V_i^T + B_P R_i B_p^T & * & * \\ 0 & 0 & V_i & X_i & * \\ E_d w^j & A_d W + B_d N & S_i B_p^T + D_p R_i B_p^T & 0 & \Phi_i \end{bmatrix} \succ 0. \tag{4.33}$$

where, $\Phi_i = Q_i + D_p S_i^T + S_i D_p^T + D_p R_i D_p^T$

Proof. In order to resolve the nonlinearity in the block (3,3) of (4.31), we first introduce the new matrix variable $X_i \succ 0$ such that

$$X_i^{-1} - \mathcal{L}_i \succ 0 \Leftrightarrow X_i^{-1} - \phi_i W^{-T} P^T e_i e_i^T P W^{-1} \succ 0. \tag{4.34}$$

Then by applying Schur complement to (4.34) followed by a congruence transformation we obtain (4.32). From (4.34), the condition (4.31) can be rewritten as

$$\begin{bmatrix} \phi_i - \mathbf{1}^T \Lambda_i \mathbf{1} & 0 & * \\ 0 & P^T \Lambda_i P & * \\ \mathcal{E} w & \mathcal{F} & V_i + V_i^T - V_i^T X^{-1} V_i \end{bmatrix} \succ 0,$$

which followed by Schur complement can be written as

$$\begin{bmatrix} \phi_i - \mathbf{1}^T \Lambda_i \mathbf{1} & 0 & * & * \\ 0 & P^T \Lambda_i P & * & * \\ \mathcal{E} w & \mathcal{F} & V_i + V_i^T & * \\ 0 & 0 & V_i & X_i \end{bmatrix} \succ 0. \tag{4.35}$$

Finally, by using Lemma 5 with the following $\mathcal{Y}$

$$\mathcal{Y} = \left[\begin{array}{c|cccc} D_p & E_d w & A_d W+B_d N & 0 & 0 \\ \hline 0 & \frac{1}{2}(\phi_i-\mathbf{1}^T\Lambda_i\mathbf{1}) & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2}P^T\Lambda_i P & 0 & 0 \\ B_p & Ew & AW+BN & V_i & 0 \\ 0 & 0 & 0 & V_i & \frac{1}{2}X_i \end{array}\right], \tag{4.36}$$

together with (4.4), we obtain the (4.33), which is a sufficient condition for (4.35). In conclusion, conditions (4.32) and (4.33) are sufficient for the feasibility of (4.31). □

Note that (4.32) and (4.23) still have nonlinearity in the block $(1, 1)$. This is linearized by adopting a similar successive linearization scheme presented in Section 3.5.1.

4.5 Iterative Algorithm for the Computation of RCI Set

In this section, we develop an iterative algorithm to obtain a RCI set of desirably large(or small) size. We adopt an indirect approach for volume maximization of the RCI set $\mathcal{C}$. Similar to [80], [83], we maximize the volume of an ellipsoidal set enclosed in the candidate RCI set is iteratively[1]. With this intention, we first introduce an ellipsoidal set

$$\mathcal{Z} \triangleq \left\{x \in \mathbb{R}^n | x^T\overline{Z}^{-1}x \leq 1\right\}, \tag{4.37}$$

and then derive a condition for the set $\mathcal{Z}$ to be contained in $\mathcal{C}$.

Lemma 9: *An ellipsoidal set $\mathcal{Z}$ is contained in $\mathcal{C}$ if there exist a $0 < \mu_i \in \mathbb{R}$ satisfying*

$$\begin{bmatrix} 2\mu_i - 1 & * \\ \mu_i P^T e_i & W^T\overline{Z}^{-1}W \end{bmatrix} \succcurlyeq 0, \ \textit{for}\ i = 1, \ldots, n_p. \tag{4.38}$$

Proof. Starting from (4.12) and using a S-Procedure, a condition for $\mathcal{Z} \subseteq \mathcal{C}$

[1] In chapter 5, we will prove that volume of the set $\mathcal{C}$ is proportional to $|\mathsf{det}(W)|$, if the matrix P is fixed. However, we avoid modifying the original approach proposed in [55] to make readers aware of alternative ways of volume maximization and minimization.

is given by

$$2\mu_i e_i^T(1 - P\theta) \geq (1 - \theta^T W^T \overline{Z}^{-1} W\theta)\ \forall\ i = 1, \ldots, n_p. \tag{4.39}$$

The equation (4.39) can be rearranged as

$$\begin{bmatrix}1\\ \theta\end{bmatrix}^T \begin{bmatrix}2\mu_i - 1 & *\\ \mu_i P^T e_i & W^T\overline{Z}^{-1}W\end{bmatrix} \begin{bmatrix}1\\ \theta\end{bmatrix} \succcurlyeq 0.$$

which holds if the condition (4.38) is satisfied. □

Note that (4.38) has nonlinearity in the block $(2,2)$ similar to the $(1,1)$ block of (4.32) and (4.23). Hence next, we use the similar approach presented in Section 3.5.1 to handle these nonlinearities.

On the lines of (3.42), we introduce L^n, Y_i^n and F^n defined as

$$Y_i^n \triangleq (X_i^n)^{-1}W^n,\ F^n \triangleq (\overline{Z}^n)^{-1}W^n,\ L^n \triangleq (U^n)^{-1}W^n \tag{4.40}$$

where $(W^n, X_i^n, \overline{Z}^n, U^n)$ are the values of $(W, X_i, \overline{Z}, U)$ obtained at the n-th step. Hence, the sufficient LMI condition for (4.23) to be used at $n+1$-st step of our iterative scheme is

$$\begin{bmatrix}(L^n)^T W + W^T L^n - (L^n)^T U L^n & * & * & *\\ AW + BN & Q_y + B_p R_y B_p^T & 0 & *\\ CW + DN & 0 & I & 0\\ A_d W + B_d N & S_y B_P^T + D_p R_y B_P^T & 0 & \Psi_y\end{bmatrix} \succeq 0, \tag{4.41}$$

and the sufficient LMI condition for (4.32) is

$$\begin{bmatrix}(Y_i^n)^T W + W^T Y_i^n - (Y_i^n)^T X_i Y_i^n & e_i P^T \phi_i\\ \phi_i e_i^T P & \phi_i\end{bmatrix} \succ 0. \tag{4.42}$$

Following similar arguments, we can linearize the inequality (4.38) as

$$\begin{bmatrix}2\mu_i - 1 & *\\ \mu_i P^T e_i & (F^n)^T W + W^T F^n - (F^n)^T \overline{Z} F^n\end{bmatrix} \succ 0, \tag{4.43}$$

Finally, we note that invertibility of W is implied by the block $(1,1)$ of (4.42), whereas invertibilty of $\overline{Z}$ is implied by the block $(2,2)$ of (4.43).

Now we are ready to propose the iterative algorithm developed based on Lemma 9, which at the $n+1$-st step is

Algorithm 3: $[n+1$ **step**$]$

$$\begin{array}{ll} \max\limits_{\substack{W, N, \overline{Z}, U, X_i, V_i \\ \Lambda_i, \Pi, M_y, M_i, \phi_i, \mu_i}} & \log\det(\overline{Z}) \\ \text{subject to:} & (4.16), (4.18), (4.22), (4.33), (4.41), (4.42) \text{ and } (4.43) \end{array}$$

It can be easily verified that the solution from step n is feasible at step $n+1$, thus **Algorithm 3** is recursively feasible. As we maximize the volume of the ellipsoid $\boldsymbol{\mathcal{Z}}$ at each iteration, the volume of the set $\boldsymbol{\mathcal{C}}$ increases iteratively until it converges to a stationary point.

We again remark that constraints (4.22) and (4.41) are optional and should be only imposed when (4.11) has to be satisfied, else they can be removed from the algorithm. The conditions for invariance (4.33) and (4.42) are sufficient condition, hence conservative to some extent. Further, by updating the matrix W at each iteration, we are basically scaling and rotating the initial polyhedron to a desirably large volume which is control invariant for some control gain K.

Algorithm 3 requires a user-defined matrix P, which determines the representational complexity of the set $\boldsymbol{\mathcal{C}}$. In general, it is difficult to devise a guideline for the selection of the matrix P following which an RCI set is guaranteed to exist. In fact, this is the common drawback of many similar approaches (see, e.g., [42], [49], [51], [86]), which adopt the optimization-based strategy to compute RCI sets with restricted or predefined complexity. We explain some procedures to select the initial matrix P based on the heuristic in the following remark.

Remark 4: We can always choose matrix P in (5.4), while assuming $W = I$, such that it defines an initial candidate polytope of desired shape and complexity within the state constraints. We emphasize that the candidate polytope need not be RCI. A simple approach to select the candidate set could be to distribute the hyperplanes on an arbitrarily small ball of radius ϵ enclosed within the state constraints. For example, in the two-dimensional case, the i-th row could be selected as

$$\epsilon > 0,\ e_i^T P = \begin{bmatrix} \frac{1}{\epsilon}\cos\left(\frac{\pi(i-1)}{n_p}\right) & \frac{1}{\epsilon}\sin\left(\frac{\pi(i-1)}{n_p}\right) \end{bmatrix},\ i = 1, \ldots, n_p. \tag{4.44}$$

With this choice of the initial candidate set, the state and input constraints are most likely to be satisfied. Thus, at the first iteration, the algorithm now has to guarantee the satisfaction of invariance conditions, which can be achieved by generating a suitable matrix W. Alternatively, having an approximate knowledge of the shape of the RCI set can be very helpful while selecting the initial polytope. Thus one could select the initial set by performing a few iterations of the geometric approach [13] (convergence not needed) to get an initial estimate of the RCI set. Note that this could be still beneficial in the case of a higher-dimensional system where the geometric approach is known to be computationally inefficient and may not even converge.

With different choices of matrix P, **Algorithm 3** calculates different final RCI set $\boldsymbol{\mathcal{C}}$. Although the non-uniqueness of the solution obtained from **Algorithm 3** may be seen as a drawback, we observe that the obtained solutions could be cleverly combined to obtain a larger RCI set, as explained in the following remark.

Remark 5: Note that the convex-hull of the different RCI sets, obtained from different initial conditions in **Algorithm 3**, results in a RCI set (possibly of higher complexity) [1]. To understand this, let $(\boldsymbol{\mathcal{C}}_i, K_i)$, $i = 1, \ldots, m$ be different RCI sets and the corresponding controller obtained using **Algorithm 3** for different matrix P. Any $x \in \mathsf{Conv}(\boldsymbol{\mathcal{C}}_1, \ldots, \boldsymbol{\mathcal{C}}_m)$ can be always written as $x = \sum_{i=1}^{m} \alpha_i x_i$, where $x_i \in \boldsymbol{\mathcal{C}}_i$, and $\sum_{i=1}^{m} \alpha_i = 1$ and $\alpha_i > 0$. It is then easy to show that invariance inducing controller for the convex-hull of the RCI sets is $u = \sum_{i=1}^{m} \lambda_i K_i x_i$.

Remark 6: An algorithm to compute desirably small RCI set can be constructed by minimizing the volume of the ellipse containing the candidate RCI set $\boldsymbol{\mathcal{C}}$. This can be achieved by replacing (4.43) by $\begin{bmatrix} 1 & * \\ W\theta^j & \underline{Z} \end{bmatrix} \succ 0$, $j = 1, \ldots, 2\sigma$ and minimizing the $\mathsf{trace}(\underline{Z})$.

4.6 Examples

4.6.1 Uncertain System

In this section, the **Algorithm 3** is demonstrated with a discrete-time system with rational parameter dependence. The system dynamics are described as in (4.1) by

$$\mathcal{A} = \begin{bmatrix} 1+\delta_1 & 1+\delta_1 \\ 0 & 1+\frac{\delta_2}{1+\delta_1} \end{bmatrix},\ \mathcal{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\ \mathcal{E} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{4.45}$$

where δ_1 and δ_2 are the uncertain parameters. The state and control input constraints are

$$\begin{bmatrix} 0.10 & -0.10 & 0.10 & -0.10 & 0 & 0 \\ 0.15 & -0.10 & -0.15 & 0.15 & 0.25 & -\frac{1}{6} \end{bmatrix}^T x \leq \mathbf{1},\ |u| \leq 3. \tag{4.46}$$

In the sequel, we first illustrate the proposed algorithm by using specific bounds on the uncertain parameters and the disturbance in (4.45). We then provide a more comprehensive comparison by using the same example and varying the bounds on the uncertain parameters, disturbance and the complexity of the RCI set.

Fixed bounds on uncertainty and disturbance

In this subsection, we assume fixed bounds on the uncertain parameters and the disturbance as $|\delta_i| \leq 0.2, i = 1, 2$ and $-0.01 \leq w \leq 0.1$. The **Algorithm 3** has been implemented with CVX [68] by using the solver SeDuMi. The volume of set $\boldsymbol{\mathcal{C}}$ and the vertices of the polytope $\boldsymbol{\Theta}$ and $\boldsymbol{\mathcal{W}}$ have been calculated with [87]. The matrix P has been chosen to define a regular polytope of desired complexity by assuming $W = I$. The maximum volume RCI set and the associated state feedback gain matrices are computed as

$$P_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ \begin{bmatrix} W_2 \\ \hline K_2 \end{bmatrix} = \begin{bmatrix} 5.1435 & -1.8495 \\ 0.6220 & 2.6267 \\ \hline -0.1673 & -0.7539 \end{bmatrix}, \tag{4.47}$$

$$P_3 = \begin{bmatrix} 20 & 20 \\ -20 & 0 \\ 0 & -25 \end{bmatrix},\ \begin{bmatrix} W_3 \\ \hline K_3 \end{bmatrix} = \begin{bmatrix} 109.066 & -41.292 \\ 34.4998 & 91.375 \\ \hline -0.1889 & -0.7765 \end{bmatrix}, \tag{4.48}$$

$$P_4 = \begin{bmatrix} -18 & -55 \\ 18 & 55 \\ 55 & -18 \\ 55 & 18 \end{bmatrix},\ \begin{bmatrix} W_4 \\ \hline K_4 \end{bmatrix} = \begin{bmatrix} 346.5940 & -33.7629 \\ -13.0947 & 219.992 \\ \hline -0.3013 & -0.6029 \end{bmatrix}. \tag{4.49}$$

The subscripts in (4.47) indicate the set complexity n_p. In Fig. 4.1 plots

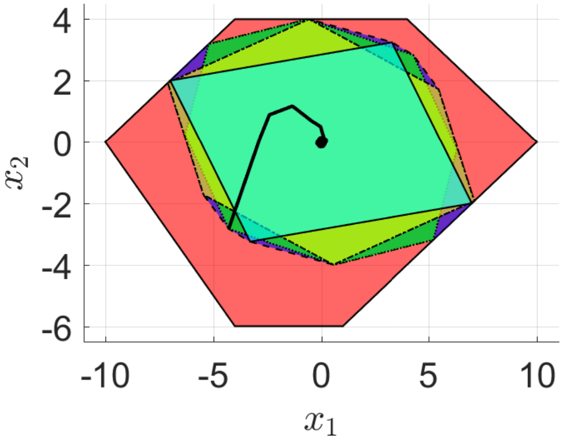

Figure 4.1: Admissible set $\boldsymbol{\mathcal{X}}$ (red) and maximum volume RCI set from **Algorithm 3** with $n_p = 2$ (cyan), $n_p = 3$ (yellow), $n_p = 4$ (green) and convex hull of RCI sets (blue)

are shown for the set of admissible states $\boldsymbol{\mathcal{X}}$ in (4.3), maximum volume RCI set with complexity $n_p = 2$ (volume=58.6), $n_p = 3$ (volume=72.8), $n_p = 4$ (volume=75.5) and convex hull of all RCI sets (volume=82.1). A state trajectory is plotted in Fig. 4.1 (in black) obtained by randomly varying the input disturbance and the uncertain parameters within the chosen bounds.

Comparison of Different Complexity RCI Sets for Varying Uncertainty and Disturbance Bounds

Fig. 4.2 shows the volume of the RCI set, with varying bounds on the uncertain parameters and disturbance, for $n_p = 2,\ 3,\ 4$. Not surprisingly, it can be observed that as n_p increases, feasible solutions are obtained for larger bounds on parameters uncertainty and disturbance. Hence, n_p can be used as an additional tuning parameter to obtain a RCI with larger volume.

4.6.2 LTI System

We consider a double integrator system (see, *Example 1*,[80]) to compare **Algorithm 3** against the approach in [80]. The resulting RCI sets and the

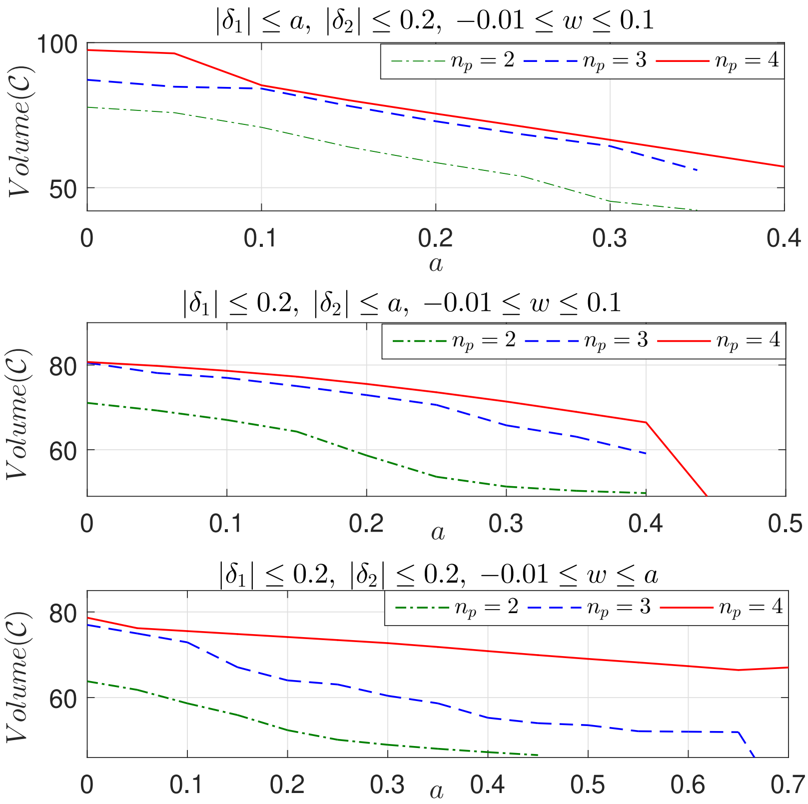

Figure 4.2: $Volume(\mathcal{C})$ obtained by varying n_p and bounds on δ_1, δ_2, w

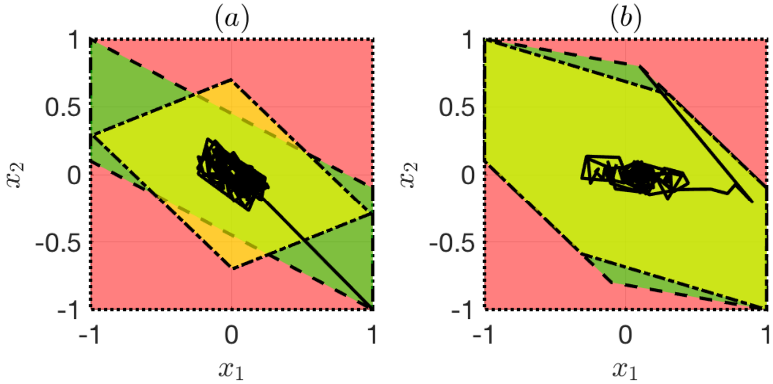

Figure 4.3: In (a) and (b) are the RCI sets $\mathcal{C}$ with complexity $n_p = 2$ and $n_p = 3$ obtained from **Algorithm 3** (green, dashed) and the approach in [80] (yellow, dashed-dot) with admissible set $\mathcal{X}$ (red, dotted).

associated feedback laws obtained using **Algorithm 3** are

$$P_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; \left[\frac{W_2}{K_2}\right] = \left[\begin{array}{cc} 1.0000 & 0.0000 \\ -0.5500 & 0.4500 \\ \hline -0.8630 & -1.7407 \end{array}\right], \tag{4.50}$$

$$P_3 = \begin{bmatrix} 10 & 10 \\ 10 & 0 \\ 1 & 11 \end{bmatrix}, \; \left[\frac{W_3}{K_3}\right] = \left[\begin{array}{cc} 10.0000 & 0.0000 \\ -1.0000 & 9.0000 \\ \hline -0.2251 & -1.2219 \end{array}\right]. \tag{4.51}$$

Fig. 4.3 shows the RCI sets obtained by using the two algorithms and it can be seen that **Algorithm 3** generates relatively larger RCI sets than the approach in [80], which indicate that **Algorithm 3** is indeed less conservative, for this example.

4.6.3 High Dimensional LTI System

We next consider a problem from [42], in which the temperature profile of a one-dimensional bar is controlled. The temperature profile is controlled using three inputs, which are temperatures of the bar at three locations. The discrete-time system is given by

$$x(k+1) = Ax(k) + Bu(k), \tag{4.52}$$

where , $\left[\begin{array}{c|c} A & B \end{array}\right] =$

$$\left[\begin{array}{rrrrrr|rrr} 0.858 & -0.053 & 0.021 & 0.193 & -0.029 & 0.027 & -0.438 & -1.607 & -0.474 \\ -0.053 & 0.716 & 0.039 & 0.051 & 0.179 & 0.162 & -0.074 & -1.479 & 1.389 \\ 0.021 & 0.039 & 0.698 & 0.031 & 0.135 & -0.206 & -1.647 & 1.011 & 0.204 \\ 0.193 & 0.051 & 0.031 & 0.567 & 0.053 & -0.014 & 0.866 & 1.869 & 0.821 \\ -0.029 & 0.179 & 0.135 & 0.053 & 0.658 & -0.031 & 0.633 & 0.004 & -1.364 \\ 0.027 & 0.162 & -0.206 & -0.014 & -0.031 & 0.504 & -0.997 & 1.185 & -0.554 \end{array}\right].$$

subject to the following input constraints

$$|u| \leq [1\ 1\ 1]^T. \tag{4.53}$$

To avoid numerical issues, we additionally consider state constraints $|x_i| \leq 100,\ i = 1, \ldots, 6$, when the proposed algorithm are employed. We compactly summarize the comparisons[2] in the Table 4.1. We illustrate the advantage of

Approach	$\lvert\mathsf{det}(W)\rvert$	Volume
From [46]	7.056×10^4	-
From [42]	1.011×10^6	-
Algorithm 1($\boldsymbol{\mathcal{C}}_6$)	1.594×10^{10}	1.020×10^{12}
Algorithm 3($\boldsymbol{\mathcal{C}}_8$)	-	2.168×10^{12}
Geometric Approach [29]($\boldsymbol{\mathcal{S}}_\infty$)	-	3.299×10^7
Algorithm 3 ($\boldsymbol{\mathcal{C}}_\gamma$)	1.013×10^7	6.486×10^8

Table 4.1: Comparison of Algorithms.

having the flexibility to select the complexity of the RCI set, by noting the volumes of the set $\boldsymbol{\mathcal{C}}_6$ and $\boldsymbol{\mathcal{C}}_8$. The sets $\boldsymbol{\mathcal{C}}_6$ and $\boldsymbol{\mathcal{C}}_8$ are RCI set with complexity $n_p = 6$ and $n_p = 8$, respectively. We next compare sets where performance can be guaranteed. Towards this, we first compute an RCI set $\boldsymbol{\mathcal{S}}_\infty$ using the geometric approach while having an LQR controller with $Q = I$ and $R = I$ in closed-loop. The RCI set $\boldsymbol{\mathcal{C}}_\gamma$ is then computed using **Algorithm** 3, with the

[2]**Note:**$\lvert\mathsf{det}(W)\rvert \propto$ Volume

matrix

$$C = \begin{bmatrix} Q^{1/2} \\ 0 \end{bmatrix}, \; D = \begin{bmatrix} 0 \\ R^{1/2} \end{bmatrix} \tag{4.54}$$

in (4.2). To keep the comparison fair, the performance bound in (4.11) is set to $\gamma = 4444$, which was the maximum LQR cost observed in the set $\boldsymbol{\mathcal{S}}_\infty$. In the Table 4.1, we can observe that the proposed algorithm is able to generate a much larger set with in which the performance can be guaranteed.

CHAPTER 5

Full Complexity RCI sets with Piecewise Affine Inputs

5.1 Introduction

As seen in the preceding chapter, increasing the candidate set complexity allows us to compute RCI sets of larger volume. The algorithms computing RCI sets assumed the invariance inducing controller to be linear state-feedback and computed by the algorithm while optimizing the RCI set volume. Using linear state-feedback for invariance can be advantageous for the applications where the constrainted control of the dynamical system is the objective, and computational resource are limited. Nonetheless, it was proven in [88] that even for simple LTI systems, assigning a linear state feedback controller to an already known polytopic RCI set is not always feasible. In fact, [88] showed that a piece-wise affine controller can always be assigned. Thus, if the ultimate goal is to compute desirably large or small volume RCI sets, a linear state feedback law is not suitable for inducing invariance. Therefore in this chapter, we will focus on directly computing optimized polytopic RCI sets of predefined complexity without imposing any restrictive feedback structure. The results presented in this chapter can be found in [57].

5.2 Brief Literature Survey and Comparison

There are a few approaches in the literature that directly computes RCI sets without restricting control inputs to be in a specific state feedback form. A well-known method is a geometric approach [29], which can directly generate RCI sets. In each iteration of the approach, numerical procedures like Minkowski sum, polytopic projection and minimal set representation calculation are performed, which can be computationally demanding. For polytopic systems, the computational complexity can further grow exponentially with each additional system vertex and dimension [1]. Thus, the geometric approach becomes unusable even for small dimensional uncertain systems [30], [31]. Another method that can be used for the purpose is [83], which solve a particle swarm optimization (PSO) problem to generate RCI set. Since it solves a highly nonconvex problem, the resultant set could be very conservative, e.g., see Example 3.6.2.

In this chapter, we directly compute the RCI sets without imposing any structural restrictions on the control input so that possibly larger RCI sets are obtained. Similar to previous chapters, we consider rationally parameter-dependent systems. The algorithm generates RCI sets of monotonically increasing volume at each iteration until convergence. Hence, it can be terminated at any iteration, unlike the geometric approach which needs a termination condition to be satisfied to obtain a maximal RCI set eventually. Further, by applying the procedure in [21], [89], [90], we show that solutions obtained from the proposed algorithm can be used to construct a piece-wise affine controller for the simple constrained control of the system.

5.3 Problem Formulation

5.3.1 System and Constraints

We consider a discrete-time uncertain system described by

$$x^+ = \mathcal{A}(\Delta)x + \mathcal{B}(\Delta)u + \mathcal{E}(\Delta)w, \tag{5.1}$$

where $x \in \mathbb{R}^{n_x}$ is the current state vector, x^+ is the successor state vector, and $u \in \mathbb{R}^{n_u}$ and $w \in \mathbb{R}^{n_w}$ are the control input and the (additive) disturbance vectors respectively. Furthermore, $\mathcal{A}(\Delta)$, $\mathcal{B}(\Delta)$ and $\mathcal{E}(\Delta)$ are the rationally parameter dependent system matrices expressed as an LFT of the form (3.4). Δ is an uncertain (and possibly) time-varying parameter matrix that satisfies (3.2) and (3.3).

The system is subject to the following polytopic state and input constraints, respectively:

$$\begin{aligned} \boldsymbol{\mathcal{X}} &= \{x : Hx \leq \mathbf{1}\}, \\ \boldsymbol{\mathcal{U}} &= \{u : Gu \leq \mathbf{1}\}. \end{aligned} \tag{5.2}$$

Here $H \in \mathbb{R}^{n_h \times n_x}$ and $G \in \mathbb{R}^{n_g \times n_u}$ are given matrices and $\mathbf{1}$ represents the vector of ones of compatible dimension. We assume $w \in \boldsymbol{\mathcal{W}}$, with $\boldsymbol{\mathcal{W}}$ as a C-set and represented by the convex-hull of a set of finitely many known vertices $\boldsymbol{\mathcal{W}}_v = \left\{w^1, w^2, \cdots, w^\gamma\right\}$:

$$\boldsymbol{\mathcal{W}} = \mathsf{conv}(\boldsymbol{\mathcal{W}}_v). \tag{5.3}$$

We allow the sets $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{W}}$ to be non-symmetric.

5.3.2 Candidate RCI set

As in the earlier chapters, the goal is to compute a 0-symmetric RCI set with a predefined complexity n_p described as

$$\boldsymbol{\mathcal{C}} = \left\{x \in \mathbb{R}^{n_x} : -\mathbf{1} \leq PW^{-1}x \leq \mathbf{1}\right\}, \tag{5.4}$$

where $P \in \mathbb{R}^{n_p \times n_x}$ and $W \in \mathbb{R}^{n_x \times n_x}$. We assume that W is invertible, which would be later guaranteed by the LMI conditions for invariance.

A set $\boldsymbol{\mathcal{C}} \subseteq \boldsymbol{\mathcal{X}}$ is RCI if for each x

$$x \in \boldsymbol{\mathcal{C}} \;\Rightarrow\; \exists u \in \boldsymbol{\mathcal{U}} :\; x^+ \in \boldsymbol{\mathcal{C}},\; \forall (w, \Delta) \in (\boldsymbol{\mathcal{W}}, \boldsymbol{\Delta}). \tag{5.5}$$

Notice that we do not impose any structure on the control input, as we did in the earlier chapters. The problem of finding an RCI set can now be formulated as follows:

Problem 3: Given a matrix P and the discrete-time system (5.1) subject to constraints (5.2) and (5.3) with given $\boldsymbol{\Delta}$, H and G, find a matrix W such that

1. the controlled system in (5.1) satisfies (5.5);
2. the set $\boldsymbol{\mathcal{C}}$ in (5.4) must satisfy $\boldsymbol{\mathcal{C}} \subseteq \boldsymbol{\mathcal{X}}$.

Similar to Problem 2, the representational complexity of the set $\boldsymbol{\mathcal{C}}$ (*i.e.* n_p) is decided by the choice of the matrix P. For $P = I$, the candidate RCI set $\boldsymbol{\mathcal{C}}$ has a complexity similar to the one considered in [42], [47], [56]. Thus, in the next section, we derive tractable feasibility conditions for the solvability of **Problem 3**. The obtained feasibility conditions will be then utilized to develop an iterative algorithm to compute RCI sets with potentially increased volume at each step.

5.4 Sufficient Conditions for Invariance

Let us first introduce a state transformation

$$\theta \triangleq W^{-1}x \Leftrightarrow x = W\theta. \tag{5.6}$$

Using (5.6), the controlled system (5.1) can be expressed in the transformed space as

$$\theta^+ = \underbrace{W^{-1}\mathcal{A}(\Delta)W}_{\bar{\mathcal{A}}(W,\Delta)}\theta + \underbrace{W^{-1}\mathcal{B}(\Delta)}_{\bar{\mathcal{B}}(W,\Delta)}u + \underbrace{W^{-1}\mathcal{E}(\Delta)}_{\bar{\mathcal{E}}(W,\Delta)}w. \tag{5.7}$$

For notational simplicity, we will suppress the arguments of the matrices $\mathcal{A}(\Delta)$, $\mathcal{B}(\Delta)$, $\mathcal{E}(\Delta)$, $\bar{\mathcal{A}}(W,\Delta)$, $\bar{\mathcal{B}}(W,\Delta)$ and $\bar{\mathcal{E}}(W,\Delta)$ in the later parts. Using (5.6), we can also express the set $\boldsymbol{\mathcal{C}}$ as

$$\boldsymbol{\mathcal{C}} = \{W\theta \in \mathbb{R}^{n_x} : \theta \in \boldsymbol{\Theta}\}, \tag{5.8}$$

where $\boldsymbol{\Theta}$ is a symmetric set defined as follows:

$$\boldsymbol{\Theta} \triangleq \{\theta \in \mathbb{R}^{n_x} : -\mathbf{1} \leq P\theta \leq \mathbf{1}\}. \tag{5.9}$$

We have thus introduced a θ-state-space in which the candidate RCI set is a known 0-symmetric set identified as in (5.9). The RCI set $\boldsymbol{\mathcal{C}}$ in the x-state-space will be determined as in (5.8) based on the choice of W.

According to the (5.5), for invariance of $\boldsymbol{\Theta}$, we have to show that for each $\theta \in \boldsymbol{\Theta}$, there exists a $u \in \boldsymbol{\mathcal{U}}$ for which $\theta^+ \in \boldsymbol{\Theta}$, $\forall (w, \Delta) \in (\boldsymbol{\mathcal{W}}, \boldsymbol{\Delta})$. Verifying the existence of u for each $\theta \in \boldsymbol{\Theta}$ for invariance is an intractable problem. To obtain a tractable formulation, we first note that the set $\boldsymbol{\Theta}$ can always be expressed as the convex hull of finitely many (known) vertices. The vertices (and thus the number of them) are determined by the choice of the matrix P. With the set of vertices represented as $\mathbf{\Theta}_v = \{\theta^1, \ldots, \theta^{2\sigma}\}$, we can write

$$\mathbf{\Theta} = \mathsf{conv}\left(\mathbf{\Theta}_v\right). \tag{5.10}$$

For now, we ignore the state constraints on θ obtained by using (5.2) and (5.6), and propose a necessary and sufficient condition for the invariance of the set $\mathbf{\Theta}$.

Lemma 10: *Consider following statements:*

i. $\mathbf{\Theta}$ *in (5.9) is an RCI set for the system (5.7), i.e. for each* $\theta \in \mathbf{\Theta}$,

$$\exists u \in \boldsymbol{\mathcal{U}} : \bar{\mathcal{A}}\theta + \bar{\mathcal{B}}u + \bar{\mathcal{E}}w \in \mathbf{\Theta}, \forall (w, \Delta) \in (\boldsymbol{\mathcal{W}}, \mathbf{\Delta}). \tag{5.11}$$

ii. for each $\theta^j \in \mathbf{\Theta}_v$,

$$\exists u^j \in \boldsymbol{\mathcal{U}} : \bar{\mathcal{A}}\theta^j + \bar{\mathcal{B}}u^j + \bar{\mathcal{E}}w \in \mathbf{\Theta}, \forall (w, \Delta) \in (\boldsymbol{\mathcal{W}}, \mathbf{\Delta}). \tag{5.12}$$

The above two statements are equivalent.

Proof. It can be clearly seen that ($i \Rightarrow ii$) since $\theta^j \in \mathbf{\Theta}$. To prove the converse statement (i.e. $ii \Rightarrow i$), we first assume the existence of u^j as in (5.12) and aim to construct u for any given $\theta \in \mathbf{\Theta}$ such that the invariance condition expressed in (5.11) is satisfied. As implied by the notation, the idea would be to compute the control input as a convex combination of u^js. This convex combination is identified by first expressing θ in terms of the extreme points θ^j as

$$\theta = \sum_{j=1}^{2\sigma} \alpha_j \theta^j, \text{ where } \sum_{j=1}^{2\sigma} \alpha_j = 1, \alpha_j \in [0, 1]. \tag{5.13}$$

By using the α^js identified from this decomposition, the control input is then constructed as

$$u = \sum_{j=1}^{2\sigma} \alpha_j u^j \in \boldsymbol{\mathcal{U}}. \tag{5.14}$$

With this input, the transformed state vector in the next step would be obtained based on (5.7) as

$$\theta^+ = \bar{\mathcal{A}} \underbrace{\left(\sum_{j=1}^{2\sigma} \alpha_j \theta^j \right)}_{\theta} + \bar{\mathcal{B}} \underbrace{\left(\sum_{j=1}^{2\sigma} \alpha_j u^j \right)}_{u} + \bar{\mathcal{E}} \underbrace{\left(\sum_{j=1}^{2\sigma} \alpha_j \right)}_{1} w \tag{5.15}$$

$$= \sum_{j=1}^{2\sigma} \alpha_j \underbrace{(\bar{\mathcal{A}}\theta^j + \bar{\mathcal{B}}u^j + \bar{\mathcal{E}}w)}_{y(w,\Delta)}. \tag{5.16}$$

We know from (5.12) that $y(w, \Delta) \in \mathbf{\Theta}, \forall (w, \Delta) \in (\boldsymbol{\mathcal{W}}, \mathbf{\Delta})$. Since θ^+ is obtained as a convex combination of $y(w, \Delta) \in \mathbf{\Theta}$ and as $\mathbf{\Theta}$ is a convex set, it then necessarily follows that $\theta^+ \in \mathbf{\Theta}, \forall (w, \Delta) \in (\boldsymbol{\mathcal{W}}, \mathbf{\Delta})$. □

Similar results were originally proved [89] and extended to uncertain systems in [90]. **Lemma 10** shows that for invariance of the set $\mathbf{\Theta}$, existence of $u \in \boldsymbol{\mathcal{U}}$ needs to be verified only for the finite set of points $\theta^j \in \mathbf{\Theta}_v$. This is a crucial observation that paves the way toward a tractable solution of **Problem 3**.

We next derive matrix inequality conditions for the invariance of the set $\mathbf{\Theta}$ based on (5.12). For each $\theta^j \in \mathbf{\Theta}_v$, we can see that condition (5.12) can be rewritten as

$$\exists u^j \in \boldsymbol{\mathcal{U}} : -\mathbf{1} \leq P(\bar{\mathcal{A}}\theta^j + \bar{\mathcal{B}}u^j + \bar{\mathcal{E}}w) \leq \mathbf{1}, \forall (w, \Delta) \in (\boldsymbol{\mathcal{W}}, \mathbf{\Delta}). \tag{5.17}$$

Condition (5.17) must be satisfied element-wise. Using (5.7), we hence express (5.17) equivalently as

$$\exists u^j \in \boldsymbol{\mathcal{U}} \text{ and } \phi_{j,i} \in \mathbb{R}_+ : \phi_{j,i}(1 - (e_i^T P W^{-1}(\mathcal{A}W\theta^j + \mathcal{B}u^j + \mathcal{E}w))^2) \geq 0,$$
$$\forall (w, \Delta) \in (\boldsymbol{\mathcal{W}}, \mathbf{\Delta});\ i = 1, \ldots, n_p, \tag{5.18}$$

where e_i represents the i-th column of the identity matrix of size $n_p \times n_p$ and $\phi_{j,i}$s are introduced for the convenience of our derivations in the sequel.

Condition (5.18) is a nonlinear parameter-dependent necessary and sufficient condition for the invariance of the set $\mathbf{\Theta}$, which has to be satisfied for each $\theta^j \in \mathbf{\Theta}_v$.

Having obtained the condition for invariance of the set $\mathbf{\Theta}$, we now proceed to present the main result of this chapter, which gives feasibility conditions for **Problem 3**. In order to deal with uncertain parameter dependence, we employ a specific version of the full-block S-procedure Lemma 5. As ingredients that emerge through this procedure, we introduce a set of multiplier matrices (associated with $\boldsymbol{\Delta}$) and an inner approximation thereof respectively as

$$\boldsymbol{\mathcal{M}} \triangleq \{M : (2.20)\} \text{ and } \boldsymbol{\mathcal{M}}_{pol} \triangleq \{M : (2.27)\}. \tag{5.19}$$

Note that $\boldsymbol{\mathcal{M}}_{pol} \subseteq \boldsymbol{\mathcal{M}}$ is characterized by LMI conditions (as inherited from [78]) and is hence used to express the theorem statement.

Theorem 5: ***Problem 3*** *is feasible if there exist* $u^j \in \mathbb{R}^{n_u}$, $V_{j,i} = V_{j,i}^T \in \mathbb{R}^{n_x \times n_x}$, $\phi_{j,i} \in \mathbb{R}_+$, $M_{j,i} \in \boldsymbol{\mathcal{M}}_{pol}$; $i = 1, \ldots, n_p$; $j = 1, \ldots, 2\sigma$ *and* $W \in \mathbb{R}^{n_x \times n_x}$ *with which (5.20), (5.21) and (5.22) are satisfied for* $i = 1, .., n_p$, $j = 1, \ldots, 2\sigma$ *and* $k = 1, \ldots, \gamma$. *A robust control invariant set can then be obtained as in (5.4).*

$$HW\theta^j \leq \mathbf{1},\, Gu^j \leq \mathbf{1}, \tag{5.20}$$

$$\begin{bmatrix} W^T V_{j,i}^{-1} W & * \\ \phi_{j,i} e_i^T P & \phi_{j,i} \end{bmatrix} \succ 0, \tag{5.21}$$

$$\begin{bmatrix} \phi_{j,i} & * & * \\ AW\theta^j + Bu^j + Ew^k & V_{j,i} + B_p R_{j,i} B_p^T & * \\ A_d W\theta^j + B_d u^j + E_d w^k & S_{j,i} B_p^T + D_p R_{j,i}^T B_p^T & \Phi_{j,i} \end{bmatrix} \succeq 0. \tag{5.22}$$

where, $\Phi_{j,i} = Q_{j,i} + S_{j,i} D_p^T + D_p S_{j,i}^T + D_p R_{j,i} D_p^T$.

Proof. Recalling **Lemma 10**, we aim at finding $u^j \in \boldsymbol{\mathcal{U}}; j = 1, \ldots, 2\sigma$ that satisfy (5.12). In this fashion we will have established that $\boldsymbol{\Theta}$ in (5.9), thus $\boldsymbol{\mathcal{C}}$ in (5.8) are RCI sets. We hence assume that the control input is formed as in (5.14) and observe in reference to (5.6) that the state and control input constraints in (5.2) read as (5.20).

With the intention to resolve the nonlinearity (5.18), we now introduce a positive-definite matrix variable $V_{j,i}$ that satisfies

$$V_{j,i}^{-1} - \phi_{j,i} W^{-T} P^T e_i e_i^T P W^{-1} \succ 0. \tag{5.23}$$

Note that this condition can be expressed equivalently as in (5.21) by applying first a congruence transformation with W and then the Schur complement lemma. With $V_{j,i}$ satisfying (5.23), a sufficient condition for (5.18) can be formulated as

$$\phi_{j,i} - (\mathcal{A}W\theta^j + \mathcal{B}u^j + \mathcal{E}w)^T V_{j,i}^{-1}(\mathcal{A}W\theta^j + \mathcal{B}u^j + \mathcal{E}w) \geq 0. \tag{5.24}$$

This reads after an application of the Schur complement lemma as the parameter-dependent LMI

$$\begin{bmatrix} \phi_{j,i} & * \\ \mathcal{A}(\Delta)W\theta^j + \mathcal{B}(\Delta)u^j + \mathcal{E}(\Delta)w & V_{j,i} \end{bmatrix} \succ 0. \tag{5.25}$$

By now applying **Lemma 5** with

$$\mathcal{Y} = \left[\begin{array}{c|cc} D_p & A_dW\theta^j + B_du^j + E_dw & 0 \\ \hline 0 & \frac{1}{2}\phi_{j,i} & 0 \\ B_p & AW\theta^j + Bu^j + Ew & \frac{1}{2}V_{j,i} \end{array}\right], \tag{5.26}$$

we arrive at an LMI condition in terms of $M_{j,i} \in \boldsymbol{\mathcal{M}}$, which needs to be satisfied for all $w \in \boldsymbol{\mathcal{W}}$. Thanks to the fact that dependence on w is affine and $\boldsymbol{\mathcal{W}}$ in (5.3) is polytopic, we arrive at the condition in (5.22) expressed in terms of the vertices v^j. □

It should be noted at this point that $(1,1)$ block of (5.21) has nonlinear dependence on $(W, V_{j,i})$. This issue was also faced in Chapter 3 and Chapter 4, and resolved by adapting a successive linearization approach. We follow similar lines to develop an iterative scheme for volume optimization in the next section.

5.5 Iterative RCI Set Computation

In this section, we first briefly recall the linearization approach from Section 3.5.1 and Lemma 4. We then formulate a cost based on which successive volume optimization can be performed. Towards this, we first introduce the term $Y_{j,i}$, which is updated at each iteration, and at the $n+1$-st step defined as

$$Y_{j,i} = Y_{j,i}^n \triangleq (V_{j,i}^n)^{-1}W^n, \tag{5.27}$$

where $(W^n, V_{j,i}^n)$ are the values of $(W, V_{j,i})$ obtained at the n-th step. Hence, the sufficient LMI condition for (5.21) to be used at the $n+1$-st step of our iterative scheme is

$$\begin{bmatrix} (Y_{j,i}^n)^T W + W^T Y_{j,i}^n - (Y_{j,i}^n)^T V_{j,i} Y_{j,i}^n & e_i P^T \phi_{j,i} \\ \phi_{j,i} e_i^T P & \phi_{j,i} \end{bmatrix} \succ 0. \tag{5.28}$$

It has been already shown in the previous chapters that such an iterative scheme reduces the conservatism. Note that (5.28) still holds if $W = W^n$ and $V_{j,i} = V_{j,i}^n$, which confirms that the solutions obtained at the n-th step are also feasible at the $n+1$-st step. We also recall that the invertibility of matrix W is implied from the $(1,1)$ block of (5.28).

5.5.1 Iterative Volume Maximization

We now develop an iterative scheme based on **Theorem 5** for the computation of RCI sets of potentially increasing volumes at each step. In the previous chapter, we used an indirect approach to maximize (or minimize) the volume of the RCI set by iteratively maximizing the volume of the ellipsoidal set enclosed therein. In contrast, we now present a direct approach in which the volume of the actual RCI set is aimed to be optimized.

5.5.2 Volume Computation

As the first major step towards developing our algorithm, we show that the volume of the set $\mathcal{C}$ is proportional to $|\mathsf{det}(W)|$ once P is fixed. For this, we use the fact that any 0-symmetric polytopic set $\mathcal{C}$ can be decomposed into n_x-simplices $\mathcal{C}_m,\ m = 1, \ldots, 2\mu$, which are identified in terms of their vertex points x_m^i as

$$\mathcal{C}_m = \left\{ \sum_{i=0}^{n_x} \alpha_i x_m^i : \sum_{i=0}^{n_x} \alpha_i = 1, \alpha_i \in [0,1] \right\}. \tag{5.29}$$

These simplices would have the following properties:

i. $\mathcal{C}_m$ is non-empty;

ii. $\mathsf{int}(\mathcal{C}_{m_1} \cap \mathcal{C}_{m_2}) = \varnothing$, if $m_1 \neq m_2$;

iii. $\bigcup_{m=1}^{2\mu} \mathcal{C}_m = \mathcal{C}$.

Each $\boldsymbol{\mathcal{C}}_m$ would have origin as a common vertex point (i.e. $x_m^0 = 0, \forall m = 1, \dots 2\mu$). The remaining n_x vertices of $\boldsymbol{\mathcal{C}}_m$ can be represented as the columns of a matrix

$$X_m = \begin{bmatrix} x_m^1 & x_m^2 & \cdots & x_m^{n_x} \end{bmatrix}. \tag{5.30}$$

From [91], it is known that the volume of the simplex $\boldsymbol{\mathcal{C}}_m$ is given by the Cayley-Menger determinant:

$$\mathsf{Volume}(\boldsymbol{\mathcal{C}}_m) = \frac{1}{n_x!} |\mathsf{det}(X_m)|. \tag{5.31}$$

Let the matrix Θ_m be defined as

$$\Theta_m = W^{-1} X_m \Leftrightarrow X_m = W \Theta_m. \tag{5.32}$$

From (5.6), we know that the columns of the matrix Θ_m represent n_x independent vertices from the set $\boldsymbol{\Theta}_v$.

Lemma 11: *The volume of the set $\boldsymbol{\mathcal{C}}$ identified as in (5.4) with a fixed P is proportional to $|\mathsf{det}(W)|$.*

Proof. Since the simplices $\boldsymbol{\mathcal{C}}_m$ have disjoint interiors, the volume of the set $\boldsymbol{\mathcal{C}}$ can be written as

$$\mathsf{Volume}(\boldsymbol{\mathcal{C}}) = \sum_{m=1}^{2\mu} \mathsf{Volume}(\boldsymbol{\mathcal{C}}_m). \tag{5.33}$$

It then follows from (5.31) and (5.32) that

$$\mathsf{Volume}(\boldsymbol{\mathcal{C}}) = \frac{1}{n_x!} \sum_{m=1}^{2\mu} |\mathsf{det}(X_m)| = \frac{1}{n_x!} \sum_{m=1}^{2\mu} |\mathsf{det}(W\Theta_m)|. \tag{5.34}$$

Since the set $\boldsymbol{\mathcal{C}}$ is symmetric, we can always order Θ_ms as

$$\Theta_{m+\mu} = -\Theta_m, m = 1, \dots, \mu. \tag{5.35}$$

With $|\mathsf{det}(\Theta_m)| = |\mathsf{det}(-\Theta_m)|$, the summation in (5.34) can be rewritten as

$$\mathsf{Volume}(\boldsymbol{\mathcal{C}}) = \frac{2}{n_x!} |\mathsf{det}(W)| \sum_{m=1}^{\mu} |\mathsf{det}(\Theta_m)|. \tag{5.36}$$

For a fixed P matrix, Θ_ms will be fixed too, which implies that

$$\mathsf{Volume}(\mathcal{C}) \propto |\mathsf{det}(W)|. \tag{5.37}$$

□

Note that the simplicial decomposition of the set $\mathcal{C}$ is not unique, which is though not a problem for our purposes. Once P is fixed, **Lemma 11** basically justifies to maximize $|\mathsf{det}(W)|$ to obtain a set $\mathcal{C}$ of desirably large volume. Simplicial decomposition will also be useful when we consider controller design in the next section.

Iterative Volume Maximization

In order to obtain an RCI set with a desirably large volume, we hence need to solve a determinant maximization problem under LMI conditions presented in **Theorem 5**. Such a problem reads as a generalized semi-definite program only when W is required to be symmetric [81], which would necessarily introduce potential conservatism. An iterative volume maximization approached was developed in Section 3.5.2 without enforcing symmetry on W, which we will also use in this chapter. We introduce a condition, which was formulated in Section 3.5.2 in terms of W^n as

$$W^T W^n + (W^n)^T W - (W^n)^T W^n \succcurlyeq \bar{Z} \succ 0. \tag{5.38}$$

Note that this condition is necessarily satisfied with $W = W^n$. As a result, maximization of $\mathsf{det}(\bar{Z})$ under (5.38) would lead to a solution W^{n+1} that satisfies

$$|\mathsf{det}(W^{n+1})| \geq |\mathsf{det}(W^n)|. \tag{5.39}$$

This allows us to develop the following iterative algorithm to compute RCI sets of increased volume at each step for a priori chosen matrix P:

Algorithm 4: [$n+1$**-st step**]

$$\begin{array}{cc} \max\limits_{W, \bar{Z}, V_{j,i}, M_{j,i}, \phi_{j,i}, u^j} & \mathsf{log\,det}(\bar{Z}) \\ \text{subject to:} & (5.20), (5.22), (5.28) \text{ and } (5.38) \end{array}$$

Initial Optimization to Compute W^0: Condition (5.38) is removed and $\log\det(\bar{Z})$ is changed to $\log\det(W + W^T)$; (5.28) is imposed with $Y^n_{j,i} \to \psi I$, where ψ is selected by a line search to find an initial feasible solution.

The invariance conditions (5.22) and (5.28) are only sufficient and hence might be conservative. By updating the matrix W at each iteration, we are actually scaling and rotating the initial polyhedron to a desirably large volume. It has been already shown before that the solution from step n is feasible at step $n+1$ in (5.28) and (5.38), which ensures that **Algorithm 4** is recursively feasible. Hence the volume of set $\boldsymbol{\mathcal{C}}$ increases iteratively until it converges to some staionary point. The algorithm should be terminated in practice when the change in $|\det(W)|$ is below a certain value or when infeasibility occurs due to numerical reasons.

One might obtain different RCI sets from **Algorithm 4** for different P choices (as well as different initial solutions) and these can in fact be combined to construct larger RCI sets. As is known from [1], [85], the convex hull and union of different RCI sets is a larger RCI set with higher complexity.

Remark 7: In **Algorithm 4**, the SDP consists of $(2\sigma\times n_p\times\Big(\gamma+1+\frac{\eta(\eta+1)}{2}\Big)$ $+1)$ LMIs due to (5.22), (5.28), (5.38); and $(2\sigma \times (n_h + n_g))$ scalar inequalities representing the system constraints in (5.20). Furthermore, these inequalities are in terms of $(2 + 2\sigma \times (1 + 4n_p))$ matrix variables and $(2\sigma \times n_p)$ scalar variables. The complexity due to system and invariance constraints can in fact be reduced significantly in the case when the sets $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{W}}$ are 0-symmetric. Because of the symmetry, the invariance condition needs to be verified only on non-symmetric elements of $\boldsymbol{\Theta}_v$. In this way the number of LMIs and scalar inequalities reduce to $\Big(\sigma \times n_p \times \Big(\gamma + 1 + \frac{\eta(\eta+1)}{2}\Big) + 1\Big)$ and $\sigma \times (n_h + n_g)$ respectively. We then also have less number of matrix and scalar variables as $(2 + \sigma(1 + 4n_p))$ and $(\sigma \times n_p)$ respectively. The computational complexity can be reduced by also considering alternative relaxation schemes or structured multipliers at the cost of potential conservatism. For instance, Pólya relaxation might be replaced with the so-called convex hull relaxation (see e.g. [78]) to reduce the number of LMIs. The number of variables can be reduced by using D-scales, DG scales or block-diagonal sub-blocks in the multiplier matrix M [8].

Algorithm 4 would provide RCI sets for which control inputs that ensure invariance are known to exist. But to realize constrained control, a method needs to be devised to compute a feasible input for any given $x \in \boldsymbol{\mathcal{C}}$. In the next section, we present an offline approach to constrained controller design based on an existing result.

5.6 Controller Design

The purpose of this section is to design a piecewise affine controller based on the approach presented in [89], [90] for the constrained control of the system in (5.1). The approach requires that at each vertex of the set $\mathcal{C}$, there should exist an admissible control action that brings the state to the interior of the set $\mathcal{C}$ within finite time. To be able to design such a controller, we consider using the RCI set computation algorithm developed in the previous section together with following lemma.

Lemma 12 ([1])**:** *A set $\mathcal{C}$ is λ-contractive for the system* (5.1) *if and only if it is controlled-invariant for the modified system*

$$x^{+} = \frac{\mathcal{A}(\Delta)x + \mathcal{B}(\Delta)u + \mathcal{E}(\Delta)w}{\lambda}. \tag{5.40}$$

For a specific $\lambda \in (0, 1)$ choice, we would then obtain an RCI set $\mathcal{C}$ for the modified system in (5.40), which would serve as a λ-contractive set for the original system of (5.1). A controller designed for the modified system of (5.40) would hence serve as a stabilizing piecewise-affine controller for the original system (5.1), thus ensuring asymptotic convergence to origin in the absence of any disturbance input. This controller will also keep the trajectory of (5.1) in $\mathcal{C}$ for any disturbance input within the considered disturbance set.

To this end, we first recall the decomposition of the RCI set $\mathcal{C}$ into n_x-simplicies $\mathcal{C}_m$ as in (5.29). The control law is then formulated as a domain-dependent state feedback of the form

$$x(k) \in \mathcal{C}_m \Rightarrow u(k) = K_m x(k). \tag{5.41}$$

If $x(k)$ happens to be at a point that is common to $\mathcal{C}_i$ and $\mathcal{C}_j$ (e.g. a point on a common edge), then the control input can be computed with either K_i or K_j. To identify the state-feedback gain matrices K_m, we first use the control inputs associated with the vertices x_m^j to form a matrix as

$$U_m = \begin{bmatrix} u_m^1 & u_m^2 & \cdots & u_m^{n_x} \end{bmatrix}. \tag{5.42}$$

The associated state feedback gain matrix is then computed as

$$K_m = U_m X_m^{-1},\ m = 1, \ldots, 2\mu. \tag{5.43}$$

In [89], [90], it is shown that the piece-wise affine control law obtaind in this fashion is stabilizing and is recursively feasible, which indeed imply that it is an invariance-inducing controller. Indeed, if $x(k)$ is expressed as a convex combination of the vertices of $\boldsymbol{\mathcal{C}}_m$, $u(k)$ of (5.41) can then also be expressed as a convex combination of the associated vertices u_m^j, which basically corresponds to the control law proposed in (5.14):

$$x(k) = X_m\alpha \Rightarrow u(k) = K_m X_m \alpha = U_m \alpha. \tag{5.44}$$

Assuming that the system starts from an initial state within the computed RCI set, it is also possible to develop a control algorithm based on online optimization by using the associated W matrix and the vertices θ^j, $j = 1, ..., 2\sigma$[21]. At each time step, a linear program would then be solved to obtain a set of weights, with which the control input would be constructed. With k'th time step indicated with a superscript ^{k}, the weights are obtained and used to construct the control input as follows:

$$\begin{aligned} \beta^k &= \arg\min_{\beta\in\mathbb{R}^{2\sigma}} \left\{ \sum_{j=1}^{2\sigma} \beta_j : \sum_{j=1}^{2\sigma} \beta_j W\theta^j = x(k),\, 0 \le \beta_j \le 1\,, j = 1, \ldots, 2\sigma \right\} \\ u(k) &= \sum_{j=1}^{2\sigma} \beta_j^k u^j. \end{aligned} \tag{5.45}$$

A controller implemented in this fashion would ensure asymptotic stability, as established in detail by [21]. An online approach might be preferable especially when working with high dimensional systems, for which obtaining a simplicial decomposition of the set $\boldsymbol{\mathcal{C}}$ would be difficult.

The closed-loop transient performance of the system can (to some extent) be shaped by the choice of the contraction factor λ, which also affects the size of the set $\boldsymbol{\mathcal{C}}$. However, a known weakness of such a control structure is that a full range of control inputs are utilized only at the boundary of the set $\boldsymbol{\mathcal{C}}$. In the interior of the set $\boldsymbol{\mathcal{C}}$, progressively smaller control inputs are applied as the system trajectories approach origin. Thus it may take a long time for a controller to stabilize the system. It follows from [21] that this weakness could easily be overcome by modifying the proposed controller into a so-called interpolation-based controller, which suboptimally minimizes a predefined cost function.

5.7 Illustrative Examples

We now provide some examples to illustrate the potential of the approach. The algorithm was implemented in Matlab on a 3.1 GHz Intel Core i7-555U macOS computer with 8 GB RAM with YALMIP [67] and the solver SeDuMi [64]. The computation of the volume, vertices and projections of the polytope was done using MPT [87].

5.7.1 Double Integrator

The proposed algorithm is demonstrated with a discrete-time double integrator that has rational parameter dependence. The system dynamics are described as in (5.1) with

$$\mathcal{A} = \begin{bmatrix} 1+\delta_1 & 1+\delta_1 \\ 0 & 1+\frac{\delta_2}{1+\delta_1} \end{bmatrix},\ \mathcal{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathcal{E} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{5.46}$$

where $|\delta_{1,2}| \leq 0.25$ represent the uncertain parameters. The state and control input constraints and the input disturbance bound are expressed as

$$|x| \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix},\ |u| \leq 3,\ |w| \leq 0.6. \tag{5.47}$$

In order to make comparisons with the existing algorithms [42], [47], [49] and the geometric approach, we first introduce a new parameter as $\delta_3 = \delta_2/(1+\delta_1)$ and thus view (5.46) as a system that has affine dependence on (δ_1, δ_3). We illustrate the proposed algorithm by computing the sets $\mathcal{C}_4$ and $\mathcal{C}_5$ associated with $n_p = 4$ and $n_p = 5$ respectively. These sets are compared with the maximal RCI set $\mathbf{\Omega}_\infty$ obtained by using an existing geometric approach, in which the control inputs are also assumed to be free. **FIGURE 5.1**(a) shows the generated RCI sets. We were unable to compute any RCI set with the methods from [42], [47], [49], [55], [56], in which the control input is assumed to have a linear structure and/or a conservative system description is used to account for parametric uncertainty. It can be observed that the maximal RCI set from the geometric approach (Volume=40.2445) is described by 10 hyperplanes and has a slightly larger volume than the one obtained using the proposed approach with the same complexity (Volume=39.9720). The non-symmetric vertices of the set $\mathcal{C}_5$ and the corresponding inputs are given by

$$
\begin{aligned}
X &= \begin{pmatrix} x^1 & x^2 & x^3 & x^4 & x^5 \\ 3.3198 & 5.0000 & 5.0000 & 2.2185 & -1.6636 \\ -3.6425 & -3.4323 & -1.4801 & 1.2336 & 3.2409 \end{pmatrix} \\
U &= \begin{pmatrix} u^1 & u^2 & u^3 & u^4 & u^5 \\ 2.9922 & 2.6635 & -1.1764 & -2.9825 & -3.0000 \end{pmatrix}
\end{aligned} \tag{5.48}
$$

Notice that the vertices $\{x^5, -x^5\} \notin \boldsymbol{\Omega}_\infty$, this is due to the approximation of (5.46) with an affine parameter dependent system while computing $\boldsymbol{\Omega}_\infty$. Lastly, in **FIGURE 5.1**(*b*), we demonstrate the controller proposed in the Section *5.6* in closed-loop with the system (5.46). Also, the proposed decomposition of the set $\mathcal{C}$ into simplices is shown. The closed-loop trajectory, when starting from one of the vertices, can be seen in black (dot-dashed) is produced by randomly varying disturbances and uncertainties within their bounds.

5.7.2 4-D Vehicle Lateral Dynamics

We now compare the proposed algorithm with a recent work [56] using a 4-dimensional bicycle model for vehicle lateral dynamics [56]. We assume the longitudinal velocity of the vehicle to be an uncertain parameter as $V_x \in [50, 70]$ km/h, which enters the system dynamics rationally. For a fair comparison, we keep the complexity of the RCI set same as [56] by selecting $P = I$. Using **Algorithm 4**, the nonsymmetric vertices of the set $\mathcal{C}$ and corresponding control inputs were found to be

$$\begin{bmatrix} X \\ \hline U \end{bmatrix} =$$

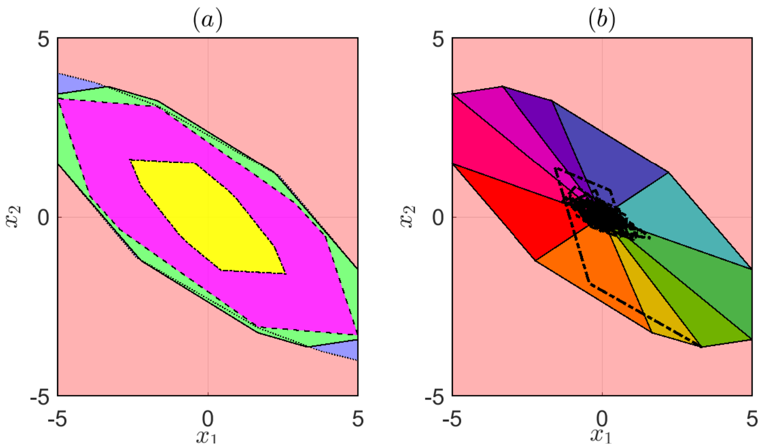

Figure 5.1: (a) Admissible states $\boldsymbol{\mathcal{X}}$ (red) and maximal RCI set $\boldsymbol{\Omega}_\infty$ geometric approach (blue; dotted), and maximum volume RCI set $\boldsymbol{\mathcal{C}}_5$ (green; solid) and $\boldsymbol{\mathcal{C}}_4$ (magenta; dashed) from **Algorithm 4** with complexity n_p=5 and n_p=4, respectively, and minimum volume RCI set using **Remark 3** with n_p=5 (yellow; dot-dashed) and (b) Admissible states $\boldsymbol{\mathcal{X}}$ (red) and Simplex decomposition of the maximum volume RCI set (with $\lambda = 0.99$) from **Algorithm 4** with n_p=5 (colored).

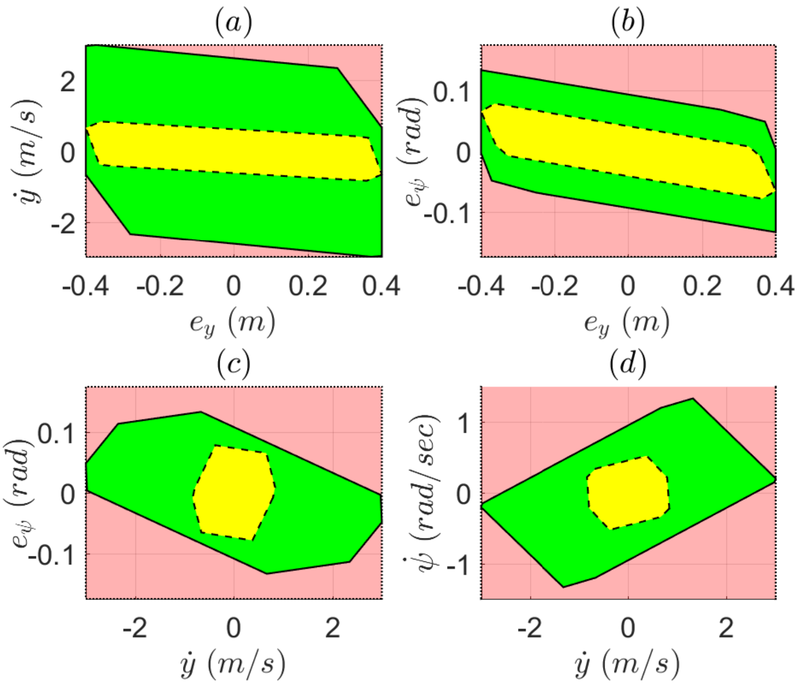

Figure 5.2: (a) Projection of admissible set $\boldsymbol{\mathcal{X}}$ (red; dotted), maximum volume RCI set $\boldsymbol{\mathcal{C}}$ from **Algorithm 4**(green; solid) and using [56] (yellow; dashed).

$$
\left(\begin{array}{cccccccc}
x^1 & x^2 & x^3 & x^4 & x^5 & x^6 & x^7 & x^8 \\
0.3714 & 0.4000 & -0.2800 & -0.2514 & 0.2514 & 0.2800 & -0.4000 & -0.3714 \\
-3.0000 & -2.9754 & -2.3478 & -2.3233 & -1.3182 & -1.2937 & -0.6661 & -0.6415 \\
0.0485 & 0.0035 & 0.1136 & 0.0687 & 0.0681 & 0.0232 & 0.1332 & 0.0883 \\
-0.2040 & -0.1502 & -0.0704 & -0.0165 & -1.3353 & -1.2815 & -1.2017 & -1.1479 \\
\hline
u^1 & u^2 & u^3 & u^4 & u^5 & u^6 & u^7 & u^8 \\
-0.0791 & 0.0776 & -0.0873 & 0.0873 & 0.0517 & 0.0873 & -0.0426 & 0.0873 \\
-0.1235 & 0.1407 & -1.0000 & -0.9999 & 1.0000 & 0.9991 & 0.3675 & -0.9999
\end{array}\right)
\tag{5.49}
$$

The plots in **FIGURE 5.2**-$(a), (b), (c)$ and (d) show the projections of the admissible states $\boldsymbol{\mathcal{X}}$, and the maximum volume RCI set $\boldsymbol{\mathcal{C}}$ obtained from **Algorithm 4** after 3000 iterations, and the algorithm in [56], respectively. The RCI set obtained from the proposed algorithm is much larger than the compared approach. However, larger set comes at the cost of a modestly complex controller. Furthermore, even though the computational complexity is high (see **Remark 7**), each iteration of the proposed algorithm took 24 seconds on average compared to 14 seconds taken by [56]. We also tried to compute the RCI set by using the approximate polytopic model, but were unable to compute any RCI set with the methods from [47], [49]. On the other hand, the geometric approach did not converge even after more than 24 hours.

5.7.3 2-D Toy Example

Lastly, in order to demonstrate the tradeoff between complexity and the volume, we use an example from the [7]. The system dynamics is

$$
x(k+1) = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0.22 \\ 0.20 \end{bmatrix} u(k), \tag{5.50}
$$

and the system constraints are given by

$$
|x(k)| \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \ |u(k)| \leq 2. \tag{5.51}
$$

We compare the λ-contractive sets obtained using the geometric approach [29] and the **Algorithm 4** for $\lambda = 0.9$. All the computed sets are shown in Figure 5.3.

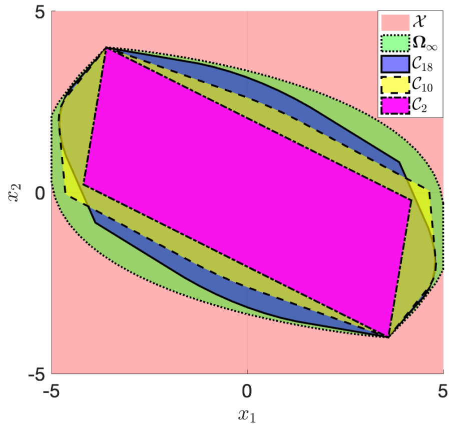

Figure 5.3: Admissible states $\boldsymbol{\mathcal{X}}$ (red), maximal RCI set $\mathbf{\Omega}_\infty$ using geometric approach (green; dotted), maximum volume RCI set from **Algorithm 4** are $\boldsymbol{\mathcal{C}}_{18}$ (blue; solid), $\boldsymbol{\mathcal{C}}_{10}$ (yellow; dashed) and $\boldsymbol{\mathcal{C}}_2$ (magenta; dot-dashed)

Using the geometric approach, we computed $\mathbf{\Omega}_\infty$, which is defined by exactly 250 half spaces and $\mathsf{Volume}(\mathbf{\Omega}_\infty) = 58.1948$. Next, using the proposed **Algorithm 4**, we compute RCI sets $\boldsymbol{\mathcal{C}}_{18}$, $\boldsymbol{\mathcal{C}}_{10}$ and $\boldsymbol{\mathcal{C}}_2$, with different representational complexities $n_p = 18$, $n_p = 10$ and $n_p = 2$, respectively. For $n_p > 18$, no significant improvement was observed in the volume of the computed RCI sets. Since the final set depends of the initial matrix P, there may exist P for which the final RCI set is larger than the one presented in Figure 5.3. The volume of the obtained RCI set are $\mathsf{Volume}(\boldsymbol{\mathcal{C}}_{18}) = 49.5251$, $\mathsf{Volume}(\boldsymbol{\mathcal{C}}_{10}) = 44.0217$ and $\mathsf{Volume}(\boldsymbol{\mathcal{C}}_2) = 31.8994$.

As expected, the sets computed using the proposed algorithm are smaller than $\mathbf{\Omega}_\infty$. Nevertheless, the tradeoff between the complexity and the volume can be a deciding factor for many to use the proposed approach. For example, the set $\boldsymbol{\mathcal{C}}_{18}$ covers 85.1% of the set $\mathbf{\Omega}_\infty$ with a reduction in the representational complexity by 85.6%. Further, we can obtain a larger RCI set by computing the convex hull of the sets $\boldsymbol{\mathcal{C}}_{18}$ and $\boldsymbol{\mathcal{C}}_{10}$. The convex hull is represented by 32 hyperplanes and has a volume of 50.5714.

CHAPTER 6

RCI sets for LPV Systems

6.1 Introduction

As done in the previous chapters and [33], [37], [46], [49], [92], a common practice is to treat the time-varying parameters in the system dynamics as bounded uncertainties, even if these parameters are observable. The invariance inducing control laws are typically assumed to be only state-dependent, without exploiting the observed parameter information. In this way, the obtained RCI sets can be potentially conservative and, in the worst case, even empty. The conservatism can also be understood from the fact that a system which can be stabilized using parameter-dependent feedback laws may not be stabilized robustly [93].

A natural extension of the results presented in the previous chapters is to develop methods to compute RCI sets for the LPV systems that utilize the observed parameter information. Thus, in this chapter, we allow the RCI set and invariance inducing control law to be parameter-dependent. Such sets and the corresponding controllers are termed as parameter-dependent RCI (PD-RCI) sets and parameter-dependent control laws (PDCLs). The advantages of using a PDCL and PD-RCI set are motivated as follows:

- *Parameter-dependent control laws*: PDCL controllers can possibly stabilize LPV systems which may not be robustly stabilizable. Moreover, we treat PDCL as an additional optimization variable in our algorithm to compute an RCI set with desirably large volume, instead of computing it independently. Thus, PDCL is optimal for the constructed RCI set. We remark that a similar construction was proposed in a robust framework in [42], [46], [49], [55], [56].
- *Parameter dependent RCI-sets*: The scheduling parameters affect the system's time evolution, and thus the set of initial states for which invariance can be achieved. Therefore, only considering fixed (or parameter-independent) RCI set description for all scheduling parameters could be restrictive and may lead to conservative (namely, small-volume) sets. This restrictiveness motivates us to allow the RCI set description to be parameter-dependent. Moreover, such a set description also provides a mapping between the initial scheduling parameter and the set of initial states for which invariance can be achieved, which could be useful for analysis.

In this chapter, we will develop an iterative algorithm to compute a PD-RCI set of desirably large volume and the invariance inducing PDCL gain for the LPV systems. The representational complexity of the PD-RCI sets can be predefined. We derive new LMI conditions for invariance by employing Finsler's lemma and Polya's relaxation. These conditions are constructed to ensure invariance for all future (unknown) values of the scheduling parameters. Additionally, we can also compute PD-RCI sets within which desired quadratic performance can be guaranteed using the algorithm. The results presented in this chapter can be found in [58]. Note that, the notations used in this chapter are independent of the previous chapters and will be introduced in the text whenever necessary.

6.2 Problem Formulation

6.2.1 System and Constraints

Let us consider a discrete-time polytopic LPV system described by

$$x(t+1) = \mathcal{A}(\xi(t))x(t) + \mathcal{B}(\xi(t))u(t) + \mathcal{E}(\xi(t))w(t), \tag{6.1}$$

$$z(t) = \mathcal{C}(\xi(t))x(t) + \mathcal{D}(\xi(t))u(t), \tag{6.2}$$

where $t \in \mathbb{Z}_+$is the time index, $x(t) \in \mathbb{R}^{n_x}$ and $z(t) \in \mathbb{R}^{n_z}$ are the current state and the output vectors, $x(t+1)$ is the successor state vector, and $u(t) \in \mathbb{R}^{n_u}$ and $w(t) \in \mathbb{R}^{n_w}$ are the control and the (additive) disturbance input vectors, respectively. The system matrices $\mathcal{A}(\xi(t))$, $\mathcal{B}(\xi(t))$, $\mathcal{C}(\xi(t))$, $\mathcal{D}(\xi(t))$ and $\mathcal{E}(\xi(t))$ (possibly) depend on the time-varying scheduling parameter $\xi(t)$, which takes value in unit simplex,

$$\boldsymbol{\Xi} = \left\{ \xi \in \mathbb{R}^{N_\xi} : \sum_{k=1}^{N_\xi} \xi_k = 1, \quad \xi_k \geq 0 \right\}. \tag{6.3}$$

It is assumed that the current value of $\xi(t)$ is always available. The polytopic system matrices are given by

$$\begin{bmatrix} \mathcal{A}(\xi(t)) & \mathcal{B}(\xi(t)) & \mathcal{E}(\xi(t)) \\ \mathcal{C}(\xi(t)) & \mathcal{D}(\xi(t)) & 0 \end{bmatrix} = \sum_{k=1}^{N_\xi} \xi_k(t) \begin{bmatrix} A^k & B^k & E^k \\ C^k & D^k & 0 \end{bmatrix}, \tag{6.4}$$

where A^k, B^k, C^k, D^k, E^k are real matrices of compatible dimensions. The system is subjected to the following polytopic state and input constraints, and bounded disturbance:

$$\begin{aligned} \boldsymbol{\mathcal{X}}_u &= \left\{ (x, u) : [H_x \; H_u] \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \leq \mathbf{1} \right\}, \\ \boldsymbol{\mathcal{W}} &= \{ w : -\mathbf{1} \leq Gw(t) \leq \mathbf{1} \}. \end{aligned} \tag{6.5}$$

where $H_x \in \mathbb{R}^{n_h \times n_x}$, $H_u \in \mathbb{R}^{n_h \times n_u}$ and $G \in \mathbb{R}^{n_g \times n_w}$ are given matrices.

6.2.2 Candidate RCI set and Controller

In this chapter, we want to compute a 0-symmetric PD-RCI set with a pre-defined complexity n_p described as

$$\boldsymbol{\mathcal{S}}(\xi(t)) = \left\{ x \in \mathbb{R}^{n_x} : -\mathbf{1} \leq \mathcal{P}(\xi(t)) W^{-1} x(t) \leq \mathbf{1} \right\}, \tag{6.6}$$

where $\mathcal{P}(\xi(t)) \triangleq \sum_{k=1}^{N_\xi} \xi_k(t) P^k$, $P^k \in \mathbb{R}^{n_p \times n_x}$ and $W \in \mathbb{R}^{n_x \times n_x}$. In order to have a non-empty and bounded set description $\boldsymbol{\mathcal{S}}(\xi(t))$, the matrix W should be invertible and $\mathsf{Rank}(\mathcal{P}(\xi(t))) = n_x$, $\forall \xi \in \boldsymbol{\Xi}$. This will be later guaranteed by proper LMI conditions. Note that, if $P^k = P$ for all $k = 1, \ldots, N_\xi$, then $\mathcal{P}(\xi(t)) = P$, which is similar to the RCI set description considered in Chapters 4-5.

Furthermore, invariance of the set $\boldsymbol{\mathcal{S}}(\xi(t))$ is achieved using a parameter dependent controller, which is to be found and expressed as

$$u(t) = \mathcal{K}(\xi(t))x(t), \tag{6.7}$$

where $\mathcal{K}(\xi(t)) \triangleq \sum_{k=1}^{N_\xi} \xi_k(t)K^k$ and $K^k \in \mathbb{R}^{n_u \times n_x}$. The closed loop representation of the system (6.1) and (6.2) with the controller (6.7) can be written as

$$x(t+1) = \overbrace{(\mathcal{A}(\xi(t)) + \mathcal{B}(\xi(t))\mathcal{K}(\xi(t))}^{\mathcal{A}_{\mathcal{K}}(\xi(t))} x(t) + \mathcal{E}(\xi(t))w(t), \tag{6.8}$$

$$z(t) = \overbrace{(\mathcal{C}(\xi(t)) + \mathcal{D}(\xi(t))\mathcal{K}(\xi(t))}^{\mathcal{C}_{\mathcal{K}}(\xi(t))} x(t). \tag{6.9}$$

To the best of authors' knowledge, there is no related work which computes a described PD-RCI set for the system (6.1). Thus we first formalize the definition of the set by adapting the standard definition of the RCI set to the LPV setting in the sequel.

We say a set $\boldsymbol{\mathcal{S}}(\xi(t))$ is a PD-RCI set if for any given $\xi(t) \in \boldsymbol{\Xi}$ and each $x(t) \in \boldsymbol{\mathcal{S}}(\xi(t))$

$$\mathcal{A}_{\mathcal{K}}(\xi(t))\boldsymbol{\mathcal{S}}(\xi(t)) \oplus \mathcal{E}(\xi(t))\boldsymbol{\mathcal{W}} \subseteq \boldsymbol{\mathcal{S}}(\xi(t+1)), \quad \forall \xi(t+1) \in \boldsymbol{\Xi}, \tag{6.10}$$

$$\boldsymbol{\mathcal{S}}(\xi(t)) \subseteq \boldsymbol{\mathcal{X}}(\xi(t)), \tag{6.11}$$

where the set $\boldsymbol{\mathcal{X}}(\xi(t)) = \{x : (H_x + H_u\mathcal{K}(\xi(t)))x(t) \leq \mathbf{1}\}$. Condition (6.10) should be satisfied for $\forall\xi(t+1) \in \boldsymbol{\Xi}$ since $\xi(t+1)$ is unknown at time t, which also implies $x(t+1) \in \bigcap_{\forall \xi(t+1) \in \boldsymbol{\Xi}} \boldsymbol{\mathcal{S}}(\xi(t+1))$. Moreover, for each $\xi(t) \in \boldsymbol{\Xi}$, the set $\boldsymbol{\mathcal{S}}(\xi(t))$ should also satisfy (6.11) to fulfill the system constraints. If there exists a set $\boldsymbol{\mathcal{S}}(\xi(t))$ and a controller $\mathcal{K}(\xi(t))$ satisfying conditions (6.10) and (6.11) then for some initial $\xi(t) \in \boldsymbol{\Xi}$, we can select any initial $x(t)$ on the corresponding slice of the set $\boldsymbol{\mathcal{S}}(\xi(t))$ to utilize the invariance property. Thus, in case we are also allowed to choose the initial $\xi(t)$, then we can always select the largest slice of the set $\boldsymbol{\mathcal{S}}(\xi(t))$.

6.2.3 Performance Constraints

In some applications (e.g., MPC and IBC), it may be desirable to have guaranteed performance within the PD-RCI sets for the closed loop system (6.8) and (6.9). For this purpose, we consider quadratic performance constraints defined as

$$\sum_{i=0}^{\infty} \|z(t+i)\|_2^2 \leq \gamma,\ 0 \leq \gamma < \infty. \tag{6.12}$$

Note that the performance constraint (6.12) can be only satisfied if $w(t) = 0\,\forall t \geq 0$ (or $w(t)$ eventually becomes zero after a finite time). Thus, in our formulation, we will assume $w(t) = 0$ only when performance constraints are considered.

Our aim here is to compute $\mathcal{P}(\xi(t))$, W and $\mathcal{K}(\xi(t))$, which together define the PD-RCI set (6.6) and the invariance inducing controller (6.7). To avoid solving a highly nonlinear problem and for the convenience of our presentation, we divide this computation into two subproblems. The first subproblem computes W and $\mathcal{K}(\xi(t))$ for given parameter-independent matrix P. Then, in the second subproblem, we optimize w.r.t. the parameter-dependent matrix $\mathcal{P}(\xi(t))$ and we update the controller $\mathcal{K}(\xi(t))$. In the following, the two subproblems are formalized.

Problem 4: For a given matrix $P_{init} \in \mathbb{R}^{n_p \times n_x}$ such that $\mathcal{P}(\xi(t)) = P_{init}$ and the discrete-time system (6.1) subject to constraints (6.5), find a matrix W and the control law $\mathcal{K}(\xi(t))$ that satisfies conditions (6.10), (6.11) and (6.12) for any arbitrary variation of $\xi(t) \in \boldsymbol{\Xi}$.

Problem 5: For a given matrix W and the discrete-time system (6.1) subject to constraints (6.5), find the matrix $\mathcal{P}(\xi(t))$ and the control law $\mathcal{K}(\xi(t))$ that satisfies conditions (6.10), (6.11) and (6.12) for any arbitrary variation of $\xi(t) \in \boldsymbol{\Xi}$.

Observe that by solving **Problem 4** we obtain an RCI set which is independent of the parameter $\xi(t)$, since $\mathcal{P}(\xi(t)) = P_{init}$. In order to obtain a PD-RCI set $\boldsymbol{\mathcal{S}}(\xi(t))$, we solve **Problem 4** and **Problem 5** sequentially. In both problems, conditions (6.12) is optional and only imposed if performance is desired. Even though we present our formulation in the form of feasibility problems, our final goal is to design iterative algorithms to compute a desirably large PD-RCI set.

In the next section, we derive parameter-dependent matrix inequality conditions for (6.10), (6.11) and (6.12). These conditions will be later used to obtain LMI conditions which solve **Problem 4** and **Problem 5**.

6.3 Sufficient Parameter Dependent Conditions for Invariance and Performance

For brevity, we will first suppress the time dependent representation of the considered signals and use superscript '+' to indicate its successor value. Also, arguments of the matrices $\mathcal{A}_{\mathcal{K}}(\xi)$, $\mathcal{E}(\xi)$, $\mathcal{C}_{\mathcal{K}}(\xi)$, $\mathcal{P}(\xi)$ and the set $\boldsymbol{\mathcal{S}}(\xi)$ will be suppressed. We are now ready to derive sufficient conditions for invariance and the performance constraints.

6.3.1 Parameter dependent conditions for invariance and system constrains

From (6.6) and (6.10), a set $\boldsymbol{\mathcal{S}}$ is invariant if for a given $\xi \in \boldsymbol{\Xi}$, and for each $x \in \boldsymbol{\mathcal{S}}(\xi)$

$$(1 - (e_i^T \mathcal{P}(\xi^+) W^{-1} x^+)^2) \geq 0, \forall (w, \xi^+) \in (\boldsymbol{\mathcal{W}}, \boldsymbol{\Xi}), i = 1, \ldots, n_p. \quad (6.13)$$

Using S-procedure [72], (6.5) and (6.6), an equivalent condition for (6.13) can be written as

$$\begin{aligned}\phi_i(1 - (e_i^T \mathcal{P}(\xi^+) W^{-1} x^+)^2) \geq (\mathbf{1} - \mathcal{P} W^{-1} x)^T \Lambda_i (\mathbf{1} + \mathcal{P} W^{-1} x) \\ + (\mathbf{1} - G w)^T \Gamma_i (\mathbf{1} + G w), \forall (w, \xi^+) \in (\boldsymbol{\mathcal{W}}, \boldsymbol{\Xi}), i = 1, \ldots, n_p, \quad (6.14)\end{aligned}$$

where $\phi_i \in \mathbb{R}_+$, $\Lambda_i \in \mathbb{D}_+^{n_p}$ and $\Gamma_i \in \mathbb{D}_+^{n_g}$. The vector x^+ in (6.14) should satisfy (6.8), hence we can rewrite (6.14) as

$$\chi_1^T \begin{bmatrix} r_i & 0 & 0 & 0 \\ 0 & W^{-T} \mathcal{P}^T \Lambda_i \mathcal{P} W^{-1} & 0 & 0 \\ 0 & 0 & G^T \Gamma_i G & 0 \\ 0 & 0 & 0 & -p_i \end{bmatrix} \chi_1 \succeq 0,$$
$$\forall \begin{bmatrix} 0 & -\mathcal{A}_{\mathcal{K}} & -\mathcal{E} & I \end{bmatrix} \chi_1 = 0, \quad (6.15)$$

where $\chi_1 = \begin{bmatrix} 1 & x^T & w^T & (x^+)^T \end{bmatrix}^T$, $r_i = \phi_i - \mathbf{1}^T \Lambda_i \mathbf{1} - \mathbf{1}^T \Gamma_i \mathbf{1}$ and $p_i = W^{-T} \mathcal{P}^T(\xi^+) e_i \phi_i e_i^T \mathcal{P}(\xi^+) W^{-1}$. We will use **Lemma 6**, in order to derive a sufficient condition for (6.15). In particular, by choosing $\Psi_i(\xi) = \begin{bmatrix} 0 & 0 & 0 & \mathcal{V}_i(\xi)^{-1} \end{bmatrix}^T$ in **Lemma 6**, where $\mathcal{V}_i(\xi) = \sum_{k=1}^{N_\xi} \xi_k(t) V_i^k$, with $V_i^k \in \mathbb{R}^{n_x \times n_x}$, and by using congruence transformation, we get a sufficient condition for (6.15) as follows

$$\begin{bmatrix} r_i & 0 & 0 & 0 \\ 0 & \mathcal{P}^T \Lambda_i \mathcal{P} & 0 & \mathcal{A}_{\bar{\mathcal{K}}}^T \\ 0 & 0 & G^T \Gamma_i G & \mathcal{E}^T \\ 0 & * & * & \mathsf{He}(\mathcal{V}_i) - \mathcal{V}_i^T p_i \mathcal{V}_i \end{bmatrix} \succ 0, \forall \xi^+ \in \mathbf{\Xi},\ i = 1, \ldots, n_p, \tag{6.16}$$

where $\mathcal{A}_{\bar{\mathcal{K}}} = \mathcal{A}_{\mathcal{K}} W$ and $\bar{\mathcal{K}}(\xi) = \mathcal{K}(\xi) W \triangleq \sum_{k=1}^{N_\xi} \xi_k(t) \bar{K}^k$. With the intention to resolve the nonlinearity in the $(4,4)$-block of (6.16), we now introduce a positive-definite matrix variable X_i that satisfies

$$X_i^{-1} - p_i \succ 0. \tag{6.17}$$

Thus from (6.16) and (6.17), we obtain a sufficient parameter dependent matrix inequality condition for (6.10) as

$$\begin{bmatrix} W^T X_i^{-1} W & * \\ \phi_i e_i^T \mathcal{P}(\xi^+) & \phi_i \end{bmatrix} \succ 0, \tag{6.18a}$$

$$\begin{bmatrix} r_i & 0 & 0 & 0 \\ 0 & \mathcal{P}^T \Lambda_i \mathcal{P} & 0 & \mathcal{A}_{\bar{\mathcal{K}}}^T \\ 0 & 0 & G^T \Gamma_i G & \mathcal{E}^T \\ 0 & * & * & \mathsf{He}(\mathcal{V}_i) - \mathcal{V}_i^T X_i^{-1} \mathcal{V}_i \end{bmatrix} \succ 0, \tag{6.18b}$$

$$\forall \xi^+ \in \mathbf{\Xi}, i = 1, \ldots, n_p.$$

In the next lemma we present sufficient parameter dependent invariance conditions for the system (6.8), satisfying the invariance condition (6.10) and state constraint condition (6.11) . We would refer readers to **Lemma 4**, which will be utilized in the derivation.

Lemma 13: *For some arbitrary matrices* $Y_i \in \mathbb{R}^{n_x \times n_x}$, $\bar{\Lambda}_i \in \mathbb{D}_+^{n_p}$, $i = 1, \ldots, n_p$, $\bar{\Pi}_j \in \mathbb{D}_+^{n_p}$, $j = 1, \ldots, n_h$ *and* $P_0^k \in \mathbb{R}^{n_p \times n_x}$, $k = 1, \ldots, N_\xi$, *if there exist* $P^k \in \mathbb{R}^{n_p \times n_x}$, $\bar{K}^k \in \mathbb{R}^{n_u \times n_x}$, $V_i^k \in \mathbb{R}^{n_x \times n_x}$, $W \in \mathbb{R}^{n_x \times n_x}$, ,

$X_i = X_i^T \in \mathbb{R}^{n_x \times n_x}$, $\phi_i \in \mathbb{R}_+, \Lambda_i \in \mathbb{D}_+^{n_p}$, $\Gamma_i \in \mathbb{D}_+^{n_g}$, $\Pi_j \in \mathbb{D}_+^{n_p}$ *satisfying conditions* (6.19a),(6.19b), (6.19c) *and* (6.20) *reported below, then a PD-RCI set can be obtained as in (6.6) and the PDCL as* $\mathcal{K}(\xi) = \bar{\mathcal{K}}(\xi)W^{-1}$*:*

$$\begin{bmatrix} W^T Y_i + Y_i^T W - Y_i^T X_i Y_i & * \\ \phi_i e_i^T \mathcal{P}(\xi^+) & \phi_i \end{bmatrix} \succ 0, \tag{6.19a}$$

$$\phi_i - \mathbf{1}^T \Lambda_i \mathbf{1} - \mathbf{1}^T \Gamma_i \mathbf{1} \succ 0, \tag{6.19b}$$

$$\sum_{k=1}^{N_\xi} \xi_k^2 \boldsymbol{M}_i^{k,k}(\bar{\Lambda}_i, \Lambda_i) + \sum_{k=1}^{N_\xi - 1} \sum_{l=k+1}^{N_\xi} \xi_k \xi_l (\boldsymbol{M}_i^{k,l}(\bar{\Lambda}_i, \Lambda_i) + \boldsymbol{M}_i^{l,k}(\bar{\Lambda}_i, \Lambda_i)) \succ 0, \tag{6.19c}$$

$$\sum_{k=1}^{N_\xi} \xi_k^2 \boldsymbol{R}_j^{k,k}(\bar{\Pi}_j, \Pi_j) + \sum_{k=1}^{N_\xi - 1} \sum_{l=k+1}^{N_\xi} \xi_k \xi_l (\boldsymbol{R}_j^{k,l}(\bar{\Pi}_j, \Pi_j) + \boldsymbol{R}_j^{l,k}(\bar{\Pi}_j, \Pi_j)) \succeq 0, \tag{6.20}$$

where,

$$\boldsymbol{P}^{k,l}(\bar{\Lambda}_i, \Lambda_i) = \mathsf{He}((P^k)^T \bar{\Lambda}_i P_0^l) - (P_0^k)^T \bar{\Lambda}_i \Lambda_i^{-1} \bar{\Lambda}_i P_0^l, \tag{6.21}$$

$$\boldsymbol{M}_i^{k,l}(\bar{\Lambda}_i, \Lambda_i) = \begin{bmatrix} \boldsymbol{P}^{k,l}(\bar{\Lambda}_i, \Lambda_i) & * & * & * \\ 0 & G^T \Gamma_i G & * & * \\ A^k W + B^k \bar{K}^l & E^k & \mathsf{He}(V_i^k) & * \\ 0 & 0 & V_i^k & X_i \end{bmatrix},$$

$$\boldsymbol{R}_j^{k,l}(\bar{\Pi}_j, \Pi_j) = \begin{bmatrix} 2 - \mathbf{1}^T \Pi_j \mathbf{1} & e_j^T (H_x W + H_u \bar{K}^l) \\ * & \boldsymbol{P}^{k,l}(\bar{\Pi}_j, \Pi_j) \end{bmatrix}.$$

Proof. For some matrix Y_i, we obtain (6.19a) by applying **Lemma 4** in $(1,1)$ block of (6.18a). Next, we consider (6.18b), it can be rewritten as

$$\begin{bmatrix} r_i & 0 \\ 0 & \bar{\mathcal{M}}_i \end{bmatrix} \succ 0. \tag{6.22}$$

The condition (6.19b) is directly implied from $(1,1)$ block of (6.22), it also implies $\bar{\mathcal{M}}_i \succ 0$. Furthermore, by using **Lemma 4** again in the $(1,1)$ block of $\bar{\mathcal{M}}_i$ in (6.22) with some matrix $\mathcal{Y} = \bar{\Lambda}_i \mathcal{P}_0(\xi)$, where $\mathcal{P}_0(\xi) = \sum_{k=1}^{N_\xi} \xi_k P_0^k$, and $\bar{\Lambda}_i \in \mathbb{D}_+^{n_p}$, followed by application of Shur complement lemma, we get

$$\begin{bmatrix} \mathsf{He}(\mathcal{P}^T\bar{\Lambda}_i\mathcal{P}_0)-\mathcal{P}_0^T\bar{\Lambda}_i\Lambda_i^{-1}\bar{\Lambda}_i\mathcal{P}_0 & 0 & \mathcal{A}_{\mathcal{K}}^T & 0 \\ * & G^T\Gamma_i G & \mathcal{E}^T & 0 \\ * & * & \mathsf{He}(\mathcal{V}_i) & \mathcal{V}_i^T \\ * & * & * & X_i \end{bmatrix} \succ 0. \tag{6.23}$$

It is straightforward to verify that (6.23) can be rewritten in polynomial form[1] as (6.19c).

Thus a sufficient condition for invariance condition (6.18) is given by (6.19). Additionally, the PD-RCI set $\boldsymbol{\mathcal{S}}$ has to satisfy the state and input constraints condition (6.11). By employing S-procedure, it can be equivalently expressed as follows

$$2(1-(e_j^T(H_x+H_u\mathcal{K})x)) \succeq (\mathbf{1}-\mathcal{P}W^{-1}x)^T\Pi_j(\mathbf{1}+\mathcal{P}W^{-1}x), \tag{6.24}$$

By expressing (6.24) in a quadratic form, using congruence transformation and applying **Lemma 4** with some matrix $\mathcal{Y}_1 = \bar{\Pi}_j\mathcal{P}_0(\xi)$, the following sufficient condition is obtained

$$\begin{bmatrix} 2-\mathbf{1}^T\Pi_j\mathbf{1} & e_j^T(H_xW+H_u\bar{\mathcal{K}}(\xi)) \\ * & \mathsf{He}(\mathcal{P}^T\bar{\Pi}_j\mathcal{P}_0)-(\mathcal{P}_0)^T\bar{\Pi}_j\Pi_j^{-1}\bar{\Pi}_j\mathcal{P}_0 \end{bmatrix} \succeq 0, \tag{6.25}$$

which in turn can be equivalently written as (6.20). □

Note that the matrix $\boldsymbol{P}^{k,l}(\bar{\Pi}_j,\Pi_j)$ in (6.20) has a similar form as (6.21). We again emphasize that a solution which satisfies (6.19) and (6.20) for any arbitrary choice of matrices Y_i, $\bar{\Lambda}_i$, $\bar{\Pi}_j$ and P_0^k gives a PD-RCI set $\boldsymbol{\mathcal{S}}$ and an invariance inducing PDCL $\mathcal{K}$. From **Lemma 4** we know that ideally, the choices of these matrices should be $Y_i = X_i^{-1}W$, $\bar{\Lambda}_i = \Lambda_i$, $\bar{\Pi}_j = \Pi_j$ and $P_0^k = P^k$, which makes the obtained conditions nonlinear. Thus, later we will present a systematic way to select these matrices, which helps us to achieve the least conservative tractable linear conditions.

[1] Deriving the polynomials in (6.19c), we recall the introduced notation for matrix valued functions $\boldsymbol{L}^{k,l}(\bar{\Theta},\Theta) = L(X^k,Y^l,\bar{\Theta},\Theta)$ and the simplex assumption, $\sum_{k=1}^{N_\xi}\xi_k = 1$.

6.3.2 Parameter dependent performance constraints

We next derive parameter dependent matrix inequality conditions for performance constraint (6.12). Since we consider performance for $w(t) = 0,\ \forall t \geq 0$, we can ignore the matrix $\mathcal{E}$ in (6.8). Now, let $\mathcal{Q}(\xi(t)) = \sum_{k=1}^{N_\xi} \xi_k(t) Q^k \succeq 0$ and $Q^k \in \mathbb{R}^{n_x \times n_x}$, then the performance constraint (6.12) is satisfied by the closed loop system (6.8) and (6.9) within the set $\boldsymbol{\mathcal{S}}$ if [49], [94]:

$$\left\| \mathcal{Q}^{-1/2} x(t) \right\|_2^2 \leq \gamma,\ \forall x(t) \in \boldsymbol{\mathcal{S}}(\xi), \tag{6.26a}$$

$$\left\| \mathcal{Q}^{-1/2}(\xi^+) x(t+i+1) \right\|_2^2 - \left\| \mathcal{Q}^{-1/2} x(t+i) \right\|_2^2 \ \leq\ - \|z(t+i)\|_2^2\,. \tag{6.26b}$$

It is easy to verify that (6.26) implies (6.12) by summing both sides of (6.26b) from $i = 0$ to $i = \infty$. In the next lemma we present parameter dependent sufficient conditions for (6.26a) and (6.26b).

Lemma 14: *For a given γ, and some arbitrary matrices $\bar{\Upsilon} \in \mathbb{D}^{n_p}$ and $P_0^k \in \mathbb{R}^{n_p \times n_x}$, $k = 1, \ldots, N_\xi$, the performance constraints* (6.12) *is fulfilled by the closed-loop system* (6.8) *and* (6.9) *within the set $\boldsymbol{\mathcal{S}}(\xi)$, if there exist $P^k \in \mathbb{R}^{n_p \times n_x}$, $W \in \mathbb{R}^{n_x \times n_x}$, $Q^k \in \mathbb{R}^{n_x \times n_x}$, $S^k \in \mathbb{R}^{n_x \times n_x}$, $F^k \in \mathbb{R}^{n_z \times n_z}$ and $\Upsilon \in \mathbb{D}_+^{n_p}$ satisfying the following conditions:*

$$\sum_{k=1}^{N_\xi} \xi_k^2 \boldsymbol{N}^{k,k} + \sum_{k=1}^{N_\xi - 1} \sum_{l=k+1}^{N_\xi} \xi_k \xi_l (\boldsymbol{N}^{k,l} + \boldsymbol{N}^{l,k}) \succeq 0. \tag{6.27a}$$

$$\sum_{k=1}^{N_\xi} \xi_k^2 \boldsymbol{L}^{k,k}(\bar{\Upsilon}, \Upsilon) + \sum_{k=1}^{N_\xi - 1} \sum_{l=k+1}^{N_\xi} \xi_k \xi_l (\boldsymbol{L}^{k,l}(\bar{\Upsilon}, \Upsilon) + \boldsymbol{L}^{l,k}(\bar{\Upsilon}, \Upsilon)) \ \succeq\ 0. \tag{6.27b}$$

where,

$$\boldsymbol{N}^{k,l} = \begin{bmatrix} \mathsf{He}(W) - Q^k & * & * & * & * \\ A^k W + B^k \bar{K}^l & \mathsf{He}(S^k) & * & * & * \\ 0 & S^k & Q^k & * & * \\ C^k W + D^k \bar{K}^l & 0 & 0 & \mathsf{He}(F^k) & * \\ 0 & 0 & 0 & F^k & I \end{bmatrix},$$

$$\boldsymbol{L}^{k,l}(\bar{\Upsilon},\Upsilon) = \begin{bmatrix} \gamma - \mathbf{1}^T\Upsilon\mathbf{1} & * & * \\ 0 & \boldsymbol{P}^{k,l}(\bar{\Upsilon},\Upsilon) & * \\ 0 & W & Q^k \end{bmatrix}.$$

Proof. Let $\chi_2 = \begin{bmatrix} x(t+i)^T & x(t+i+1)^T & z(t+i)^T \end{bmatrix}$. Since $x(t+i+1)$ and $z(t+i)$ in (6.26b) have to satisfy (6.8) and (6.9), this can be expressed as

$$\chi_2^T \begin{bmatrix} \mathcal{Q}^{-1} & * & * \\ 0 & -\mathcal{Q}^{-1}(\xi^+) & 0 \\ 0 & 0 & -I \end{bmatrix} \chi_2 \succeq 0, \forall \begin{bmatrix} -\mathcal{A}_{\mathcal{K}} & I & 0 \\ -\mathcal{C}_{\mathcal{K}} & 0 & I \end{bmatrix} \chi_2 = 0. \tag{6.28}$$

Choosing $\Psi = \begin{bmatrix} 0 & \mathcal{S}^{-1}(\xi) & 0 \\ 0 & 0 & \mathcal{F}^{-1}(\xi) \end{bmatrix}^T$ in **Lemma 6**, and using congruence transform followed by Schur complement, a sufficient condition for (6.28) can be expressed as

$$\begin{bmatrix} W^T\mathcal{Q}^{-1}W & * & * & * & * \\ \mathcal{A}_{\bar{\mathcal{K}}} & \mathsf{He}(\mathcal{S}) & * & * & * \\ 0 & \mathcal{S} & \mathcal{Q}(\xi^+) & * & * \\ \mathcal{C}_{\bar{\mathcal{K}}} & 0 & 0 & \mathsf{He}(\mathcal{F}) & * \\ 0 & 0 & 0 & \mathcal{F} & I \end{bmatrix} \succeq 0. \tag{6.29}$$

The nonlinearity in the $(1,1)$-block of (6.29) can be resolved by application of **Lemma 4**. To avoid inverse parameter dependent term (see, *Remark 1*), we select $\mathcal{Y} = I$ while linearizing (6.29). As before, the obtained inequality condition can be thus written in polynomial form (6.27a).

Lastly for (6.26a), an equivalent condition obtained by employing S-procedure is

$$\gamma_2 - x(t)\mathcal{Q}^{-1}x(t) \succeq (\mathbf{1} - \mathcal{P}W^{-1}x(t))^T\Upsilon(\mathbf{1} + \mathcal{P}W^{-1}x(t)). \tag{6.30}$$

Expressing (6.30) in quadratic form and by applying **Lemma 4** with $\mathcal{Y} = \bar{\Upsilon}\mathcal{P}_0(\xi)$, a parameter dependent matrix inequality condition can be written as (6.27b). □

Notice that the performance constraints (6.27b) depend on matrices $\bar{\Upsilon}$ and P_0^k, their ideal choices are Υ and P^k, respectively. Nonetheless, to avoid nonlinearity, later we will present systematic choices of these matrices.

In this section, we have obtained parameter dependent matrix inequality conditions for invariance (6.10), system constraints (6.11), and performance constraints (6.12) which are given by (6.19), (6.20) (Lemma 13), and (6.27) (Lemma 14), respectively. These matrix inequality conditions are linear if $\boldsymbol{P}^{k,l}$ is linear. Assuming P_0^k is known, the linearity of the matrix $\boldsymbol{P}^{k,l}$ in turn depends on the matrices $\bar{\Lambda}_i$ (and $\bar{\Pi}_j$, $\bar{\Upsilon}$) and P^k. Resolving the nonlinearity in $\boldsymbol{P}^{k,l}$ was one of the main motivations of the formulations of **Problem 4** and **Problem 5**. Furthermore, the matrix inequality conditions are parameter dependent, hence solving them in the current form can be intractable. Nevertheless, since $\xi_k \in \Xi$ are positive, it is easy to verify that a sufficient condition for (6.19c) would be $\boldsymbol{M}_i^{k,k}(\bar{\Lambda}_i, \Lambda_i) \succeq 0,\ k = 1, \dots, N_\xi$ (necessary conditions) and $\boldsymbol{M}_i^{k,l}(\bar{\Lambda}_i, \Lambda_i) + \boldsymbol{M}_i^{l,k}(\bar{\Lambda}_i, \Lambda_i) \succeq 0,\ k = 1, \dots, N_\xi - 1, l = k+1, \dots, N_\xi$ (sufficient conditions). Similar matrix inequality conditions can be also obtained for system and performance constraints. From [74], [75], it is known that these sufficient conditions can be conservative. By applying Pólya relaxation, the conservatism can be reduced at the cost of an increased number of total LMI conditions depending upon the choice of the order of relaxation.

In the sequel, we derive sufficient LMI conditions for obtained parameter dependent matrix inequality conditions using Polya's relaxation.

6.4 Tractable LMI Feasibility Conditions

At this point, we would direct readers to **Theorem 1**, which will be extensively used in the subsequent derivations of feasibility conditions. We will need following modified multinomial coefficients, which were originally defined in **Theorem 1**, $\mathfrak{X}_q^i(d, a) = d!/(\beta_1! \cdots (\beta_i - a)! \cdots \beta_r!)$ if $\beta_i - a \in \mathbb{Z}_+$ otherwise 0, and $\mathfrak{X}_q^{ij}(d, a, b) = d!/(\beta_1! \cdots (\beta_i - a)! \cdots (\beta_j - b)! \cdots \beta_r!)$ if $(\beta_i - a), (\beta_j - b) \in \mathbb{Z}_+$ otherwise 0.

As explained earlier, in order to resolve the nonlinearity due to $\boldsymbol{P}^{k,l}$, in the next theorem we fix the matrices $P^k = P_0^k = P_{init}$, $k = 1, \dots, N_\xi$, where P_{init} is some known matrix. Thus we can allow matrices $\bar{\Lambda}_i = \Lambda_i$, $\bar{\Pi}_j = \Pi_j$, $\bar{\Upsilon} = \Upsilon$. We now present one of the main results which gives tractable LMI feasibility conditions for **Problem 4**.

Theorem 6: *Let $\mathcal{P}(\xi) = \mathcal{P}_0(\xi) = P_{init}$ be a given matrix and $d \in \mathbb{Z}_+$ be the desired order of Polya's relaxation, then **Problem 4** has a feasible solution if,*

i. for invariance and system constraints, there exist $\bar{K}^k \in \mathbb{R}^{n_u \times n_x}$, $V_i^k \in$

$\mathbb{R}^{n_x \times n_x}$, $k = 1, \cdots, N_\xi$, $W \in \mathbb{R}^{n_x \times n_x}$, $X_i = X_i^T \in \mathbb{R}^{n_x \times n_x}$, $\phi_i \in \mathbb{R}_+, \Lambda_i \in \mathbb{D}_+^{n_p}$, $\Gamma_i \in \mathbb{D}_+^{n_g}$, $i = 1, \ldots, n_p$, $\Pi_j \in \mathbb{D}_+^{n_p}$, $j = 1, \ldots, n_h$ *satisfying:*

$$\begin{bmatrix} \mathsf{He}(W^T Y_i) - Y_i^T X_i Y_i & * \\ \phi_i e_i^T P_{init} & \phi_i \end{bmatrix} \succ 0, \tag{6.31a}$$

$$\phi_i - \mathbf{1}^T \Lambda_i \mathbf{1} - \mathbf{1}^T \Gamma_i \mathbf{1} \succ 0, \tag{6.31b}$$

$$\begin{aligned} \boldsymbol{\mathcal{M}}_i(d,q) \triangleq & \sum_{k=1}^{N_\xi} \mathfrak{X}_q^k(d,2) \boldsymbol{M}_i^{k,k}(\Lambda_i, \Lambda_i) \\ & + \sum_{k=1}^{N_\xi - 1} \sum_{l=k+1}^{N_\xi} \mathfrak{X}_q^{kl}(d,1,1)(\boldsymbol{M}_i^{k,l}(\Lambda_i, \Lambda_i) + \boldsymbol{M}_i^{l,k}(\Lambda_i, \Lambda_i)) \succ 0, \\ & q = 1, \ldots, \mathfrak{L}(d+2, N_\xi), \end{aligned} \tag{6.31c}$$

$$\begin{aligned} \boldsymbol{\mathcal{R}}_j(d,q) \triangleq & \sum_{k=1}^{N_\xi} \mathfrak{X}_q^k(d,2) \boldsymbol{R}_j^{k,k}(\Pi_j, \Pi_j) \\ & + \sum_{k=1}^{N_\xi - 1} \sum_{l=k+1}^{N_\xi} \mathfrak{X}_q^{kl}(d,1,1)(\boldsymbol{R}_j^{k,l}(\Pi_j, \Pi_j) + \boldsymbol{R}_j^{l,k}(\Pi_j, \Pi_j)) \succ 0, \\ & q = 1, \ldots, \mathfrak{L}(d+2, N_\xi). \end{aligned} \tag{6.32}$$

ii. for performance constraints (6.12)*, there exist* $W \in \mathbb{R}^{n_x \times n_x}$, $\bar{K}^k \in \mathbb{R}^{n_u \times n_x}$, $Q^k \in \mathbb{R}^{n_x \times n_x}$, $S^k \in \mathbb{R}^{n_x \times n_x}$, $F^k \in \mathbb{R}^{n_z \times n_z}$,$k = 1, \cdots, N_\xi$, *and* $\Upsilon \in \mathbb{D}_+^{n_p}$ *for a given* γ_2 *satisfying*

$$\boldsymbol{\mathcal{N}}(d,q) \triangleq \sum_{k=1}^{N_\xi} \mathfrak{X}_q^k(d,2) \boldsymbol{N}^{k,k} + \sum_{k=1}^{N_\xi - 1} \sum_{l=k+1}^{N_\xi} \mathfrak{X}_q^{kl}(d,1,1)(\boldsymbol{N}^{k,l} + \boldsymbol{N}^{l,k}) \succ 0, \tag{6.33a}$$

$$\begin{aligned}\boldsymbol{\mathcal{L}}(d,q) \triangleq &\sum_{k=1}^{N_\xi} \mathfrak{X}_q^k(d,2)\boldsymbol{L}^{k,k}(\Upsilon,\Upsilon)\\ &+\sum_{k=1}^{N_\xi-1}\sum_{l=k+1}^{N_\xi} \mathfrak{X}_q^{kl}(d,1,1)(\boldsymbol{L}^{k,l}(\Upsilon,\Upsilon)+\boldsymbol{L}^{l,k}(\Upsilon,\Upsilon)) \succ 0,\\ &\qquad q=1,\ldots,\mathfrak{L}(d+2,N_\xi).\end{aligned} \tag{6.33b}$$

An RCI set can then be obtained as in (6.6) and the PDCL $\mathcal{K}(\xi) = \bar{\mathcal{K}}(\xi)W^{-1}$.

Proof. *i.* Considering $\mathcal{P}(\xi) = \mathcal{P}_0(\xi) = P_{init}$, (6.31a) and (6.31b) are directly obtained from (6.19a) and (6.19b), respectively. Next we consider (6.19c), which is a homogeneous matrix valued polynomial of degree 2 and choose $\bar{\Lambda}_i = \Lambda_i$. For some $d \in \mathbb{Z}_+$, a polynomial of degree $(d+2)$ can be obtained by multiplying both the sides of (6.19c) by $(\sum_{k=1}^{N_\xi}\xi_k)^d$, which is given by

$$\sum_{q=1}^{\mathfrak{L}(d+2,N_\xi)} \boldsymbol{\mathcal{M}}_i(d,q)\xi_1^{\beta_1}\xi_2^{\beta_2}\cdots\xi_{N_\xi}^{\beta_{N_\xi}} \succeq 0, \beta_1\beta_2\cdots\beta_{N_\xi} = \mathcal{J}_q(d{+}2), \tag{6.34}$$

Since ξ_k belongs to a unit simplex (6.3), from **Theorem 1**, a sufficient condition for (6.19c) is (6.31c). Similarly, (6.20) can be written as homogeneous polynomial of degree $(d+2)$, letting $\bar{\Pi}_j = \Pi_j$ and using Polya's relaxation theorem we obtain (6.32).

ii. Can be proved using similar approach as earlier. Notice that in (6.33b) we substitute $\bar{\Upsilon} = \Upsilon$.

□

Note that, even if P^k's are assumed to be constant in **Theorem 6**, the variable matrix W allows to reshape the RCI set. A similar construction to find initial RCI set was also proposed in [49], [55]. We now proceed to formulating feasibility conditions for **Problem 5** in the next theorem. In the theorem, the matrices P^k's are now variable and thus according to **Remark 1** we now fix $\bar{\Lambda}_i = \Lambda_i^0$, $\bar{\Pi}_j = \Pi_j^0$, $\bar{\Upsilon} = \Upsilon^0$.

Theorem 7: *Let* $\mathcal{P}_0(\xi)$ *and* $W \in \mathbb{R}^{n_x \times n_x}$ *be given matrices and* $d \in \mathbb{Z}_+$ *be the desired order of Polya's relaxation, then* ***Problem 5*** *has a feasible solution if,* *i. for invariance and system constraints, there exist* $P^k \in \mathbb{R}^{n_p \times n_x}$, $\bar{K}^k \in \mathbb{R}^{n_u \times n_x}$, $V_i^k \in \mathbb{R}^{n_x \times n_x}$, $k = 1, \cdots, N_\xi$, $X_i = X_i^T \in \mathbb{R}^{n_x \times n_x}$, $\phi_i \in \mathbb{R}_+, \Lambda_i \in \mathbb{D}_+^{n_p}$, $\Gamma_i \in \mathbb{D}_+^{n_g}$, $i = 1, \ldots, n_p$, $\Pi_j \in \mathbb{D}_+^{n_p}$, $j = 1, \ldots, n_h$ *satisfying:*

$$\begin{bmatrix} \mathsf{He}(W^T Y_i) - Y_i^T X_i Y_i & * \\ e_i^T P^k & \phi_i^{-1} \end{bmatrix} \succ 0, \tag{6.35a}$$

$$\begin{bmatrix} \phi_i^{-1} - \mathbf{1}^T \bar{\Gamma}_i \mathbf{1} & * \\ \phi_i^{-1} \mathbf{1} & \Lambda_i^{-1} \end{bmatrix} \succeq 0, \tag{6.35b}$$

$$\begin{aligned} \bar{\boldsymbol{\mathcal{M}}}_i(d,q) &\triangleq \sum_{k=1}^{N_\xi} \mathfrak{X}_q^k(d,2) \bar{\boldsymbol{M}}_i^{k,k}(\Lambda_i^0, \Lambda_i) \\ &+ \sum_{k=1}^{N_\xi - 1} \sum_{l=k+1}^{N_\xi} \mathfrak{X}_q^{kl}(d,1,1)(\bar{\boldsymbol{M}}_i^{k,l}(\Lambda_i^0, \Lambda_i) + \bar{\boldsymbol{M}}_i^{l,k}(\Lambda_i^0, \Lambda_i)) \succ 0, \end{aligned} \tag{6.35c}$$

$$\begin{aligned} \boldsymbol{\mathcal{R}}_j(d,q) &\triangleq \sum_{k=1}^{N_\xi} \mathfrak{X}_q^k(d,2) \boldsymbol{R}_j^{k,k}(\Pi_j^0, \Pi_j) \\ &+ \sum_{k=1}^{N_\xi - 1} \sum_{l=k+1}^{N_\xi} \mathfrak{X}_q^{kl}(d,1,1)(\boldsymbol{R}_j^{k,l}(\Pi_j^0, \Pi_j) + \boldsymbol{R}_j^{l,k}(\Pi_j^0, \Pi_j)) \succ 0, \\ & q = 1, \ldots, \mathfrak{L}(d+2, N_\xi). \end{aligned} \tag{6.36}$$

where,

$$\bar{\boldsymbol{M}}_i^{k,l}(\Lambda_i^0, \Lambda_i) = \begin{bmatrix} \boldsymbol{P}^{k,l}(\Lambda_i^0, \Lambda_i) & * & * & * \\ 0 & G^T \bar{\Gamma}_i G & * & * \\ A^k W + B^k \bar{K}^l & \phi_i^{-1} E^k & \mathsf{He}(V_i^k) & * \\ 0 & 0 & V_i^k & X_i \end{bmatrix} \tag{6.37}$$

ii. for performance constraints, there exist $P^k \in \mathbb{R}^{n_p \times n_x}$, $\bar{K}^k \in \mathbb{R}^{n_u \times n_x}$, $Q^k \in \mathbb{R}^{n_x \times n_x}$, $S^k \in \mathbb{R}^{n_x \times n_x}$, $F^k \in \mathbb{R}^{n_z \times n_z}$, $k = 1, \cdots, N_\xi$, *and* $\Upsilon \in \mathbb{D}_+^{n_p}$

for a given γ_2 satisfying

$$\boldsymbol{\mathcal{N}}(d,q) \triangleq \sum_{k=1}^{N_\xi} \mathfrak{X}_q^k(d,2)\boldsymbol{N}^{k,k} + \sum_{k=1}^{N_\xi-1}\sum_{l=k+1}^{N_\xi} \mathfrak{X}_q^{kl}(d,1,1)(\boldsymbol{N}^{k,l}+\boldsymbol{N}^{l,k}) \succ 0, \tag{6.38a}$$

$$\begin{aligned}\boldsymbol{\mathcal{L}}(d,q) \triangleq & \sum_{k=1}^{N_\xi} \mathfrak{X}_q^k(d,2)\boldsymbol{L}^{k,k}(\Upsilon^0,\Upsilon) \\ & + \sum_{k=1}^{N_\xi-1}\sum_{l=k+1}^{N_\xi} \mathfrak{X}_q^{kl}(d,1,1)(\boldsymbol{L}^{k,l}(\Upsilon^0,\Upsilon) + \boldsymbol{L}^{l,k}(\Upsilon^0,\Upsilon)) \succ 0. \end{aligned} \tag{6.38b}$$

A PD-RCI set can then be obtained as in (6.6) and the PDCL is $\mathcal{K}(\xi) = \bar{\mathcal{K}}(\xi)W^{-1}$.

Proof. *i*. We obtain (6.35a) from (6.19a) by application of congruence transform and using the fact that it is affinely dependent on the parameter. By using Schur complement lemma on (6.19b), and substituting $\bar{\Gamma}_i = \phi_i^{-2}\Gamma_i$ we get (6.35b). We now consider (6.23), replacing $\Gamma_i = \phi_i^2\bar{\Gamma}_i$ and $\bar{\Lambda}_i = \Lambda_i^0$, and using congruence transform it can be written as

$$\sum_{k=1}^{N_\xi} \xi_k^2 \bar{\boldsymbol{M}}_i^{k,k}(\Lambda_i^0,\Lambda_i) + \sum_{k=1}^{N_\xi-1}\sum_{l=k+1}^{N_\xi} \xi_k\xi_l(\bar{\boldsymbol{M}}_i^{k,l}(\Lambda_i^0,\Lambda_i) + \bar{\boldsymbol{M}}_i^{l,k}(\Lambda_i^0,\Lambda_i)) \succ 0. \tag{6.39}$$

Where $\bar{\boldsymbol{M}}_i^{k,l}$ is given in (6.37). Since (6.39) is homogeneous matrix valued polynomial of degree 2, by employing Polya's relaxation theorem we obtain (6.35c). Similarly, (6.36) was obtained from (6.20), by substituting $\bar{\Pi}_j = \Pi_j^0$ and using Polya's relaxation theorem.

ii. Can be proved using similar approach as earlier. Notice that in (6.33b) we replace $\bar{\Upsilon} = \Upsilon^0$.

□

By finding an feasible solution for **Theorem 7** we obtain a PD-RCI set.

However, the inequalities (6.35), (6.36) and (6.38) depend on the matrices P_0^k, Λ_i^0, Π_j^0 and Υ^0, which are the initial guess of matrices P^k, Λ_i, Π_j and Υ, respectively. Finding an initial guess for these matrices is not straight forward; we thus obtain them by solving **Problem 4**. It is easy to verify that using solutions from **Problem 4** to initialize **Problem 5** always preserves feasibility of solutions, see **Remark 1**.

We next present our iterative algorithm to compute PD-RCI set and invariance inducing PDCL.

6.5 Iterative PD-RCI Set Computation

Our primary goal is to compute PD-RCI set (6.6) of desirably large volume and the PDCL controller (6.7). Thus, we need to formulate a method which computes a maximum volume set feasible to conditions proposed in **Theorem 6** and **Theorem 7**. In the original form, the conditions in these theorems were nonlinear, and to make them tractable for solving, we linearized them by using **Lemma 4**. As mentioned in the **Remark 1**, the linearization introduces conservatism, which can be reduced by using an iterative scheme. In the iterative scheme, we first consider **Problem 4** in which we assumed $P(\xi) = P^0(\xi) = P_{init}$. As shown in Section 5.5.2, the volume of the considered RCI set is directly proportional to $|\mathsf{det}(W)|$. Thus, we next propose an optimization problem which computes a desirably large RCI set for **Problem 4**.

6.5.1 Initial RCI set computation

We develop an iterative scheme in which we solve a determinant maximization problem under LMI conditions presented in **Theorem 6**. We will try to iteratively maximize the volume to avoid enforcing symmetry on W. Similar to previous chapters, we first introduce a condition

$$W^T W^0 + (W^0)^T W - (W^0)^T W^0 \succcurlyeq Z \succ 0. \tag{6.40}$$

Note that this condition is necessarily satisfied with $W = W^0$. As a result, maximization of $\mathsf{det}(Z)$ under (6.40) would lead to a solution W that satisfies

$$|\mathsf{det}(W)| \geq |\mathsf{det}(W^0)|. \tag{6.41}$$

Moreover, as explained in **Remark 1**, at each iteration we update $Y_i = (X_i^0)^{-1}W^0$ in (6.31a), where X_i^0 is previous solution of X_i. This allows us to develop the following iterative algorithm to compute RCI sets of increased volume at each step for a priori chosen matrix P_{init}

$$\left.\begin{array}{cc} \max\limits_{\substack{\phi_i, W, \bar{K}^k, V_i^k, X_i, \Lambda_i, \Gamma_i \\ \Pi_j, Q^k, S^k, F^k, \Upsilon, Z}} & \mathsf{log\,det}(Z) \\ \text{subject to:} & (6.31), (6.32), (6.33) \text{ and } (6.40) \end{array}\right\} \tag{6.42}$$

Initial Optimization to Compute W^0: Condition (6.40) is removed and $\mathsf{log\,det}(Z)$ is changed to $\mathsf{log\,det}(W + W^T)$; (6.31a) is imposed with $Y_i \to \psi I$.

Thus, by solving (6.42) iteratively, we obtain a desirably large RCI set and the initial matrices for the **Problem 5**. Note that it is sufficient to solve (6.42) for just one iteration. Nevertheless, providing optimized solutions to **Problem 5** could be beneficial, as it may help in achieving larger PD-RCI sets. At this point, we recall Remark 4, which proposes few heuristics to select initial matrix P_{init}. We next present an algorithm which computes PD-RCI sets.

6.5.2 Computation of PD-RCI sets

In order to compute a desirably large set for **Problem 5**, we formulate a new optimization problem for volume maximization of PD-RCI set treating matrices P^k's as its optimization variables. For this problem, we fix the matrix W obtained by solving (6.42). By construction, for each $\xi \in \boldsymbol{\Xi}$, $\boldsymbol{\mathcal{S}}(\xi)$ is an 0-symmetric polytope in the state-space. Thus, an intuitive way to maximize the volume of such a set is to compute matrices P^k's such that the sum of the volumes of each slice of $\boldsymbol{\mathcal{S}}(\xi)$ corresponding to a $\xi \in \boldsymbol{\Xi}$ is maximized. However, maximizing infinite slices of the PD-RCI set would lead to solving a semi-infinite problem, which may be intractable. Nevertheless, to deal with such intractability, we only maximize the slices $\boldsymbol{\mathcal{S}}(\xi^m)$ corresponding to the finite set of grid points $\xi^m \in \boldsymbol{\Xi}$, $m = 1, \ldots, N_m$ (see, Section 2.3.3).

Since the slices $\boldsymbol{\mathcal{S}}(\xi^m)$ are polytopes, we can utilize the volume maximization approach presented in **Appendix 6.A**.

Proposition 1: *The maximum volume polytopic invariant sets* $\boldsymbol{\mathcal{S}}(\xi^m) = \left\{x \in \mathbb{R}^{n_x} : -\mathbf{1} \leq \mathcal{P}(\xi^m)W^{-1}x \leq \mathbf{1}\right\}$, $m = 1, \ldots, N_m$, *can be obtained by solv-*

ing the following convex Semi-definite Programming *(SDP) problem in an iterative manner,*

$$
\left.\begin{array}{cc}
\underset{\substack{\phi_i, P^k, \bar{K}^k, V_i^k, X_i, \Lambda_i, \Gamma_i \\ \Pi_j, Q_1^k, S^k, F^k, \Upsilon, \sigma_n^m}}{\min} & \sum_{n=1}^{N_\sigma} \sum_{m=1}^{N_m} \sigma_n^m \\
\textit{subject to:} & \sigma_n^m \geq 0, \\
 & \begin{bmatrix} \widetilde{P}W^{-1} \\ -\widetilde{P}W^{-1} \end{bmatrix} \widetilde{x}_n - \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \leq \begin{bmatrix} \widetilde{\sigma} \\ \widetilde{\sigma} \end{bmatrix}, \\
 & (6.35), (6.36) \textit{ and } (6.38).
\end{array}\right\} \tag{6.43}
$$

where $\widetilde{P} = [\mathcal{P}(\xi^1)^T, \cdots, (\mathcal{P}(\xi^{N_m})^T]^T \in \mathbb{R}^{n_p N_m \times n_x}$, $\widetilde{\sigma} = [\sigma_n^1 \mathbf{1}, \cdots, \sigma_n^{N_m} \mathbf{1}]^T \in \mathbb{R}^{n_p N_m}$ *and* $\{\widetilde{x}_n\}_{n=1}^{N_\sigma}$ *are the vertices of some known* n_x *dimensional outer bounding box* $\mathfrak{B}$ *which contains the state constraint set* $\boldsymbol{\mathcal{X}}$*.*

Proof. We refer the reader to **Section 6.A.1** for the details of the volume maximization algorithm which is based on Monte-Carlo technique. The convex cost function formulated based on the algorithm presented in **Appendix 6.A.1**, is combined with LMI conditions (6.35), (6.36) and (6.38) for invariance, system and performance constraints respectively, giving an SDP problem (6.43). □

Assuming that the initial values of $\mathcal{P}_0, W, Y_i, \Lambda_i^0, \Gamma_j^0$ and Υ^0 are available after solving (6.42), we summarize the whole approach to compute PD-RCI set in **Algorithm 1**.

Algorithm 1 : Computing PD-RCI set.

Input: System (6.1), $\boldsymbol{\mathcal{X}_u}$, $\boldsymbol{\mathcal{W}}$, $\mathcal{P}_0$, W, Y_i Λ_i^0, Γ_j^0, Υ^0
Output: $\mathcal{P}(\xi)$, $\mathcal{K}(\xi)$
while Iteration ≥ 0 **do**

  [$\mathcal{P}$,$\bar{\mathcal{K}}$,X_i,Λ_i, Γ_j, Υ] $\leftarrow$ solve (6.43)

  Update: $Y_i \leftarrow X_i^{-1}W$, $\mathcal{P}_0 \leftarrow \mathcal{P}$, $\Lambda_i^0 \leftarrow \Lambda_i$,
  $\Gamma_j^0 \leftarrow \Gamma$, $\Upsilon^0 \leftarrow \Upsilon$

  Iteration $\leftarrow$ Iteration $- 1$

We have already shown that **Algorithm 1** always has a feasible solution at the first iteration if initialized using solutions from (6.42). The update scheme in the algorithm alleviates the conservatism introduced while linearizing the equation (6.35a), (6.35c), (6.36) and (6.38b) using **Lemma 4**. The systematic update procedure also guarantees that the solutions from the previous iteration are feasible in the current iteration (see, **Remark 1**). Thus, we find a new PD-RCI set of larger volume at each iteration until the specified number of iterations are performed, or convergence is achieved. We purposely present termination of the algorithm based on the number of iteration instead of convergence to emphasize that latter is not necessary.

Next, we address some known implementation issues and some workarounds which could potentially help to overcome it.

6.5.3 Practical Issues

Computational complexity

We compute PD-RCI set by solving problem (6.42) to obtain an initial solution, and in each iteration of the **Algorithm 1**, we solve an updated optimization problem (6.43). The computation complexity of both the optimization problems is similar. Depending upon the selected order of Polya's relaxation, the optimization problems will consist of $n_p \times (N_\xi + 1 + \mathfrak{L}(d+2, N_\xi))$ LMI constraints for invariance (6.35), $n_h \times \mathfrak{L}(d+2, N_\xi)$ LMI constraints for system constraints (6.36), and $2 \times \mathfrak{L}(d+2, N_\xi)$ LMI constraints for performance (6.38). Furthermore, for volume maximization we introduced $2n_pN_\xi \times N_\sigma$ affine inequality constraints. All the constraints are in terms of $(3n_p + N_\xi(8+n_p) + n_h)$ matrix variables and $(n_p + N_\sigma \times N_m)$ scalar variables. The computational complexity can be largely impacted by choice of representational complexity n_p and the relaxation order d. Theoretically, their values should be as large as possible to have the least conservative formulation. Thus, by choosing n_p and d carefully, a trade-off can be achieved between computational complexity and conservatism. One strategy could be to keep d small initially and select matrix P_{init} and thus n_p that represents a full dimensional bounded polytope.If the algorithm is infeasible, then d can be increased along with n_p, and if feasible, they can still be improved to obtain larger PD-RCI set. Another strategy to reduce computational complexity, again at the cost of conservatism, could be to reduce the number of system vertices N_ξ. We can reduce N_ξ by modifying the system description such that trajectories of the modified system overbounds the trajectories of the original. However, it may not always be possible to reduce N_ξ.

Computation of the RCI set for quasi-LPV systems

If the scheduling parameters $\xi(t)$ are function of system states and input, then the system is referred to as *quasi-LPV* (qLPV). Thus, in qLPV systems, initial $\xi(t)$ cannot be selected independently from the $x(t)$ and $u(t)$. To address this issue, we can keep the RCI set description independent of parameter i.e., by restricting $P^k = P,\ \forall k = 1, \dots, N_\xi$ or construct a set $\breve{\boldsymbol{\mathcal{S}}} = \bigcap_{\forall \xi(t) \in \boldsymbol{\Xi}} \boldsymbol{\mathcal{S}}(\xi(t))$. Notice that the set $\breve{\boldsymbol{\mathcal{S}}}$ also satisfies conditions (6.10) and (6.11) if $\boldsymbol{\mathcal{S}}(\xi(t))$ does, and since $\breve{\boldsymbol{\mathcal{S}}}$ is independent of ξ, we call it RCI set. The set $\breve{\boldsymbol{\mathcal{S}}}$ may be larger compared to the one obtained by restricting P^k due to larger number of variables involved in the overall optimization problem when computing former. Even though we define $\breve{\boldsymbol{\mathcal{S}}}$ as the intersection of infinite slices of $\boldsymbol{\mathcal{S}}(\xi(t))$, we will next prove that it can be accurately obtained by performing intersections of only a few selected slices. More specifically, we will prove that

$$\breve{\boldsymbol{\mathcal{S}}} = \bigcap_{\forall \xi \in \boldsymbol{\Xi}} \boldsymbol{\mathcal{S}}(\xi) = \bigcap_{\forall \xi^k \in \boldsymbol{\Xi}} \boldsymbol{\mathcal{S}}(\xi^k), \tag{6.44}$$

where $\xi^k,\ k = 1, \dots, N_\xi$ are the vertices of the set $\boldsymbol{\Xi}$. Since $\xi^k \in \boldsymbol{\Xi}$, to prove (6.44), it is enough to prove $\bigcap_{\forall \xi^k \in \boldsymbol{\Xi}} \boldsymbol{\mathcal{S}}(\xi^k) \subseteq \bigcap_{\forall \xi \in \boldsymbol{\Xi}} \boldsymbol{\mathcal{S}}(\xi)$.

Lemma 15: *Given any polytope of form* (6.6)*, the following relation always holds*

$$\bigcap_{\forall \xi^k \in \boldsymbol{\Xi}} \boldsymbol{\mathcal{S}}(\xi^k) \subseteq \bigcap_{\forall \xi \in \boldsymbol{\Xi}} \boldsymbol{\mathcal{S}}(\xi) \tag{6.45}$$

Proof. Consider any point $x \in \bigcap_{\forall \xi^k \in \boldsymbol{\Xi}} \boldsymbol{\mathcal{S}}(\xi^k)$, to complete the proof it is sufficient to show that $x \in \bigcap_{\forall \xi \in \boldsymbol{\Xi}} \boldsymbol{\mathcal{S}}(\xi)$.

Since $x \in \bigcap_{\forall \xi^k \in \boldsymbol{\Xi}} \boldsymbol{\mathcal{S}}(\xi^k)$, the following inequalities holds element-wise

$$\underbrace{|P^k W^{-1} x|}_{y_k} \leq \mathbf{1},\ \forall k = 1, \dots, N_\xi. \tag{6.46}$$

Thus, for any $\xi = [\xi_1, \xi_2, \cdots, \xi_{N_\xi}]^T \in \boldsymbol{\Xi}$, we have $\sum_{k=1}^{N_\xi} \xi_k y_k \leq \mathbf{1}$ which further implies

$$|\sum_{k=1}^{N_\xi} \xi_k P^k W^{-1} x| \leq \mathbf{1}. \tag{6.47}$$

Hence the relation (6.45) holds. □

Thus for qLPV systems we can construct RCI set $\breve{\boldsymbol{\mathcal{S}}}$ to guarantee constraint satisfaction.

In the next section, we will demonstrate the potential of the proposed algorithm through numerical examples.

6.6 Numerical Examples

The algorithm was implemented in Matlab on a 3.1 GHz Intel Core i7-555U macOS computer with 8 GB RAM with YALMIP [67] and the solver SeDuMi [64]. The computation of the volume and projections of the polytope was done using MPT [87].

6.6.1 1D system

We first consider a simple constrained 1-dimension system from [93] defined as

$$x(k+1) = \theta x(k) + u(k), \tag{6.48}$$

where θ is a time-varying observable parameter and we assume $|\theta| \leq 2$ and $|u| \leq 1$. Now, if θ is treated as uncertainty, then the origin is the only initial state where the constraint control objectives can be achieved, which can also be verified by applying one of the existing methods [33], [35], [46], [49], [55]. However, by transforming (6.48) in the form (6.1) and using the proposed approach, we obtain the control law $u(t) = (\xi_1 K^1 + \xi_2 K^2)x(t)$ and the RCI set $x(t) \in [-1, 1]$, where, $\xi_1 = (2-\theta)/4$, $\xi_2 = (\theta+2)/4$, $K^1 = 1$ and $K^2 = -1$. This example demonstrates the benefit of having a PDCL controller over a robust controller, which is already well known in the literature.

6.6.2 Double Integrator

For a better visualization of the PD-RCI set, we now consider a parameter-varying double integrator system

$$x^+ = \underbrace{\begin{bmatrix} 1+\theta & 1+\theta \\ 0 & 1+\theta \end{bmatrix}}_{\mathcal{A}} x + \underbrace{\begin{bmatrix} 0 \\ 1+\theta \end{bmatrix}}_{\mathcal{B}} u + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\mathcal{E}} w, \tag{6.49}$$

where $|\theta| \leq 0.25$ represent the time varying parameter. The state and control input constraints and the disturbance bounds are expressed as

$$|x| \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix},\ |u| \leq 1,\ |w| \leq 0.25. \tag{6.50}$$

In order to compute RCI set, we first rewrite (6.49) in the form (6.4) with $N_\xi = 2$ and

$$A^1 = \begin{bmatrix} 0.75 & 0.75 \\ 0 & 0.75 \end{bmatrix}, A^2 = \begin{bmatrix} 1.25 & 1.25 \\ 0 & 1.25 \end{bmatrix} \tag{6.51}$$

$$B^1 = \begin{bmatrix} 0 \\ 0.75 \end{bmatrix}, B^2 = \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}, E^1 = E^2 = \mathcal{E} \tag{6.52}$$

where $\xi_1 = (0.25 - \theta)/0.5$ and $\xi_2 = (\theta + 0.25)/0.5$. For computing PD-RCI set using proposed approach, we select P_{init} as described in **Remark 4**. By solving (6.42) for 10 iterations, we find initial values of matrices in **Algorithm 1**. Finally, the PD-RCI set $\boldsymbol{\mathcal{S}}(\xi)$ (shown in Fig. 6.1) is obtained after performing 60 iterations of **Algorithm 1**, which on average took 9.31 seconds per iterations for computations. The values of the obtained matrices are

$$[\ P^1 \mid P^2\ ] = \left[\begin{array}{cc|cc} -0.4111 & -0.1354 & -0.3257 & -0.0854 \\ 0.0303 & -0.5151 & 0.0404 & -0.3823 \\ 0.4867 & -0.2474 & 0.4867 & -0.2474 \\ 0.4884 & -0.0504 & 0.4883 & -0.0506 \end{array}\right],$$

$$\left[\begin{array}{c} W \\ \hline K^1 \\ \hline K^2 \end{array}\right] = \left[\begin{array}{cc} 2.4373 & -0.6691 \\ -0.7327 & 0.8379 \\ \hline -0.2246 & -0.7898 \\ \hline -0.1506 & -0.5601 \end{array}\right]. \tag{6.53}$$

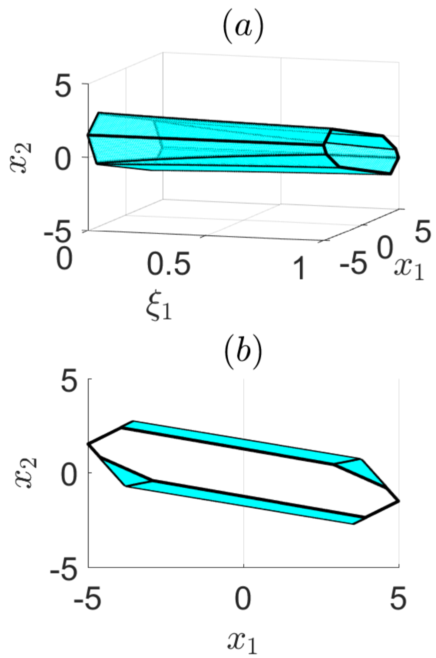

Figure 6.1: (a) Plot of the PD-RCI set $\boldsymbol{\mathcal{S}}(\xi)$ in (6.6) w.r.t ξ_1 and (b) projection $\boldsymbol{\mathcal{S}}(\xi)$ on (x_1, x_2) axis.

In Fig. 6.1 (a), we plot the PD-RCI set for $\xi_1 \in [0, 1]$, and in Fig. 6.1 (b) we show the projection of the set on the state-space axis (x_1, x_2). The RCI set $\breve{\boldsymbol{\mathcal{S}}}$ in (6.44) can be seen in the Fig. 6.1 (b) as bounded colorless region. According to (6.10), the region outside the set $\breve{\boldsymbol{\mathcal{S}}}$ highlighted in cyan consists of points which can be brought within the RCI set $\breve{\boldsymbol{\mathcal{S}}}$ in one step if the initial value of the parameter is selectable. Thus possibly enlarging the overall set of safe initial states for constraint control. To compare the volume gain between **Problem 4** and **Problem 5** we plot the volume of the set $\breve{\boldsymbol{\mathcal{S}}}$ at each iteration in Fig. 6.2. In the figure, it can be seen that there is an additional 23% gain in the volume of the RCI set, after the first 10 iterations for which **Problem 4** is solved.

We plot the set $\breve{\boldsymbol{\mathcal{S}}}$ and the maximal RCI set $\boldsymbol{\Omega}_\infty$ obtained using geomet-

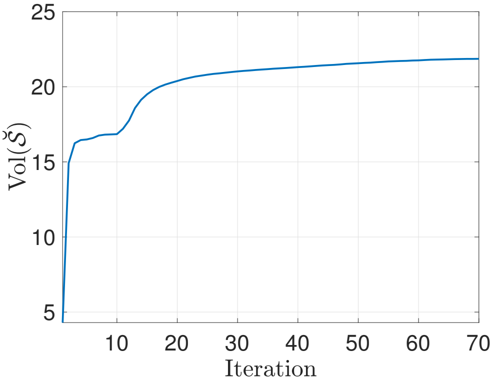

Figure 6.2: Volume of the set $\breve{\boldsymbol{\mathcal{S}}}$ plotted against the iteration.

ric approach [87] in Fig. 6.3. The geometric approach treats parameter as unknown but bounded signals and the control inputs are free from any state-feedback structure. Not surprisingly, the set $\breve{\boldsymbol{\mathcal{S}}}$ (volume equal to 21.7907) computed using the proposed approach is larger than the set the maximal RCI set $\boldsymbol{\Omega}_\infty$ (volume equal 19.3703). Moreover, the overall representational complexity of the set $\breve{\boldsymbol{\mathcal{S}}}$ is 8, which is exactly half the complexity of the set $\boldsymbol{\Omega}_\infty$. These results show the benefits of using PD-RCI sets and PDCL in the LPV setting. Lastly, to incorporate performance guarantees within the PD-RCI sets, we consider performance constraint defined as

$$\left(\sum_{t=0}^{\infty} x(t)^T \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{Q_x} x(t) + \underbrace{0.1}_{Q_u} u(t)^2 \right) \leq \underbrace{10}_{\gamma}. \tag{6.54}$$

By selecting $z(t) = \begin{bmatrix} Q_x^{1/2} \\ 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ Q_u^{1/2} \end{bmatrix} u(t)$ in (6.2), and imposing performance constraints in (6.42) and (6.43), we compute set $\breve{\boldsymbol{\mathcal{S}}}_\gamma$ (shown in Fig. 6.3) by using the procedure explained before.

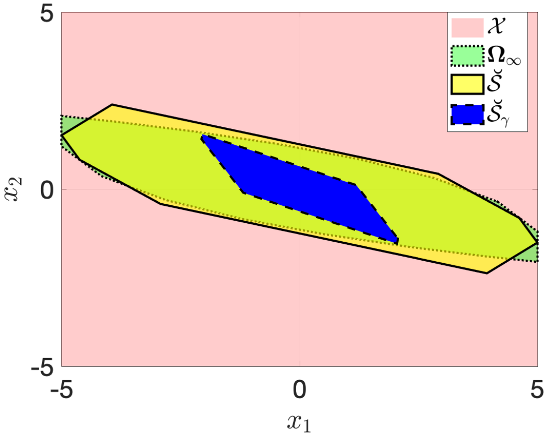

Figure 6.3: Admissible set $\boldsymbol{\mathcal{X}}$ (red), maximal RCI set $\mathbf{\Omega}_\infty$ using geometric approach (green; dotted), maximum volume RCI set $\breve{\boldsymbol{\mathcal{S}}}$ using **Algorithm 1** (yellow; solid) and RCI set guaranteeing (6.54) (blue; dashed).

6.6.3 Nonlinear System

One important application of the proposed approach is to compute RCI sets for nonlinear systems. For this purpose we consider the controlled Van der Pol oscillator system in [95]:

$$\dot{x}_1 = x_2, \tag{6.55a}$$

$$\dot{x}_2 = -x_1 + \mu(1 - x_1^2)x_2 + u, \tag{6.55b}$$

where $\mu = 2$. The system should satisfy the input constraints $|u| \leq 1$ and state constraints $|x_1| \leq 1$, $|x_2| \leq 1$. For computation and simulation purpose we use discretized version of the system with a sampling time of 0.1. For the computation of the RCI set, we first rewrite the system in the quasi-LPV form (6.4) by selecting

$$A^1 = \begin{bmatrix} 1 & 0.1 \\ -0.1 & 1 \end{bmatrix}, A^2 = \begin{bmatrix} 1 & 0.1 \\ -0.1 & 1.2 \end{bmatrix}, \tag{6.56}$$

$$B^1 = B^2 = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}, \tag{6.57}$$

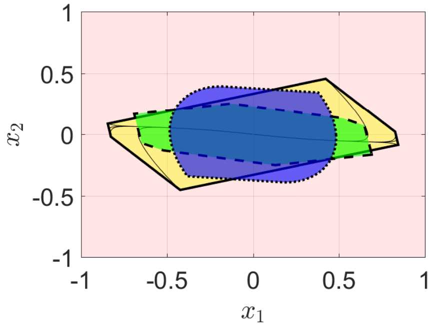

Figure 6.4: Admissible set $\boldsymbol{\mathcal{X}}$ (red), maximum volume RCI set $\breve{\boldsymbol{\mathcal{S}}}$ using **Algorithm 1** (yellow; solid), RCI set using [55] (green; dashed) and RPI set for an LQR controller (blue; dotted) for the Van der Pol oscillator system.

and $\xi_1 = (2-\mu(1-x_1^2))/2$ and $\xi_2 = \mu(1-x_1^2)/2$. Using the proposed approach we compute the matrix variables defining the RCI set and the invariance inducing controller for the nonlinear system which are given as

$$
\begin{aligned}
[P_1|P_2] &= \left[\begin{array}{cc|cc} -0.5066 & -0.1205 & -0.5066 & -0.1358 \\ -0.4349 & -0.0135 & -0.4367 & -0.0134 \\ 0.4238 & -0.2686 & 0.4237 & -0.3173 \\ 0.5280 & 0.0385 & 0.5280 & 0.0385 \end{array}\right] \\
\left[\begin{array}{c} W \\ \hline K_1 \\ \hline K_2 \end{array}\right] &= \left[\begin{array}{cc} 0.4409 & 0.0136 \\ -0.0127 & 0.1090 \\ \hline 0.8341 & -2.3111 \\ \hline 0.8727 & -3.0114 \end{array}\right]
\end{aligned} \tag{6.58}
$$

Since the scheduling parameters ξ_1, ξ_2 are state dependent, we only consider RCI set $\breve{\boldsymbol{\mathcal{S}}}$ (6.44), which is shown in Fig. 6.4. Furthermore, due to state dependence of the parameters, we can rewrite the PDCL controller as a nonlinear state-feedback controller $u = \mathcal{K}(\xi)x = x_2(0.7003x_1^2 - 3.011) - x_1(0.03855x_1^2 - 0.8727)$. The closed-loop trajectories from all the vertices of the set $\breve{\boldsymbol{\mathcal{S}}}$ is also shown in Fig. 6.4.

For comparison, we compute the RCI set of representational complexity (same as $\breve{\boldsymbol{\mathcal{S}}}$) 8 by using the method presented in [55]. The method assumes invariance inducing controller to be linear state-feedback, we show the computed set in Fig. 6.4. Furthermore, we tried to compute the maximal RCI set using the geometric approach [87]. Interestingly, the geometric approach failed to converge even after $24hrs$, so instead, we show robust positive invariant (RPI) set corresponding to a nominal system and LQR controller with tuning matrices $Q = I$ and $R = 1$. The representational complexity of the RPI set is 50. Clearly, the proposed algorithm is more advantageous, since it is able to generate visibly larger RCI sets with less representational complexity.

6.A Appendix

6.A.1 Volume maximization using Monte-Carlo integration

Based on the theory of Monte-Carlo integration [96], we present an approach which can be used to find a desirably large polytope of a predefined maximum complexity, enclosed within some known set.

Let $\boldsymbol{\mathcal{C}}$ be a set defined as,

$$\boldsymbol{\mathcal{C}} = \{x \in \mathbb{R}^{n_x} | Px \leq \mathbf{1}\},$$

where $P \in \mathbb{R}^{n_p \times n_x}$. We consider the following volume maximization problem,

$$\sup_{\boldsymbol{\mathcal{C}}} \int_{\boldsymbol{\mathcal{C}}} dx \quad \text{s.t.} \;\; \boldsymbol{\mathcal{C}} \subseteq \boldsymbol{\mathcal{X}}, \tag{6.59}$$

where $\boldsymbol{\mathcal{X}}$ is a given bounded set, not necessarily a polytope. We assume that the set containment constraints $\boldsymbol{\mathcal{C}} \subseteq \boldsymbol{\mathcal{X}}$ are already available, and are formulated as some finite-number of convex constraints (e.g., LMIs). In this section, we focus on the cost function of (6.59), which characterizes the volume of the polytopic set $\boldsymbol{\mathcal{C}}$. Typically, determining the exact volume $\int_{\boldsymbol{\mathcal{C}}} dx$ of a polytope $\boldsymbol{\mathcal{C}}$ is computationally challenging [91], [97]. In our case, the problem is even more difficult as $\boldsymbol{\mathcal{C}}$ itself is *not* known. To this end, a procedure based on Monte-Carlo methods [96] is formulated, in order to approximate the cost in (6.59).

Let $\mathfrak{B}$ be a *known* outer bounding box which contains the given set $\boldsymbol{\mathcal{X}}$. We generate N independent random samples $\widetilde{\boldsymbol{\mathcal{X}}} = \{\widetilde{x}_j\}_{j=1}^{N}$, which are uniformly distributed in the given outer-bounding box $\mathfrak{B}$.

According to Monte-Carlo integration technique, the volume of the set $\boldsymbol{\mathcal{C}}$ is approximated as,

$$\int_{\mathcal{C}} dx \approx \frac{1}{N}\mathsf{vol}(\mathfrak{B}) \sum_{\widetilde{x}_j \in \widetilde{\mathcal{X}}} \mathbb{I}_{\{\mathcal{C}\}}(\widetilde{x}_j), \tag{6.60}$$

where $\mathsf{vol}(\mathfrak{B})$ denotes the volume of the box $\mathfrak{B}$, and $\mathbb{I}_{\{\mathcal{C}\}}(\widetilde{x}_j)$ is the indicator function of the set $\mathcal{C}$ defined as,

$$\mathbb{I}_{\{\mathcal{C}\}}(\widetilde{x}_j) = \begin{cases} 1, \text{ if } & \widetilde{x}_j \in \mathcal{C}, \\ 0, \text{ if } & \widetilde{x}_j \notin \mathcal{C}. \end{cases} \tag{6.61}$$

Remark 8: From the theory of Monte Carlo integration [96], the following limit holds with probability (w.p) 1:

$$\lim_{N\to\infty} \frac{1}{N}\mathsf{vol}(\mathfrak{B}) \sum_{\widetilde{x}_j \in \widetilde{\mathcal{X}}} \mathbb{I}_{\{\mathcal{C}\}}(\widetilde{x}_j) = \int_{\mathcal{C}} dx, \quad \text{w.p.1.}$$

By using (6.60), the maximum-volume problem (6.59) is approximated as follows,

$$\max_{\mathcal{C}} \sum_{\widetilde{x}_j \in \widetilde{\mathcal{X}}} \mathbb{I}_{\{\mathcal{C}\}}(\widetilde{x}_j) \quad \text{s.t.} \;\; \mathcal{C} \subseteq \mathcal{X}. \tag{6.62}$$

Note that the cost function in problem (6.62) which is the sum of indicator functions $\mathbb{I}_{\{\mathcal{C}\}}(\widetilde{x}_j)$, is non-convex and discontinuous. We next introduce an approximation of the cost function in order to solve (6.62) in a tractable manner. We approximate the discontinuous cost in (6.62) with a continuous-concave function such that the sample points which are contained in $\mathcal{C}$ get the maximum cost, while the value of the cost for all $\widetilde{x}_j \notin \mathcal{C}$, decreases uniformly.

For the sake of convenience, without loss of generality we modify the definition of indicator functions in the cost (6.62) as follows

$$\mathbb{I}_{\{\mathcal{C}\}}(\widetilde{x}_j) = \begin{cases} 0, & \text{if } P\widetilde{x}_j - \mathbf{1} \leq 0, \\ -1, & \text{otherwise.} \end{cases} \tag{6.63}$$

Note that, this modification does not change the optimal solution of problem (6.62).

Approximation of the indicator functions

Let us first consider for each individual hyperplane $\boldsymbol{\mathcal{C}}_i \triangleq \{x : e_i^T(Px - \mathbf{1}) \leq 0\}$ of the set $\boldsymbol{\mathcal{C}}$, the following cost $T_{\{\boldsymbol{\mathcal{C}}_i\}}(\widetilde{x}_j)$,

$$T_{\{\boldsymbol{\mathcal{C}}_i\}}(\widetilde{x}_j) = \begin{cases} 0 & \text{if} \quad e_i^T(P\widetilde{x}_j - \mathbf{1}) \leq 0, \\ -e_i^T(P\widetilde{x}_j - \mathbf{1}) & \text{if} \quad e_i^T(P\widetilde{x}_j - \mathbf{1}) > 0, \end{cases} \tag{6.64}$$

which is a piece-wise linear concave approximation of the indicator functions $\mathbb{I}_{\{\boldsymbol{\mathcal{C}}_i\}}(\widetilde{x}_j)$ defined for the i-th hyperplane of $\boldsymbol{\mathcal{C}}$. The plot of the indicator function $\mathbb{I}_{\{\boldsymbol{\mathcal{C}}_i\}}(\widetilde{x}_j)$ and its approximation $T_{\{\boldsymbol{\mathcal{C}}_i\}}(\widetilde{x}_j)$ is shown in Fig.6.5. The idea of approximating non-convex indicator function $\mathbb{I}_{\{\boldsymbol{\mathcal{C}}_i\}}(\widetilde{x}_j)$ with $T_{\{\boldsymbol{\mathcal{C}}_i\}}(\widetilde{x}_j)$ is similar to the relaxation of l_0-quasi-norm with l_1-norm as introduced in [98], [99] for computing outer-approximating polytopes of non-convex semi-algebraic sets.

We now extend the idea of approximating the indicator functions defined for a single hyperplane $\boldsymbol{\mathcal{C}}_i$, to approximate the indicator function defined over the entire polytopic set $\boldsymbol{\mathcal{C}}$. In particular, we introduce the following concave function $T_{\{\boldsymbol{\mathcal{C}}\}}(\widetilde{x}_j)$ to approximate $\mathbb{I}_{\{\boldsymbol{\mathcal{C}}\}}(\widetilde{x}_j)$ defined in (6.63),

$$T_{\{\boldsymbol{\mathcal{C}}\}}(\widetilde{x}_j) = \begin{cases} 0 & \text{if } P\widetilde{x}_j - \mathbf{1} \leq 0, \\ -\left(\max\limits_{i=1,\ldots,n_p} e_i^T(P\widetilde{x}_j - \mathbf{1})\right) & \text{otherwise.} \end{cases} \tag{6.65}$$

Note that, for the points $\widetilde{x}_j \notin \boldsymbol{\mathcal{C}}$, the cost $T_{\{\boldsymbol{\mathcal{C}}\}}(\widetilde{x}_j)$ is always negative and decays uniformly in all the directions away from $\boldsymbol{\mathcal{C}}$. Based on the approximation $T_{\{\boldsymbol{\mathcal{C}}\}}(\widetilde{x}_j)$ in (6.65) of the indicator functions $\mathbb{I}_{\{\boldsymbol{\mathcal{C}}\}}(\widetilde{x}_j)$, the problem (6.62) is relaxed as follows,

$$\max_{\mathcal{C}} \sum_{\widetilde{x}_j \in \widetilde{\boldsymbol{\mathcal{X}}}} T_{\{\boldsymbol{\mathcal{C}}\}}(\widetilde{x}_j) \quad \text{s.t.} \quad \boldsymbol{\mathcal{C}} \subseteq \boldsymbol{\mathcal{X}} \tag{6.66}$$

Thus, by solving the constraint optimization problem (6.66), we try to find the matrix P (defining the polytope $\mathcal{C}$), which maximizes the number of points $\widetilde{x}_j$ inside the set $\boldsymbol{\mathcal{C}}$, in turn, maximizing its volume, while respecting the constraint $\boldsymbol{\mathcal{C}} \subseteq \boldsymbol{\mathcal{X}}$.

Remark 9: With the choice of cost function $T_{\{\boldsymbol{\mathcal{C}}\}}(\widetilde{x}_j)$ in (6.65), we aim at selecting the matrix P of the polytope $\boldsymbol{\mathcal{C}}$, such that maximum number of

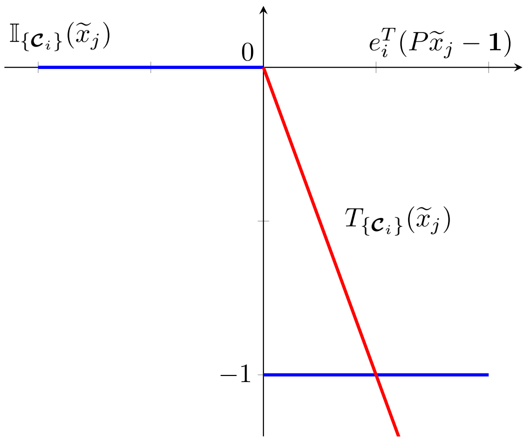

Figure 6.5: Indicator function $\mathbb{I}_{\{\boldsymbol{\mathcal{C}}_i\}}(\widetilde{x}_j)$ (blue) and its concave approximation $T_{\{\boldsymbol{\mathcal{C}}_i\}}(\widetilde{x}_j)$ (red) for the i-th hyperplane. When $e_i^T(P\widetilde{x}_j - \mathbf{1}) < 0$, $\mathbb{I}_{\{\boldsymbol{\mathcal{C}}_i\}}(\widetilde{x}_j)$ and $T_{\{\boldsymbol{\mathcal{C}}_i\}}(\widetilde{x}_j)$ are overlapped and both are 0.

points lie in the set, i.e., $\widetilde{x}_j \in \boldsymbol{\mathcal{C}}$. This is due to the fact that, the value of the cost $T_{\{\boldsymbol{\mathcal{C}}\}}(\widetilde{x}_j)$ decreases linearly for the sample points $\widetilde{x}_j$ which lie outside the set $\boldsymbol{\mathcal{C}}$. We observe that, with this choice of the cost function, it is sufficient to select the sample points $\{\widetilde{x}_j\}_{j=1}^{N_\sigma}$ which lie on the boundary of the known outer-bounding box $\boldsymbol{\mathcal{B}}$. This significantly reduces the computation cost to solve the optimization problem (6.66). Thus, we have considered only the sample points which are on the boundary of $\boldsymbol{\mathcal{B}}$, instead of uniformly distributed samples.

Finally, the cost function in (6.66) can be seen as a sum of concave functions, which can be equivalently expressed as following convex *minimization* problem,

$$
\begin{array}{cc}
\underset{P,\sigma_j}{\min} & \sum_{j=1}^{N} \sigma_j \\
\text{s.t.} & \sigma_j \geq 0, \quad \forall j = 1, \ldots, N \\
& P\widetilde{x}_j - \mathbf{1} \leq \sigma_j \mathbf{1}, \quad \forall j = 1, \ldots, N \\
& \boldsymbol{\mathcal{C}} \subseteq \boldsymbol{\mathcal{X}},
\end{array}
\tag{6.67}
$$

where $\sigma_j \in \mathbb{R}$. Thus, the final volume maximization consist of a linear cost and constraints, alongwith a set containment constraint.

Part III

Safe Controller design for Vehicle Motion Control

CHAPTER 7

Vehicle Motion Control Problem

7.1 Introduction

The recent advancements in sensing, perception and control engineering are pushing automated driving (AD) technologies into our society. It is then natural to question the maturity of the available technologies, especially concerning the new and demanding safety requirements imposed on them by various governmental and non-governmental organizations. As explained in [100], [101], for high-level autonomous vehicles, the controller has to guarantee the satisfaction of recently introduced Automotive Safety Integrity Level (ASIL) requirements mentioned in ISO 26262. Under ASIL, autonomous motion control is classified to meet the ASIL D type of requirement, which imply, the vehicle manufacturer has to guarantee a failure rate of not more than 10^{-8} events per hour. Thus, as a part ASIL D requirement, keeping the vehicle within predefined safety bounds should be guaranteed by the control algorithm. The chapter proposes a vehicle motion control strategies with guaranteed satisfaction of constraints on the maximum deviation from a reference path.

For vehicle motion control, a commonly used control architecture is presented in Figure 7.1, consisting of subsystems like trajectory planner and controller. The reference trajectory, which usually depends on the desired path, is provided to the controller by a higher-level planner. A steering action and longitudinal force input are decided by the controller, which is then applied to the vehicle for tracking the reference.

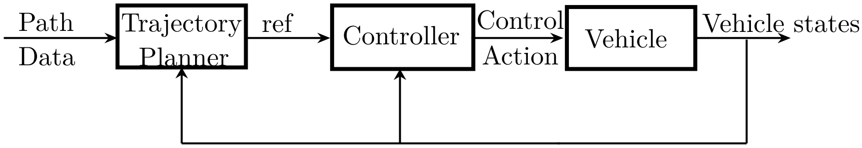

Figure 7.1: Control Architecture

It is desired that these subsystems (trajectory planner and controller) work together to achieve design specifications related to safety and performance. For instance, as a design specification, the trajectory planner should strictly generate references which can be tracked by the controller while staying within the physical limits of the actuators. Also, the reference trajectories should always keep the vehicle on the drivable area of the road. In return, the controller is required to guarantee bounded tracking error for safety. Further, the controller should also limit lateral and longitudinal accelerations to preserve ride comfort.

For such a safety-critical application, the subsystems should always meet the design specifications by construction. Towards this, a direct approach would be to construct them in a monolithic fashion. However, an obvious drawback of such a system is the size of the design problem, which can be very large and computationally intensive. Moreover, any modification in one of the subsystem involves redesigning the whole system. Hence, unlikely to be the design approach of any development team.

Another approach, which we use in this chapter, is to design each subsystem in a modular way. For this kind of modular design, a popular framework is the contract based design (CBD) approach [102]. In the CBD approach, contracts are established between subsystems in a way that each subsystem guarantees the satisfaction of its specifications, assuming that the other subsystems do not violate theirs. At the cost of potential conservatism, a large control problem is decomposed into smaller subproblems designed to meet their contract requirements. This kind of approach has already been applied in other automotive applications [103], [104].

In this chapter, we restrict our interest in designing the controller, which ensures safety by design. We decouple the controller from the planner by using the CBD approach. In order to meet the design specifications, the controller needs to impose requirements on the references which the planner is allowed to generate. These requirements are in the form of bounded rate of change of path curvature and the desired longitudinal acceleration. Thus assuming that the planner generates reference meeting the derived requirements, the controller guarantees to meet its design specification.

The control strategy should ensure constraint satisfaction; hence we opt for interpolation-based control (IBC) strategy. The main advantage of using an IBC controller over other constrained control strategies like MPC is its computational complexity, which can be very cheap, especially when dealing with nonlinear and uncertain systems [21]. Furthermore, as shown in [21], it is easy to construct an IBC controller in an explicit form. The explicit form can be beneficial for the considered safety-critical application since it can be extensively verified and validated offline before implementing the controller. As a result, we obtain a computationally efficient vehicle motion control algorithm which guarantees safety and performance by design. Since the RCI sets are used to construct an IBC controller, we will utilize some of the algorithms proposed in previous chapters for the computation of the sets.

7.2 Problem Description

We now present nonlinear vehicle dynamics taken from [105], and the system constraints. We formalize the problem statement at the end of the section.

7.2.1 Vehicle Dynamics

The overall vehicle motion can be described by

$$\dot{e}_y = \dot{y} + v_x e_\psi, \tag{7.1a}$$

$$\dot{v}_y = -v_x\dot{\psi} + \frac{1}{m_v}(F_{y_f} + F_{y_r}), \tag{7.1b}$$

$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_d, \tag{7.1c}$$

$$\ddot{\psi} = \frac{1}{I_z}(l_f F_{y_f} - l_r F_{y_r}), \tag{7.1d}$$

$$\dot{e}_x = v_x - v_d, \tag{7.1e}$$

$$\ddot{e}_x = \frac{1}{m_v}(F_x - F_{drag} - F_{rolling} - m_v g sin(\theta)) - \dot{v}_d, \tag{7.1f}$$

where $F_{yi} = -C_i\alpha_i$, $i \in \{f, r\}$, $\alpha_f = \frac{v_y + l_f\dot{\psi}}{v_x} - \delta$, $\alpha_r = \frac{v_y - l_r\dot{\psi}}{v_x}$. We summarize all the other variables and parameters in Table 7.1 and Table 7.2. For more clarity, we also explain these variables in Figure 7.2 and Figure 7.3. The equations (7.1a)-(7.1d) describes the lateral motion of the vehicle, whereas the longitudinal motion is described by (7.1e) and (7.1f).

Since the vehicle dynamics (7.1) is nonlinear, we can compactly write them as

$$\dot{z} = f(z(t), u(t), w(t)), \tag{7.2}$$

where, $z = [e_y, \dot{y}, e_\psi, \dot{\psi}, e_x, \dot{e}_x]^T$, $u = [\delta, F_x]^T$ and $w = [\dot{\psi}_d, F_d/m_v, \dot{v}_d]^T$ as state, input and disturbance vectors, respectively and $F_d = -(F_{drag} + F_{rolling} + m_v g sin(\theta))$. We classify input $\dot{\psi}_d$ and $\dot{v}_d$ as (observable) disturbance since they are provided by the trajectory planner to the controller. Note that it is sometime assumed that v_d is provided by the planner instead of $\dot{v}_d$, in this case $\dot{v}_d$ can be computed using finite difference over v_d.

7.2.2 Constraints

As shown in Table 7.1, we want to restrict some of the state and all the input variables described in (7.19) and (7.29) within a bounded set. The constraints are imposed on these variables due to various safety, physical and performance specifications. For instance, as part of ASIL D-type safety requirements, the vehicle's lateral and longitudinal deviation from the desired path should always be bounded. Hence we desire

$$|e_y| \leq 0.4, \; |e_x| \leq 1. \tag{7.3}$$

As we focus on the highway driving scenarios, we restrict the the longitudinal velocity of the vehicle to be $v_x, v_d \in [50, 100]$ *km/hr*. Furthermore, to preserve the passengers' ride comfort, we limit the lateral velocity and the lateral and longitudinal accelerations.

Finally, the steering input δ and the longitudinal force F_x are limited due to the actuators' physical limitations. Since all the considered constraints are polytopic, they can be together expressed as

$$\boldsymbol{\mathcal{Z}} \triangleq \{z|\; Hz \leq \mathbf{1}\}, \; \boldsymbol{\mathcal{U}} \triangleq \{u|\; Gu \leq \mathbf{1}\}, \tag{7.4}$$

where $\mathbf{1}$ is a vector of ones of a compatible dimension. Moreover, bounds on the input disturbances in (7.2) are assumed to be symmetric, hence can be written as

$$\boldsymbol{\mathcal{W}} \triangleq \{w| \, |Dw| \leq \mathbf{1}\}. \tag{7.5}$$

We next detail the our overall design objectives.

7.2.3 Controller Design Objectives and Challenges

The main design objectives in this control design problem are as follows

i. Using the CBD framework, decompose the planner design from the controller with the least possible conservatism.

ii. Construct a controller which guarantees the reference tracking, while keeping the vehicle within constraints (7.4).

iii. The controller should be computationally efficient for real-time implementation and suitable for offline close-loop verification.

In our formulation, we assume that the planner generates the references desired yaw rate $\dot{\psi}_d$ and acceleration $\dot{v}_d$. Thus to satisfy objective (i), designed controller should be robust to all the possible trajectories of $\dot{\psi}_d$ and $\dot{v}_d$ generated by the planner. Nevertheless, to reduce the overall design conservatism, we restrict the generated trajectories based on physical considerations and performance requirements, e.g. ride comfort.

An obvious choice for the control strategy is MPC since the vehicle dynamics (1) are nonlinear and constrained. There are already some nonlinear MPC schemes proposed solving similar problems in the literature, see, e.g.,[106], [107]. However, the main challenge with these schemes is in the real-time solution of the nonlinear program, which is known to be computationally demanding and not easily upper bounded a priori. Thus, as explained in the introduction, to meet the objective (ii) and (ii), we opt for the IBC control strategy.

In the next section, we will briefly present IBC based constrained control strategy.

7.3 Interpolation-Based Control

We now briefly discuss an existing result on the IBC control strategy proposed in [21]. Originally these results were presented for uncertain systems without

Table 7.1: Description of vehicle dynamics variables

Variable	**Description**	**Constraints**
e_y	Vehicle lateral error w.r.t to a predefined path	$\lvert e_y\rvert \leq 0.4[m]$
$\dot{y}$	Lateral velocity in the vehicle body frame	$\lvert\dot{y}\rvert \leq 3[m/s]$
e_ψ	Orientation error w.r.t to a predefined path	$\lvert e_\psi\rvert \leq 10\frac{\pi}{180}[rad/s]$
$\dot{\psi}$	yaw rate	
δ	Steering angle at the vehicle wheel base	$\lvert\delta\rvert \leq 20\frac{\pi}{180}[rad]$
v_x	Vehicle longitudinal velocity	$50 \leq v_x \leq 100[km/hr]$
v_d	Desired longitudinal velocity	$50 \leq v_d \leq 100[km/hr]$
e_x	Vehicle longitudinal error w.r.t to a predefined path	$\lvert e_x\rvert \leq 1[m]$
F_x	Longitudinal force	$\lvert F_x\rvert \leq 6500[N]$
$\dot{\psi}_d$	Desired yaw rate	$\lvert\dot{\psi}_d\rvert \leq \Upsilon_1\ [rad/s]$
$\dot{v}_d$	Desired longitudinal acceleration	$\lvert\dot{v}_d\rvert \leq \Upsilon_2\ [m/s^2]$
θ	Road inclination	$\lvert\theta\rvert \leq \bar{\theta}[rad]$

Table 7.2: Vehicle parameters

Parameter	**Description**	**Value**[units]
m_v	Mass of vehicle	2164 $[kg]$
I_z	Yaw moment of inertia	4373 $[kgm^2]$
C_r	Rear cornering stiffness coeff.	228088$[N/rad]$
C_f	Front cornering stiffness coeff.	142590 $[N/rad]$
l_r	Rear axle to CoG distance	1.6456 $[m]$
l_f	Front axle to CoG distance	1.3384 $[m]$
g	Acceleration due to gravity	9.81 $[m/s^2]$
F_{drag}	Drag force on the vehicle	$\frac{1}{2}\rho C_d A_f v_x^2\ [N]$
$F_{rolling}$	rolling resistance	$rm_v gcos(\theta)\ [N]$
ρ	Mass density of air	1.184$[kg/m^2]$
C_d	Air drag coeff.	0.36
A_f	Frontal area of vehicle	$1.6 + 0.00056(m_v - 765)[m^2]$
r	Rolling resistance coeff.	0.0105

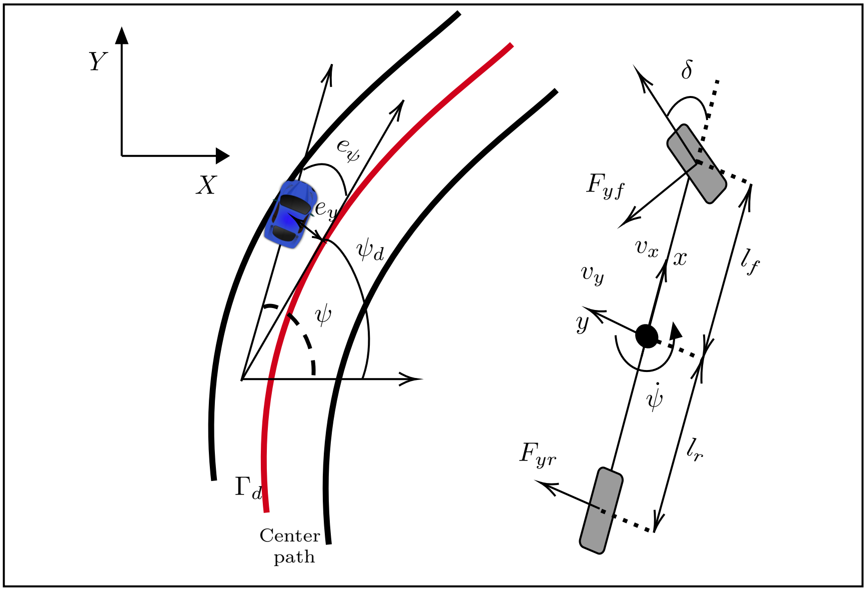

Figure 7.2: Vehicle lateral dynamics

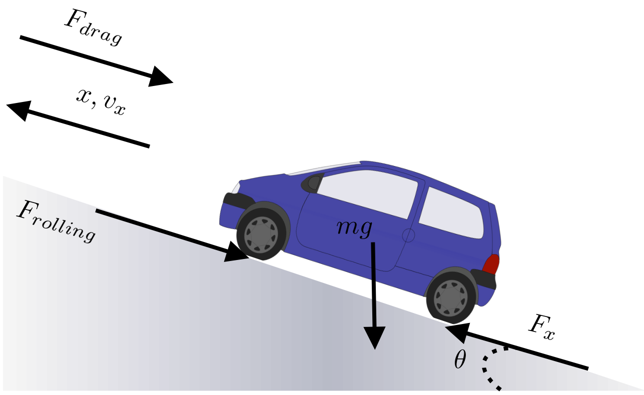

Figure 7.3: Longitudinal forces acting on a vehicle

any additive disturbance. Nevertheless, it is straightforward to extend them to the considered case. We hence introduce the following new system to detail the construction of the IBC controller.

$$x(k+1) = \mathcal{A}(\Delta)x(k) + \mathcal{B}(\Delta)u(k) + \mathcal{E}(\Delta)w(k), \tag{7.6}$$

where, $x(k) \in \mathbb{R}^{n_x}$, $u(k) \in \mathbb{R}^{n_u}$ and $w(k) \in \mathbb{R}^{n_w}$ are state, input and disturbance vectors, respectively. Further, Δ represents uncertainty in the system matrices. It is assumed that $x(k)$, $u(k)$, $w(k)$ and $\Delta(k)$ are subject (or belong) to the constraints :

$$\boldsymbol{\mathcal{X}} = \{x(k) \mid \; Hx(k) \leq \mathbf{1}\}, \tag{7.7a}$$
$$\boldsymbol{\mathcal{U}} = \{u(k) \mid \; Gu(k) \leq \mathbf{1}\}, \tag{7.7b}$$
$$\boldsymbol{\mathcal{W}} = \{w(k) \mid \; |Dw(k)| \leq \mathbf{1}\}, \tag{7.7c}$$
$$\boldsymbol{\Delta} = \{\Delta(k) \mid \; \Delta(k) \in \mathsf{Conv}(\boldsymbol{\Delta}^v)\}. \tag{7.7d}$$

Here $H \in \mathbb{R}^{n_h \times n_x}$, $G \in \mathbb{R}^{n_g \times n_u}$ and $D \in \mathbb{R}^{n_d \times n_w}$ are given matrices, and $\boldsymbol{\Delta}^v = \{\Delta^1, \Delta^2, \ldots, \Delta^\eta\}$ represents finitely many generator matrices.

In the IBC control strategy, the control inputs are usually calculated by performing interpolation between inputs from two or many controllers with different properties. As a result, the final IBC controller has properties influenced by all the interpolating controllers. Since the constraint control of (7.6) is the main objective, a desirable property for the IBC controller is to have a large feasibility region. Within the feasibility set, it is also desirable for the final controller to meet certain performance criteria. Hence, we tactically select the first controller in the interpolation to have large feasibility region and the second guaranteeing desired performance.

In the Chapter 5, we saw that for the system (7.6) and (7.7), a controller with a large feasibility region could be constructed using the vertex-based control strategy, briefly recalled next.

Vertex Control

Let $\boldsymbol{\mathcal{C}}_v$ be the contractive set obtained by solving Algorithm 4 and represented as

$$\boldsymbol{\mathcal{C}}_v = \{x(k) \mid S_v x(k) \leq \mathbf{1}\}. \tag{7.8}$$

With x^i being a vertex of $\mathcal{C}_v$ and $i = 1, \dots, 2\sigma$, we can also write $\mathcal{C}_v = \mathsf{Conv}\{x^1, x^2, \dots, x^{2\sigma}\}$. Furthermore, let u^i be the admissible corresponding control input at the vertex x^i. Then for any $x(k) \in \mathcal{C}_v$, an admissible control input can be computed as

$$u_v(k) = \sum_{i=1}^{2\sigma} \lambda_i^*(k) u^i, \tag{7.9}$$

where $\lambda_i^*(k)$ is obtained by solving

$$\begin{array}{lll} \lambda_i^*(k) & = \underset{\lambda_i}{arg\,min} & \sum_{i=1}^{2\sigma} \lambda_i \\ & st. & \sum_{i=1}^{2\sigma} \lambda_i x^i = x(k), \\ & & 0 \leq \lambda_i \leq 1. \end{array} \tag{7.10}$$

As shown in Section 5.6, the vertex control law (7.9), (7.10) guarantees recursive feasibility and closed-loop robust asymptotic stability for all initial states $x(0) \in \mathcal{C}_v$. Thus, the vertex controller is also an invariance inducing controller.

We next present the controller which can be used for inducing performance.

Performance Inducing Control

For the system (7.6) and (7.7), we assume that the desired performance criteria is given by

$$J = \sum_{k=0}^{\infty} x(k)^T Q_x x(k) + u(k)^T Q_u u(k) \leq \gamma. \tag{7.11}$$

We can compute an RCI set and controller meeting the performance criteria (7.11) by using proposed Algorithm 3. Let the associated RCI set be $\mathcal{C}_p$ and defined as

$$\mathcal{C}_p = \{x(k) \mid S_p x(k) \leq \mathbf{1}\}, \tag{7.12}$$

and the corresponding invariance inducing controller be

$$u_p(k) = K_p x(k). \tag{7.13}$$

It is obvious that the set $\mathcal{C}_p$ is smaller than $\mathcal{C}_v$ due to additional constraints for performance and linear state-feedback (7.13). We now present the IBC based strategy, which interpolates between the inputs generated by (7.9) and (7.13) to acheive large feasibility region of vertex controller, while meeting the desired performance criteria (7.11) sub-optimally.

7.3.1 Interpolation using Linear Programming

Assuming, that we already have the sets $\boldsymbol{\mathcal{C}}_p$, K_p, $\boldsymbol{\mathcal{C}}_v$ and the vertex control law (7.9), (7.10), then any given $x(k) \in \boldsymbol{\mathcal{C}}_v$ can be decomposed as

$$x(k) = \alpha(k)x_v(k) + (1 - \alpha(k))x_p(k), \tag{7.14}$$

where, $x_v(k) \in \boldsymbol{\mathcal{C}}_v$, $x_p(k) \in \boldsymbol{\mathcal{C}}_p$ and $0 \leq \alpha(k) \leq 1$. Note that, for a given $x(k)$ the choice $x_v(k), x_p(k)$ and $\alpha(k)$ can be non-unique. Consider the following control law

$$u(k) = \alpha(k)u_v(k) + (1 - \alpha(k))u_p(k), \tag{7.15}$$

where, $u_v(k)$ is the control input corresponding to $x_v(k)$ obtained using vertex control law (7.9) and $u_p(k)$ is the input corresponding to $x_p(k)$ from (7.13). Using the argument of convexity, it is easy to show that the controller (7.15) is also an invariance inducing controller and stabilizing for the system (7.6) within the set $\boldsymbol{\mathcal{C}}_v$.

Since we want to meet the performance criteria (7.11), it is desirable to have $u(k)$ as close as possible to $u_p(k)$. We can achieve this by minimizing $\alpha(k)$ in the following optimization problem

$$\begin{aligned} \alpha^*(k) \quad &= \underset{\alpha, x_v, x_p}{arg\,min} \qquad \alpha \\ & \quad st. \qquad S_v x_v \leq \mathbf{1}, \\ & \qquad\qquad S_p x_p \leq \mathbf{1}, \\ & \qquad\qquad \alpha x_v + (1-\alpha)x_p = x(k), \\ & \qquad\qquad 0 \leq \alpha \leq 1. \end{aligned} \tag{7.16}$$

Observe that (7.16) is an nonlinear optimization problem, and it also provides $x_v(k)$ and $x_p(k)$ needed for the computation of $u_v(k)$ and $u_p(k)$ in (7.15). We can equivalently write (7.16) as the following linear optimization problem.

$$\begin{aligned} \alpha^*(k) \quad &= \underset{\alpha, r_v}{arg\, min} && \alpha \\ & st. && S_v r_v \leq \alpha \mathbf{1}, \\ & && S_p(x(k) - r_v) \leq (1-\alpha)\mathbf{1}, \\ & && 0 \leq \alpha \leq 1. \end{aligned} \tag{7.17}$$

where, $x_v = r_v/\alpha$ and $x_p(k)$ can be obtained from (7.14). Thus, at each time instant 'k', we obtain $\alpha(k)$, $x_v(k)$ and $x_p(k)$ by solving (7.17), which can be further used to compute the stabilizing control input (7.15). For convenience, we compactly write the IBC controller as

$$u(k) = IBC(\boldsymbol{\mathcal{C}}_p,\, u_p,\, \boldsymbol{\mathcal{C}}_v, u_v, x(k)). \tag{7.18}$$

For uncertain systems, implementing the IBC-based control strategy is computationally cheap. This can be viewed from the fact that major computationally involved procedures are offline (i.e. computation of RCI sets) and online step involves just solving LP, which can also be implemented in explicit form [21]. However, as a tradeoff, the obtained IBC controller is suboptimal in the region $\boldsymbol{\mathcal{C}}_v \setminus \boldsymbol{\mathcal{C}}_p$. We next present design of the controller for considered vehicle motion control problem.

7.4 Proposed Control Architecture

We are now ready to present the controller design and the contracts used between the subsystems to guarantee desired behaviour. To reduce the complexity associated with the system dimension, we propose to decompose the controller design into the lateral and longitudinal controller, as shown in Figure 7.4. For the convenience of presentation, we next explain each controller design sequentially.

7.4.1 Lateral Controller

For the lateral controller design, we consider the lateral dynamics of the system given by (7.1a)-(7.1d). We compactly write them as

$$\begin{bmatrix} \dot{e}_y \\ \ddot{y} \\ \dot{e}_\psi \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & v_x & 0 \\ 0 & \frac{-(C_f+C_r)}{m_v v_x} & 0 & -v_x - \frac{l_f C_f - l_r C_r}{m_v v_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-(l_f C_f - l_r C_r)}{I_z v_x} & 0 & \frac{-(l_f^2 C_f + l_r^2 C_r)}{I_z v_x} \end{bmatrix} \begin{bmatrix} e_y \\ \dot{y} \\ e_\psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{C_f}{m_v} \\ 0 \\ \frac{l_f C_f}{I_z} \end{bmatrix} \delta + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \dot{\psi}_d. \tag{7.19}$$

The lateral dynamics of the vehicle (7.19) depends on the current longitudinal velocity v_x of the vehicle. Hence, to completely decouple it from the longitudinal dynamics, we treat v_x as time-varying uncertainty. We discretize the system (7.19) and rewrite it as

$$z_1(k+1) = A_1(v_x(k))z_1(k) + B_1\delta(k) + E_1\dot{\psi}_d(k), \tag{7.20}$$

where $z_1(k) = [e_y,\, \dot{y},\, e_\psi,\, \dot{\psi}]^T$. Furthermore, the constraints on $z_1(k)$ in (7.4) and (7.5) can be rewritten as

$$\boldsymbol{\mathcal{Z}}_1 \triangleq \{z_1 |\ H_1 z_1 \leq \mathbf{1}\},\ \boldsymbol{\mathcal{U}}_1 \triangleq \{\delta |\ G_1\delta \leq \mathbf{1}\}, \tag{7.21}$$

The bounds on $\dot{\psi}_d$ in (7.20) can be similarly written as

$$\boldsymbol{\mathcal{W}}_1 \triangleq \{\dot{\psi}_d |\ |\dot{\psi}_d| \leq \Upsilon_1\}. \tag{7.22}$$

The considered lateral dynamics (7.20) is uncertain and affected by bounded additive disturbance $\dot{\psi}_d$, for which a robust controller needs to be designed. If the RCI set for the system (7.19) and (7.21) exist, then we can design a robust lateral controller using the IBC control strategy proposed in Section 7.3. However, as highlighted in [108], [109], the RCI set may exist only for small values of Υ_1, which could make the overall design very conservative. It is known that the desired yaw rate is given by $\dot{\psi}_d = v_d\Gamma_d$, where Γ_d is the curvature of the tracked path. Thus, by allowing arbitrary variation of $\dot{\psi}_d$ within the range (7.22), we are assuming that the planner generates path whose curvature can vary arbitrarily (if the velocity v_d is changing slowly), which is not pragmatic.

Usually, the generated reference is widely influenced by road geometry, often modelled as clothoids with bounded curvature and curvature rate [105]. Thus, in line with [108], [109], we assume that the rate of change of $\dot{\psi}_d$ is bounded. We formally present the assumption on the $\dot{\psi}_d$ as

Assumption 1.

$$\dot{\psi}_d(k+1) = \dot{\psi}_d(k) + \Delta\dot{\psi}_d(k), \tag{7.23}$$

$$|\dot{\psi}_d(k)| \leq \Upsilon_1,\ |\Delta\dot{\psi}_d(k)| \leq \gamma_1. \tag{7.24}$$

We call (7.23) and (7.24) as path model, where $\Delta\dot{\psi}_d(k)$ is the variation of $\dot{\psi}_d(k)$ between consecutive sampling time instances. We assume that the trajectory planner generates piece-wise clothoidal (PWC) paths satisfying path model.

Remark 10: Here Υ_1 and γ_1 are the constant parameters whose values are decided based on the driving scenario or the performance requirements (e.g. bounded lateral acceleration or lateral jerk). These parameters act as a contract between the planner and controller, and selected by the designers of the two subsystems in agreement. Once the parameters are set to appropriate values, then the two subsystems can be designed independently.

For controller development, we can now use the extended system (7.19) and (7.23). Even though $\dot{\psi}_d(k)$ is observable, computing RCI set for the extended system is still not straight forward since it is not stabilizable due to (7.23). Hence, similar to [109], we propose using a new path model that is stabilizable by construction while still representing all the trajectories generated by (7.23) and (7.24).

New Path Model

We propose following path model

$$\dot{\psi}_p(k+1) = \alpha_1\dot{\psi}_p(k) + \beta_1 v_1(k),\ v_1(k) \in [-1, 1], \tag{7.25}$$

where $\alpha_1 = \frac{\Upsilon_1 - \epsilon_1}{\Upsilon_1}$, $\beta_1 = \gamma_1 + \epsilon_1$ are the parameters defining the new path model and $\epsilon_1 \in (0,\ \Upsilon_1]$ is the user defined. As shown in [109], the reference trajectory generated by (7.23) and (7.24) can be also generated by (7.25) by selecting $v_1(k)$ appropriately within $[-1,\ 1]$. We can thus select $\dot{\psi}_p(k) = \dot{\psi}_d(k)$ at all the time instances.

Remark 11: Here ϵ_1 can be seen as a parameter to adjust that the maximum rate of change and the peak value of $\dot{\psi}_p$. If $\epsilon_1 = \Upsilon_1$ then $\alpha_1 = 0$

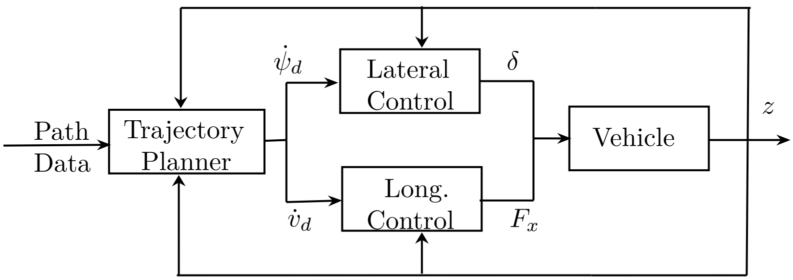

Figure 7.4: Decomposed controller design

and $|\dot{\psi}_p| \leq \gamma_1 + \Upsilon_1$, then (7.25) represents paths with bounded curvature but without curvature rate limitation, which is already known to be conservative. Thus, it is desirable to select $\epsilon_1 \in (0,\ \Upsilon_1]$, which leads to an RCI set with the largest possible volume. This can be done by doing a line search on ϵ_1.

We hence define a new extended system (7.19) and (7.25), which is suitable for designing of IBC controller as below

$$\hat{z}_1(k+1) = \begin{bmatrix} A_1(v(k)) & E_1 \\ 0 & \alpha_1 \end{bmatrix} \hat{z}_1(k) + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} u_1(k) + \begin{bmatrix} 0 \\ \beta_1 \end{bmatrix} v_1(k), \tag{7.26}$$

where $\hat{z}_1(k) = [z_1(k)^T\ \dot{\psi}_p(k)]^T$, $u_1(k) = \delta(k)$. The state and input constraints for the system (7.26), resulting from constraints (7.21) and (7.25) can be rewritten as

$$\hat{\mathcal{Z}}_1 \triangleq \{\hat{z}_1 |\ \hat{H}_1 \hat{z}_1(k) \leq \mathbf{1}\},\ \mathcal{U}_1 \triangleq \{u_1 |\ G_1 u_1(k) \leq \mathbf{1}\}, \qquad v_1(k) \in [-1,\ 1]. \tag{7.27}$$

Now following the approach presented in Section 7.3, we can construct an lateral controller for the vehicle as

$$u^{lat}(k) = IBC(\mathcal{C}_p^{lat},\ u_p^{lat},\ \mathcal{C}_v^{lat}, u_v^{lat}, \hat{z}_1(k)) \tag{7.28}$$

where, $(\mathcal{C}_p^{lat}, u_p^{lat})$ are performance inducing RCI set and controller pair for the system (7.26) and (7.27), and $(\mathcal{C}_v^{lat}, u_v^{lat})$ are RCI set and controller pair providing a large feasibility region.

We next present similar construction for the longitudinal controller.

7.4.2 Longitudinal Controller

We first recall the longitudinal dynamics (7.1e)-(7.1f) of the vehicle, which can be rewritten as

$$\begin{bmatrix} \dot{e}_x \\ \ddot{e}_x \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_x \\ \dot{e}_x \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m_v} \end{bmatrix} F_x + \begin{bmatrix} 0 & 0 \\ \frac{1}{m_v} & -1 \end{bmatrix} \begin{bmatrix} F_d \\ \dot{v}_d \end{bmatrix}. \tag{7.29}$$

where F_d is the combined additive disturbance due to various external forces. The discretized system (7.29) is then expressed as

$$z_2(k+1) = A_2 z_2(k) + B_2 F_x(k) + [E_{21}\ E_{22}] w_2(k), \tag{7.30}$$

where $z_2(k) = [e_x(k)\ \dot{e}_x(k)]^T$ and $w_2(k) = [F_d\ \dot{v}_d]^T$. The longitudinal dynamics depend on the reference $\dot{v}_d$, and hence the controller should be robust to all the possible values $\dot{v}_d \leq \Upsilon_2$. As indicated in many previous studies [110]–[112], for the passengers ride comfort, it is important to bound the vehicle's longitudinal acceleration and jerk. Thus, we make the following assumption on the reference $\dot{v}_d$ generated by the trajectory planner.

Assumption 2.

$$\dot{v}_d(k+1) = \dot{v}_d(k) + \Delta\dot{v}_d(k), \tag{7.31}$$

$$|\dot{v}_d(k)| \leq \Upsilon_2,\ |\Delta\dot{v}_d(k)| \leq \gamma_2. \tag{7.32}$$

Similar to Remark 10, bounds Υ_2 and γ_2 are selected in agreement by the designers of two subsystems and on the comfort requirement. We next present extended system which is suitable for controller design

$$\hat{z}_2(k+1) = \begin{bmatrix} A_2 & E_{22} \\ 0 & \alpha_2 \end{bmatrix} \hat{z}_2(k) + \begin{bmatrix} B_2 \\ 0 \end{bmatrix} u_2(k) + \begin{bmatrix} E_{21} & 0 \\ 0 & \beta_2 \end{bmatrix} \begin{bmatrix} F_d(k) \\ v_2(k) \end{bmatrix}, \tag{7.33}$$

where $\hat{z}_2(k) = [z_2(k),\ \dot{v}_d(k)]^T$. The related constraints can be compactly written as

$$\hat{\mathcal{Z}}_2 \triangleq \{\hat{z}_2 |\ \hat{H}_2 \hat{z}_2(k) \leq \mathbf{1}\},\ \mathcal{U}_2 \triangleq \{u_2 |\ G_2 u_2(k) \leq \mathbf{1}\}, \tag{7.34}$$

and the bounds on the input disturbance in (7.20) can be expressed as

$$\hat{\mathcal{W}}_2 \triangleq \{\hat{w}_2 |\ |D_2 \hat{w}_2| \leq \mathbf{1}\}, \tag{7.35}$$

where $\hat{w}_2(k) = [F_d(k)\ v_2(k)]^T$ and $v_2(k) \in [-1,\ 1]$. The obtained longitudinal model (7.33), and the constraints (7.34) and (7.35) are now used for constructing the IBC controller defined as

$$u^{lon}(k) = IBC(\mathcal{C}_p^{lon},\ u_p^{lon},\ \mathcal{C}_v^{lon}, u_v^{lon}, \hat{z}_2(k)), \tag{7.36}$$

where $(\mathcal{C}_p^{lon}, u_p^{lon})$ are performance inducing RCI set and controller pair for the system (7.33) and (7.34), and $(\mathcal{C}_v^{lon}, u_v^{lon})$ are RCI set and controller pair providing a large feasibility region.

In summary, we presented an IBC control strategy (7.28) and (7.36) for the autonomous motion control of the vehicle. The controllers guarantee bounded tracking error provided the reference trajectories generated by the trajectory planner satisfies Assumption 1 and Assumption 2, and thus meets all the design objectives stated in Section 7.2.3.

Remark 12: For the construction of the trajectory planner, we can use MPC controller with the kinematic model (7.42) and the inputs $\ddot{v}_d$ and $\ddot{\psi}_d$. Thus, for given waypoints $(X_r(k), Y_r(k))$, the MPC controller should compute trajectories of $\dot{\psi}_d$ and $\dot{v}_d$ satisfying Assumption 1 and Assumption 2, which is then provided to the designed controller for tracking.

7.4.3 Some Practical Issues with the Design

The proposed controller guarantees constraint satisfaction only for the fixed values of parameters in Table 7.2. In reality, many of the parameters can vary with time, e.g., the mass of the vehicle changes with the number of passengers, amount of fuel and additional loads. Thus, the impact of the parameter variations has to investigate thoroughly before assuming the controller to be safe for the implementation. Furthermore, in the controller design, we should use a more comprehensive system model that includes the actuator dynamics to ensure constraint satisfaction. However, then computing RCI sets for the comprehensive system can be challenging.

We next demonstrate the effectiveness of the proposed strategy using closed-loop simulation with the nonlinear vehicle model.

7.5 Simulations

We performed the simulations in MATLAB using vehicle constraints and parameters presented in Table 7.1 and Table 7.2 with the sampling time $T_s = 1/40s$. Based on the safety and comfort requirements, we need to provide the bounds $\Upsilon_1, \Upsilon_2, \gamma_1$ and γ_2 in the Assumption 1 and Assumption 2 to the trajectory planner to limit the tracked reference trajectories. We assume that the tracked trajectory $\dot{\psi}_d(k)$ satisfies (7.23) and (7.24) with bounds

$$\Upsilon_1 = 0.2160 \text{ and } \gamma_1 = 0.0108. \tag{7.37}$$

Since $50/3.6 \leq v_d \leq 100/3.6$, with the specification (7.37), the trajectory planner is allowed to generate reference PWC trajectories which represents paths with a minimum radius $64.3m$ at a constant velocity of $50\,Km/hr$ and $128.6m$ at a constant velocity of $100\,Km/hr$. The bounds on $\dot{\psi}_d(k)$ can be also decided based on the passenger comfort as the lateral acceleration of the vehicle $a_y \approx \dot{\psi}_d v_x$. Thus, we can bound lateral acceleration by bounding $\dot{\psi}_d(k)$.

Similarly, we bound the longitudinal acceleration $|\dot{v}_d| \leq 1.5\,m/s^2$ and jerk $|\ddot{v}_d| \leq 0.7\,m/s^3$ for passenger comfort by restricting trajectory planner to generate acceleration profile satisfying (7.31) and (7.32) with bounds

$$\Upsilon_2 = 1.5 \text{ and } \gamma_2 = 0.7. \tag{7.38}$$

The specification (7.38) are inline with the bounds recommended in [110]–[112]. We thus, construct the proposed lateral controller (7.28) and longitudinal controller (7.36) using the extended system (7.26) and (7.33), where $(\alpha_1, \beta_1, \epsilon_1) = (0.8148, 0.0508, 0.04)$ and $(\alpha_2, \beta_2, \epsilon_2) = (0.9333, 0.1175, 0.1)$. We considered following performance constraints while designing the controllers

$$J_{lat} = \sum_{k=0}^{\infty} \hat{z}_1(k)^T \hat{z}_1(k) + 0.1u_1(k)^2 \leq 100. \tag{7.39}$$

$$J_{lon} = \sum_{k=0}^{\infty} \hat{z}_2(k)^T \hat{z}_2(k) + 0.1u_2(k)^2 \leq 100. \tag{7.40}$$

We perform close-loop simulation in the following scenerio.

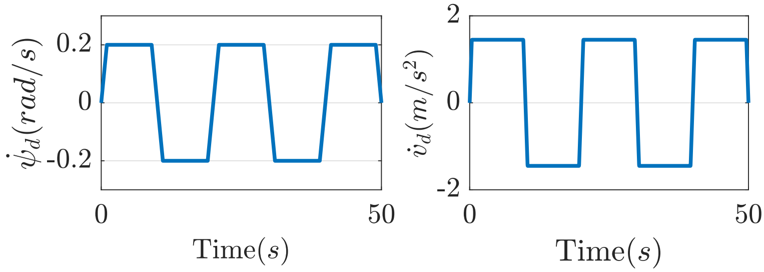

Figure 7.5: Reference trajectories from trajectory planner system.

Time Varying Longitudinal velocity

To test the designed controller, we assume that the trajectory planner generates references with the desired speed v_d and the yaw-rate $\dot{\psi}_d$ varying linearly between the peak values. The input reference trajectories of $\dot{v}_d$ and $\dot{\psi}_d$, satisfying Assumption 2 and Assumption 1 by construction, are shown in Figure 7.5. In the simulation, we assume the initial longitudinal velocity v_d to be $50\,Km/hr$ and it varies linearly between $50-100\,Km/hr$ based on the desired acceleration input, as shown in Figure 7.8. It can be seen that the desired reference trajectory excites the peak yaw-rate and the longitudinal velocity, which are treated as uncertainties in the lateral dynamics of the vehicle. In the Figure 7.9, we show the global reference positions of the vehicle obtained by using the kinematic relation (7.42) between (X, Y) and input reference trajectories (v_d, ψ_d), with initial $(X, Y) = (0, 0)$.

We then perform the closed-loop simulations using the designed controllers with the continuous time nonlinear system (7.41). We show the state trajectories of the system (7.1) in Figure 7.6 and the corresponding control input trajectory in Figure 7.7. It can viewed that the lateral error e_x and longitudinal error e_y are within the selected bounds (7.3), which shows that the proposed controllers are safe by design.

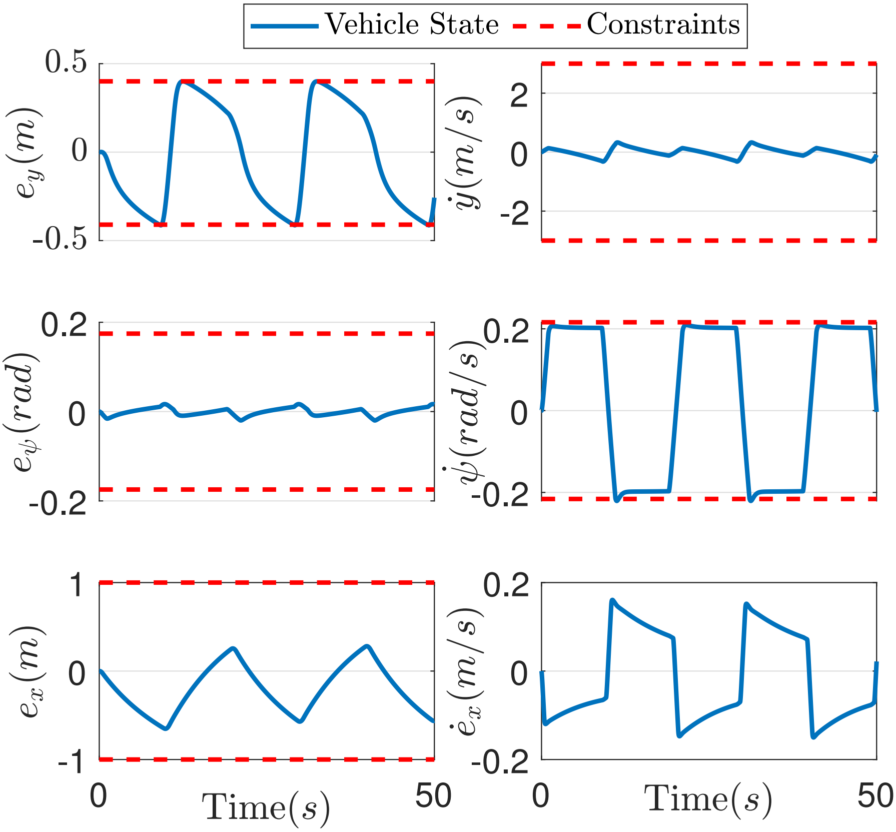

Figure 7.6: Measured vehicle states.

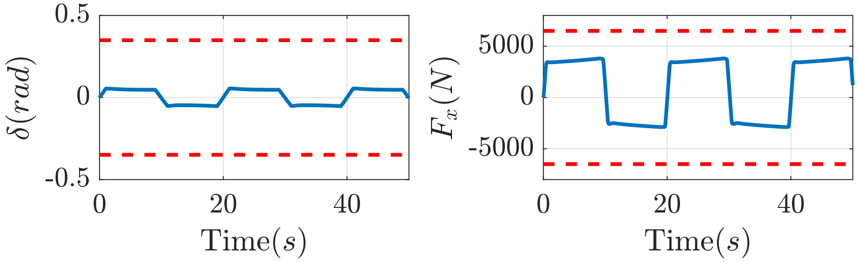

Figure 7.7: Vehicle control inputs.

Lastly, the overall tracking performance can be seen in the Figure 7.8 and Figure 7.9.

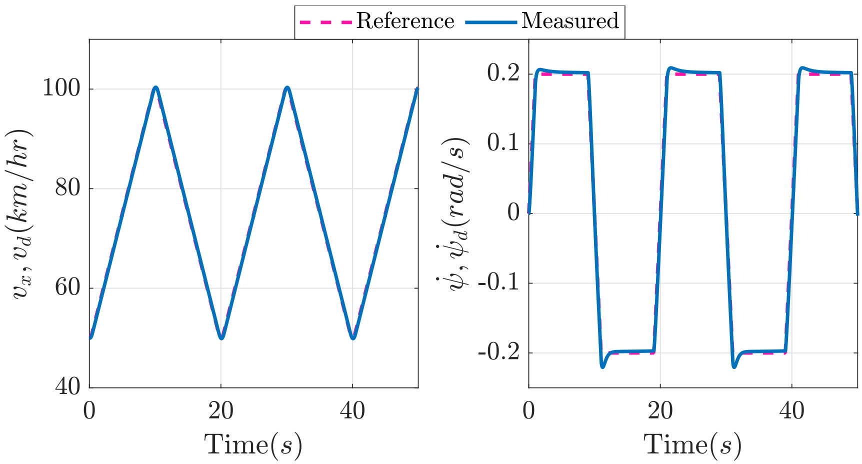

Figure 7.8: Reference and measured longitudinal velocity and yaw rate of the vehicle.

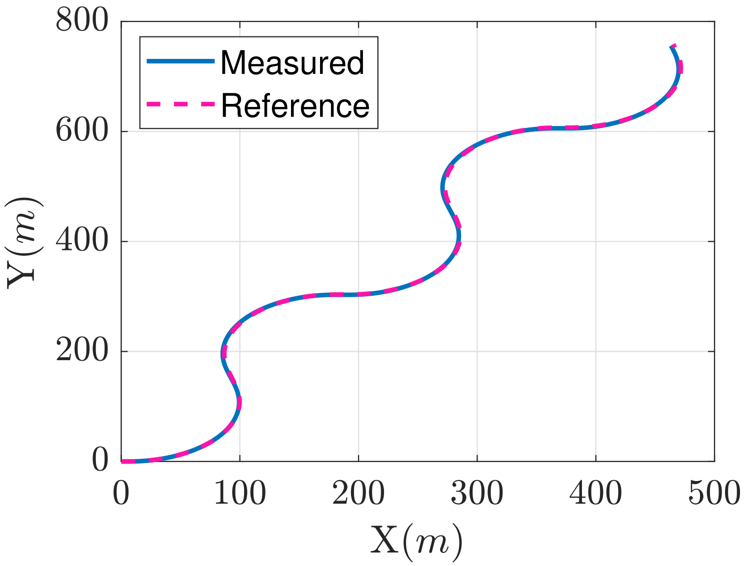

Figure 7.9: Reference and measured global positions of the vehicle.

7.A Appendix

Dynamical Model

Vehicle nonlinear dynamical model

$$\dot{y} = v_y, \tag{7.41a}$$

$$\dot{v}_y = -v_x\dot{\psi} + \frac{1}{m_v}(F_{y_f} + F_{y_r}), \tag{7.41b}$$

$$\dot{\psi} = \dot{\psi}, \tag{7.41c}$$

$$\ddot{\psi} = \frac{1}{I_z}(l_f F_{y_f} - l_r F_{y_r}), \tag{7.41d}$$

$$\dot{x} = v_x, \tag{7.41e}$$

$$\dot{v}_x = \frac{1}{m_v}(F_x - F_{drag} - F_{rolling} - mgsin(\theta)). \tag{7.41f}$$

Kinematic model

Kinematic model of the vehicle.

$$\dot{X} = v_d cos(\psi_d), \tag{7.42a}$$

$$\dot{Y} = v_d sin(\psi_d), \tag{7.42b}$$

$$\ddot{v}_d = \ddot{v}_d \tag{7.42c}$$

$$\ddot{\psi}_d = \ddot{\psi}_d. \tag{7.42d}$$

Part IV

Conclusion

CHAPTER 8

Concluding Remarks and Future Work

In the thesis, we have developed a number of iterative LMI-based algorithms to compute desirably large (or small) RCI sets of desired complexity together with associated state-feedback gains. The algorithms apply to systems with rational dependence on (possibly time-varying) uncertainties, while the computed feedback law can be simple state feedback or piecewise affine control law. Sufficient LMI conditions guaranteeing invariance have been derived thanks to state transformation, full block S-procedure, Polya's relaxations and successive linearization. Furthermore, these algorithms can generate RCI sets where a desired quadratic performance can be insured; this can be useful in the control strategies like MPC to guarantee performance in the terminal region. The capabilities of these algorithms are demonstrated by including many numerical comparisons where the proposed algorithm outperforms many similar approaches.

We also develop a similar algorithm in the case when time-varying parameters are observable. As a difference, we allow the RCI set as well the invariance inducing controller to be parameter-dependent to exploit the available information of the scheduling parameters. As an outcome, we have numerically shown that the presented iterative algorithm can generate invariant sets which are larger than the maximal RCI sets computed without exploiting scheduling parameter information. We have presented some new methods for volume maximization and minimization of a 0-symmetric polytope as an additional contribution.

In the end, we demonstrated how we could combine the solutions obtained from some on these algorithms to construct a vehicle motion controllers, which guarantees the safety and performance by design. The main advantage of the presented control strategy over traditional MPC schemes is its online computational complexity which is cheap. Furthermore, it is also possible to construct the controller in the offline form, facilitating extensive verification of the controllers before implementation. Such a feature can help in achieving the strict requirements associated with new safety standards like ISO26262.

As expected, there are many possibilities for improvements and extensions to the presented work. In the next sections, we will try to highlight some future directions for the research.

8.1 Extensions and Possible Future Directions for Research

Choice of the initial Matrix P

It was observed in the numerical exercises that the choice of initial set affects the final outcome. Thus to obtain the optimal RCI sets, the initial polytope should be selected appropriately, which is not the focus of the proposed heuristics in the **Remark** 4. In the worst case, initializing the proposed algorithm using the heuristic may lead to infeasibility, thereby encouraging investigations to identify an appropriate approach to select this initial set.

A straightforward extension, which could potentially relax the requirement on the choice of matrix P is to modify the presented approach to compute so-called quasi or extended invariant sets [113]. As a drawback, we could end up increasing the overall computational complexity of the proposed procedure.

Nonzero Symmetric RCI sets

In the thesis, we assumed candidate RCI set to be 0-symmetric. This is a logical assumption if the system constraints are 0-symmetric as well, and the controlled system is linear. In the other case, this assumption could be potentially conservative. Thus, in the extension to the presented work, we can allow the candidate RCI set description to be symmetric around some point x_c as follows

$$\mathcal{C} = \{x| - \mathbf{1} \leq PW^{-1}(x - x_c) \leq \mathbf{1}\}. \tag{8.1}$$

Note that x_c is allowed to be nonzero and should be computed by the algorithm along with the matrix P, W. Formulating similar algorithms, which computes general asymmetric polytopic RCI sets could also be a possible direction for research; however, this may not be straight forward.

High Dimensional Systems

As with almost all the literature methods [30], computing RCI sets for the high dimensional system is an issue with the proposed algorithms as well. It was evident when we unsuccessfully tried to calculate the RCI set for the complete vehicle dynamics in Chapter 7. Even though we propose algorithms which solve a convex optimization problem to compute an RCI set, they can be computationally very demanding when applied to high dimensional systems. As indicated in the Remark 7 and in Section 6.5.3, the computational complexity of the algorithms directly depend on the system dimension.

Thus an important direction of research would be to come up with methods which are capable of generating RCI sets for high dimensional systems.

General Set Constraints

In many applications, we may have systems which are subjected to non-polytopic constraints. Note that it is straightforward to extend the proposed algorithms to consider ellipsoidal system constraints. This can be achieved by the application of the S-Procedure (**Lemma** 3). However, it may not be easy to extend the presented algorithms for other convex non-polytopic sets in general. Of course, a conservative but straightforward alternative would be to replace such constraints by polytopic one.

Output Feedback

When computing control invariant sets, it is usually assumed that the invariance inducing controller is state feedback, which in turn requires states to be directly measurable. However, in practical applications, we commonly have partial state information, and thus need to use observer (assuming the system is observable) to get complete state estimate. It is known that the state estimates have estimation errors. Hence the actual system trajectories may violate the RCI sets if the estimates are used as feedback to the controller.

An approach to compute RCI set and the feedback controller, which is robust against estimation errors, is to pre-compute the observer and then utilize the extended system to compute invariant set (see, [114], [115]). Note that we can use the proposed algorithms in the thesis for the purpose. With this approach, it is not clear how these observers should be designed such that the obtained RCI sets are least conservative. Thus, an interesting direction for research would be to develop methods that select the observer and the controller gains optimizing the RCI set volume.

Similar to [116], [117], another direction of research could be to propose methods which directly computes an output feedback controller that optimize the RCI set volume.

Data-Driven Approach

Most of the methods in the literature for computing RCI sets and all our presented algorithms need a dynamic model of the real system to characterize the system time evolution, which may not always be available. Towards this, a standard approach is to identify the dynamic model of the black (or grey) box system based on some available input-output data. Since the available data are typically assumed to be affected by noise, one can use a set membership identification approach [118] to come up with a model with bounded uncertainties that best fits the data. Then by using proposed algorithms, one can compute RCI sets. As shown in [119], model which optimally fits the input-output data may not be the best for computing RCI sets. Thus, a method needs to be devised in the future, identifying the best model for computing RCI sets. In the control literature, these problems are well known as identification for control.

Recently, there has been some work done in the area of a data-driven approach, which directly computes invariant sets from the available input and the state data [120], [121]. The main advantage of these approaches is that we do not have to identify the best (or any) system model. However, the assumption that the full state information (instead of output) is available in the form of the data point, in itself can be a major limiting factor for these methods. Furthermore, the application of these methods is limited to simple linear systems or autonomous system. Thus motivates us to explore more in the direction of data-driven approaches for more general system description and less restrictive assumption.

References

[1] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Boston, MA: Birkhäuser, 2015.

[2] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.

[3] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 7, no. 11, pp. 1163–1173, 1983.

[4] E. D. Sontag, "A 'universal' construction of artstein's theorem on nonlinear stabilization," *Systems & Control Letters*, vol. 13, no. 2, pp. 117–123, 1989.

[5] J. P. LaSalle, *The stability and control of discrete processes*. Springer Science & Business Media, 2012, vol. 62.

[6] J.-P. Aubin, A. M. Bayen, and P. Saint-Pierre, *Viability theory: new directions*. Springer Science & Business Media, 2011.

[7] S. Munir, M. Hovd, and S. Olaru, "Low complexity constrained control using higher degree lyapunov functions," *Automatica*, vol. 98, pp. 215–222, 2018.

[8] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.

[9] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, Nov. 1999.

[10] Z. Artstein and S. V. Raković, "Feedback and invariance under uncertainty via set-iterates," *Automatica*, vol. 44, no. 2, pp. 520–525, 2008.

[11] S. Raković and M. Baric, "Parameterized Robust Control Invariant Sets for Linear Systems: Theoretical Advances and Computational Remarks," *IEEE Transactions on Automatic Control*, vol. 55, no. 7, pp. 1599–1614, Jul. 2010.

[12] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[13] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[14] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[15] D. Bertsekas and I. Rhodes, "On the minimax reachability of target sets and target tubes," *Automatica*, vol. 7, no. 2, pp. 233–247, Mar. 1971.

[16] W. Langson, I. Chryssochoos, S. Raković, and D. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, Jan. 2004.

[17] D. Q. Mayne and E. C. Kerrigan, "Tube-based robust nonlinear model predictive control1," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 36–41, 2007, IFAC Symposium on Nonlinear Control Systems.

[18] P. 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*, vol. 31, no. 5, pp. 457–459, May 1986.

[19] H. Nguyen, S. Olaru, P. Gutman, and M. Hovd, "Constrained interpolation-based control for polytopic uncertain systems," in *Conference on Decision and Control and European Control Conference*, 2011, pp. 4961–4966.

[20] H.-N. Nguyen, "Constrained control of uncertain, time-varying, discrete-time systems: An interpolation-based approach," in. Cham: Springer International Publishing, 2014.

[21] H.-N. Nguyen, P.-O. Gutman, S. Olaru, and M. Hovd, "Explicit improved vertex control for uncertain, time-varying linear discrete-time systems with state and control constraints," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 12, pp. 2652–2667, 2016.

[22] P. Mercader, D. Rubin, H.-N. Nguyen, A. Bemporad, and P.-O. Gutman, "Simple interpolating control," *IFAC-PapersOnLine*, vol. 51, no. 25, pp. 42–47, 2018, IFAC Symposium on Robust Control Design(ROCOND).

[23] E. Gilbert and I. Kolmanovsky, "Fast reference governors for systems with state and control constraints and disturbance inputs," *International Journal of Robust and Nonlinear Control*, vol. 9, no. 15, pp. 1117–1141, 1999.

[24] Emanuele Garone, S. D. Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, 2017, ISSN: 0005-1098.

[25] A. Colombo, M. Bahraini, and P. Falcone, "Measurement scheduling for control invariance in networked control systems," in *Conference on Decision and Control*, 2018, pp. 3361–3366.

[26] 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*, vol. 24, no. 4, pp. 1294–1307, 2016.

[27] J.-P. Aubin, "A survey of viability theory," *SIAM Journal on Control and Optimization*, vol. 28, no. 4, pp. 749–788, 1990.

[28] E. Gilbert and K. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[29] C. E. T. Dórea and J. C. Hennet, "(A, B)-Invariant Polyhedral Sets of Linear Discrete-Time Systems," *Journal of Optimization Theory and Application*, vol. 103, no. 3, pp. 521–542, Dec. 1999.

[30] M. Fiacchini and M. Alamir, "Computing control invariant sets is easy," *CoRR*, vol. abs/1708.04797, 2017, http://arxiv.org/abs/1708.04797.

[31] E. Klintberg, M. Nilsson, L. J. Mårdh, and A. Gupta, "A primal active-set minimal-representation algorithm for polytopes with application to invariant-set calculations," *Conference on Decision and Control*, pp. 6862–6867, 2018.

[32] M. Rungger and P. Tabuada, "Computing robust controlled invariant sets of linear systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3665–3670, Jul. 2017.

[33] S. Miani and C. Savorgnan, "Maxis-g: A software package for computing polyhedral invariant sets for constrained lpv systems," in *Conference on Decision and Control*, 2005, pp. 7609–7614.

[34] E. Klintberg, M. Nilsson, A. Gupta, L. J. Mårdh, and P. Falcone, "Tree-structured polyhedral invariant set calculations," *IEEE Control Systems Letters*, vol. 4, no. 2, pp. 426–431, 2020.

[35] B. Pluymers, J. A. Rossiter, J. A. K. Suykens, and B. D. Moor, "The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty," *American Control Conference*, pp. 804–809, 2005.

[36] S. Raković, E. Kerrigan, K. Kouramas, and D. Mayne, "Invariant approximations of the minimal robust positively Invariant set," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 406–410, Mar. 2005.

[37] C. Wen-Hua, D. Ballance, and J. O'Reilly, "Optimisation of attraction domains of nonlinear MPC via LMI methods," *American Control Conference*, pp. 3067–3072, 2001.

[38] T. Alamo, A. Cepeda, and D. Limon, "Improved computation of ellipsoidal invariant sets for saturated control systems," *Conference on Decision and Control*, pp. 6216–6221, 2005.

[39] S. Tarbouriech, G. Garcia, J. M. G. da Silva Jr, and I. Queinnec, *Stability and stabilization of linear systems with saturating actuators*. Springer Science & Business Media, 2011.

[40] B. D. O'Dell and E. A. Misawa, "Semi-Ellipsoidal Controlled Invariant Sets for Constrained Linear Systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 124, no. 1, pp. 98–103, Apr. 2000.

[41] B. Legat, S. V. Raković, and R. M. Jungers, "Piecewise semi-ellipsoidal control invariant sets," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 755–760, 2021.

[42] T. Blanco, M. Cannon, and B. De Moor, "On efficient computation of low-complexity controlled invariant sets for uncertain linear systems," *International Journal of Control*, vol. 83, no. 7, pp. 1339–1346, Jul. 2010.

[43] B. Pluymers, J. A. Rossiter, J. A. K. Suykens, and B. D. Moor, "Interpolation based MPC for LPV systems using polyhedral invariant sets," *American Control Conference*, pp. 810–815, 2005.

[44] F. Blanchini, S. Miani, and C. Savorgnan, "Dynamic augmentation and complexity reduction of set-based constrained control," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 14 324–14 329, 2008.

[45] N. Athanasopoulos, G. Bitsoris, and M. Lazar, "Construction of invariant polytopic sets with specified complexity," *International Journal of Control*, vol. 87, no. 8, pp. 1681–1693, Aug. 2014.

[46] M. Cannon, V. Deshmukh, and B. Kouvaritakis, "Nonlinear model predictive control with polytopic invariant sets," *Automatica*, vol. 39, no. 8, pp. 1487–1494, Aug. 2003.

[47] F. Tahir and I. Jaimoukha, "Low-Complexity Polytopic Invariant Sets for Linear Systems Subject to Norm-Bounded Uncertainty," *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1416–1421, May 2015.

[48] P. Trodden, "A One-Step Approach to Computing a Polytopic Robust Positively Invariant Set," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4100–4105, Dec. 2016.

[49] C. Liu, F. Tahir, and I. Jaimoukha, "Full-complexity polytopic robust control invariant sets for uncertain linear discrete-time systems," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 11, pp. 3587–3605, 2019.

[50] M. Fiacchini, T. Alamo, and E. Camacho, "On the computation of convex robust control invariant sets for nonlinear systems," *Automatica*, vol. 46, no. 8, pp. 1334–1338, 2010.

[51] M. A. Ben Sassi and A. Girard, "Controller synthesis for robust invariance of polynomial dynamical systems using linear programming," *Systems & Control Letters*, vol. 61, no. 4, pp. 506–512, 2012.

[52] M. Korda, D. Henrion, and C. N. Jones, "Convex computation of the maximum controlled invariant set for polynomial control systems," *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 2944–2969, 2014.

[53] I. Kolmanovsky and E. G. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Mathematical Problems in Engineering*, vol. 4, 1998.

[54] A. Gupta, H. Köroğlu, and P. Falcone, "Restricted-complexity characterization of control-invariant domains with application to lateral vehicle dynamics control," in *Conference on Decision and Control (CDC)*, 2017, pp. 4946–4951.

[55] A. Gupta and P. Falcone, "Full-complexity characterization of control-invariant domains for systems with uncertain parameter dependence," *IEEE Control System Letter*, vol. 3, no. 1, pp. 19–24, 2019.

[56] A. Gupta, H. Köroğlu, and P. Falcone, "Computation of low-complexity control-invariant sets for systems with uncertain parameter dependence," *Automatica*, vol. 101, pp. 330–337, 2019.

[57] ——, "Computation of robust control invariant sets with predefined complexity for uncertain systems," *International Journal of Robust and Nonlinear Control*, 2020, `https://doi.org/10.1002/rnc.5378`.

[58] A. Gupta, M. Mejari, P. Falcone, and D. Piga, *Computation of parameter dependent robust invariant sets for lpv models with guaranteed performance*, 2020.

[59] C. Scherer and S. Weiland, "Linear matrix inequalities in control," *Lecture Notes, Dutch Institute for Systems and Control*, 2015.

[60] J. Mohammadpour and C. W. Scherer, *Control of linear parameter varying systems with applications.* Springer Science & Business Media, 2012.

[61] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory.* Society for Industrial and Applied Mathematics, 1994.

[62] C. Briat, "Linear parameter-varying and time-delay systems," *Analysis, observation, filtering & control*, vol. 3, pp. 5–7, 2014.

[63] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming.* Society for Industrial and Applied Mathematics, 1994.

[64] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999, Version 1.05 `http://fewcal.kub.nl/sturm`.

[65] K. T. R.H Tutuncu and M. Todd, "Solving semidefinite-quadratic-linear programs using sdpt3," *Mathematical Programming Ser. B*, vol. 95, pp. 189–217, 2003.

[66] M. ApS, "The mosek optimization toolbox for matlab manual. version 9.0.," *http://docs.mosek.com/9.0/toolbox/index.html*, 2019.

[67] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *Computer-Aided Control System Design Conference*, Taipei, Taiwan, 2004.

[68] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming, version 2.1*, `http://cvxr.com/cvx`, Mar. 2014.

[69] M. C. de Oliveira and R. E. Skelton, "On stability tests for linear systems," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 125–130, 2002.

[70] P. Pinsler, "Über das vorkommen definiter und semidefiniter formen in scharen quadratischer formen," *Commentarii Mathematici Helvetici*, vol. 9, no. 1, pp. 188–192, 1936.

[71] J. Y. Ishihara, H. T. M. Kussaba, and R. A. Borges, "Existence of continuous or constant finsler's variables for parameter-dependent systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4187–4193, 2017.

[72] I. Pólik and T. Terlaky, "A survey of the s-lemma," *SIAM Review*, vol. 49, no. 3, pp. 371–418, 2007.

[73] C. Scherer, "LMI Relaxations in Robust Control," *European Journal of Control*, vol. 12, no. 1, pp. 3–29, Jan. 2006.

[74] R. C. Oliveira and P. L. Peres, "Stability of polytopes of matrices via affine parameter-dependent lyapunov functions: Asymptotically exact lmi conditions," *Linear Algebra and its Applications*, vol. 405, pp. 209–228, 2005.

[75] C. W. Scherer, "Relaxations for robust linear matrix inequality problems with verifications for exactness," *SIAM Journal on Matrix Analysis and Applications*, vol. 27, no. 2, pp. 365–395, 2005.

[76] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities, second ed.* Cambridge University Press, 1952.

[77] C. Scherer, "A Full Block S-Procedure with applications," *Conference on Decision and Control*, pp. 2602–2607, 1997.

[78] H. Köroğlu and C. Scherer, "Robust Stability Analysis against Perturbations of Smoothly Time-Varying Parameters," *Conference on Decision and Control*, pp. 2895–2900, 2006.

[79] J. Biannic and C. Roos, "Generalized state space: A new matlab class to model uncertain and nonlinear systems as linear fractional representations," *Available with the SMAC toolbox at http://w3. onera. fr/smac/gss*, 2016.

[80] C. Liu and I. Jaimoukha, "The computation of full-complexity polytopic robust control invariant sets," *Conference on Decision and Control*, pp. 6233–6238, Dec. 2015.

[81] L. Vandenberghe, S. Boyd, and S. Wu, "Determinant Maximization with Linear Matrix Inequality Constraints," *SIAM Journal on Matrix Analysis and Application*, vol. 19, no. 2, pp. 499–533, Apr. 1998.

[82] P. Falcone, H. E. Tseng, F. Borrelli, J. Asgari, and D. Hrovat, "Mpc-based yaw and lateral stabilisation via active front steering and braking," *Vehicle Systems Dynamics*, vol. 46, pp. 611–628, 2009.

[83] S. Munir, M. Hovd, G. Sandou, and S. Olaru, "Controlled contractive sets for low-complexity constrained control," in *IEEE Conference on Computer Aided Control System Design (CACSD)*, 2016, pp. 856–861.

[84] M. Lazar and R. Gielen, "On parameterized lyapunov and control lyapunov functions for discrete-time systems," in *IEEE Conference on Decision and Control*, 2010, pp. 3264–3270.

[85] S. Raković, E. Kerrigan, D. Mayne, and K. Kouramas, "Optimized robust control invariance for linear discrete-time systems: Theoretical foundations," *Automatica*, vol. 43, no. 5, pp. 831–841, 2007.

[86] F. Blanchini, "Constrained control for uncertain linear systems," *Journal of Optimization Theory and Applications*, vol. 71, pp. 465–484, 1991.

[87] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *European Control Conference*, `http://control.ee.ethz.ch/~mpt`, Zürich, Switzerland, Jul. 2013, pp. 502–510.

[88] P. d'Alessandro and E. De Santis, "Controlled invariance and feedback laws," *IEEE Transactions on Automatic Control*, vol. 46, no. 7, pp. 1141–1146, 2001.

[89] P. Gutman and M. Cwikel, "Admissible sets and feedback control for discrete-time linear dynamical systems with bounded controls and states," *IEEE Transactions on Automatic Control*, vol. 31, no. 4, pp. 373–376, 1986.

[90] F. Blanchini, "Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions," *IEEE Transactions on Automatic Control*, vol. 39, no. 2, pp. 428–433, Feb. 1994.

[91] B. Bueler, A. Enge, and K. Fukuda, *Exact volume computation for polytopes: a practical study*. DMV SEMINAR, Springer, 2000, vol. 29, pp. 131–154.

[92] J. Hanema, R. Tóth, and M. Lazar, "Heterogeneously parameterized tube model predictive control for lpv systems," *Automatica*, vol. 111, p. 108 622, 2020.

[93] F. Blanchini, S. Miani, and C. Savorgnan, "Stability results for linear parameter varying and switching systems," *Automatica*, vol. 43, no. 10, pp. 1817–1823, 2007.

[94] M. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[95] J. Hanema, R. Tóth, and M. Lazar, "Stabilizing non-linear mpc using linear parameter-varying representations," in *Conference on Decision and Control*, 2017, pp. 3582–3587.

[96] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science, 2004.

[97] M. Dyer and A. Frieze, "On the complexity of computing the volume of a polyhedron," *SIAM Journal on Computing*, vol. 17, pp. 967–974, 1988.

[98] A. Benavoli and D. Piga, "A probabilistic interpretation of set-membership filtering: Application to polynomial systems through polytopic bounding," *Automatica*, vol. 70, pp. 158–172, 2016.

[99] D. Piga and A. Benavoli, "Semialgebraic outer approximations for set-valued nonlinear filtering," in *18th European Control Conference (ECC)*, Naples, Italy, Jul. 2019, pp. 400–405.

[100] L. Ni, A. Gupta, P. Falcone, and L. Johannesson, "Vehicle lateral motion control with performance and safety guarantees," *Advances in automotive control IFAC*, vol. 49, pp. 285–290, 2016.

[101] A. Gupta and P. Falcone, "Low-complexity explicit mpc controller for vehicle lateral motion control," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2839–2844.

[102] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, "Taming dr. frankenstein: Contract-based design for cyber-physical systems*," *European Journal of Control*, vol. 18, no. 3, pp. 217–238, 2012.

[103] S. W. Smith, P. Nilsson, and N. Ozay, "Interdependence quantification for compositional control synthesis with an application in vehicle safety systems," in *IEEE Conference on Decision and Control (CDC)*, 2016, pp. 5700–5707.

[104] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, "Correctness guarantees for the composition of lane keeping and adaptive cruise control," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1216–1229, 2018.

[105] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[106] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on control systems technology*, vol. 15, no. 3, pp. 566–580, 2007.

[107] Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles," *Vehicle System Dynamics*, vol. 52, no. 6, pp. 802–823, 2014.

[108] S. Di Cairano, U. V. Kalabić, and K. Berntorp, "Vehicle tracking control on piecewise-clothoidal trajectories by mpc with guaranteed error bounds," in *Conference on Decision and Control (CDC)*, 2016, pp. 709–714.

[109] A. Gupta, M. Nilsson, P. Falcone, E. Klintberg, and L. J. Mårdh, "A framework for vehicle lateral motion control with guaranteed tracking and performance," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3607–3612.

[110] I. Bae, J. Moon, and J. Seo, "Toward a comfortable driving experience for a self-driving shuttle bus," *Electronics*, vol. 8, no. 9, 2019.

[111] M. Elbanhawi, M. Simic, and R. Jazar, "In the passenger seat: Investigating ride comfort measures in autonomous cars," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 3, pp. 4–17, 2015.

[112] L. L. Hoberock, "A Survey of Longitudinal Acceleration Comfort Studies in Ground Transportation Vehicles," *Journal of Dynamic Systems, Measurement, and Control*, vol. 99, no. 2, pp. 76–84, Jun. 1977.

[113] Y. Lee, M. Cannon, and B. Kouvaritakis, "Extended invariance and its use in model predictive control," *Automatica*, vol. 41, no. 12, pp. 2163–2169, 2005.

[114] Y. I. Lee and B. Kouvaritakis, "Receding horizon output feedback control for linear systems with input saturation," *IEE Proceedings - Control Theory and Applications*, vol. 148, no. 2, pp. 109–115, 2001.

[115] D. Mayne, S. Raković, R. Findeisen, and F. Allgöwer, "Robust output feedback model predictive control of constrained linear systems," *Automatica*, vol. 42, no. 7, pp. 1217–1222, 2006.

[116] C. E. T. Dórea, "Output-feedback controlled-invariant polyhedra for constrained linear systems," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 5317–5322.

[117] T. A. D. Almeida and C. E. T. Dorea, “Output feedback constrained regulation of linear systems via controlled-invariant sets,” *IEEE Transactions on Automatic Control*, pp. 1–1, 2020.

[118] R. L. Kosut, M. K. Lau, and S. P. Boyd, “Set-membership identification of systems with parametric and nonparametric uncertainty,” *IEEE Transactions on Automatic Control*, vol. 37, no. 7, pp. 929–941, 1992.

[119] Y. Chen, H. Peng, J. Grizzle, and N. Ozay, “Data-driven computation of minimal robust control invariant set,” in *IEEE Conference on Decision and Control (CDC)*, 2018, pp. 4052–4058.

[120] Z. Wang and R. M. Jungers, “Scenario-based set invariance verification for black-box nonlinear systems,” *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 193–198, 2021.

[121] A. Bisoffi, C. D. Persis, and P. Tesi, *Data-based guarantees of set invariance properties*, 2019.