IMT School for Advanced Studies, Lucca
Lucca, Italy

Invariant Set-based Methods for the Computation of Input and Disturbance Sets

PhD Program in Systems Science

Track in Computer Science and Systems Engineering

XXXIII Cycle

By

Sampath Kumar Mulagaleti

2023

The dissertation of Sampath Kumar Mulagaleti is approved.

PhD Program Coordinator: Prof. Rocco De Nicola, IMT School for Advanced Studies Lucca

Advisor: Prof. Alberto Bemporad, IMT School for Advanced Studies Lucca

Co-Advisor: Prof. Mario Zanon, IMT School for Advanced Studies Lucca

The dissertation of Sampath Kumar Mulagaleti has been reviewed by:

Prof. Ilya Kolmanovsky, University of Michigan, Ann Arbor, MI

Prof. Necmiye Özay, University of Michigan, Ann Arbor, MI

IMT School for Advanced Studies Lucca
2023

One small step for man, one giant crawl for a worm.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my sincere gratitude to Prof. Alberto Bemporad for having given me the opportunity to pursue a PhD. He has been an inexhaustible source of inspiration and ideas. This journey was supported with utmost patience by Prof. Mario Zanon. Discussions with him are the most stimulating, and his responses always help reveal deeper insights regarding the topic at hand. His approach to attempt at understanding things by breaking them down, and not limited to technical matters, is a source of inspiration. I also thank him heartily for the encouragement and trust, which has been very impactful towards retaining my motivation. I would like to sincerely thank Prof. Ilya Kolmanovsky and Prof. Necmiye Ozay for reviewing the thesis, and providing very constructive feedback that helps place the work within a wider context.

The experience of these past years at IMT has significantly transformed my motivations, attitudes, and character as a whole. For this, I am extremely thankful. The transformation is driven by the people I have met, that I am fortunate to have as friends. Foremost, I would like to thank Laura for being a very supporting friend at every moment, and helping me realize how important bring thorough is (and also how to be a mathematician's friend - like calculus but less useful). A big thank you to Niraj, with whom I have had many a conversations regarding everything under the sun and moon. His statement *"It is always better to know about things going bad early rather than late"* will stay with me. I have been fortunate to enter Study room 1 when it was the *DYSCO corridor : Curry house edition*. The time spent talking to and cooking with Nilay, Vihang, Manas, Chinmay, Pavan, Francis, etc., stays vividly, pleasantly, and savorily in my memory. Their advise regarding life through an IMT PhD has been invaluable. I am very grateful to my most patient roommate Adrian and housemate Surya. You have been very kind. The several (few) vacations with Adrian, Ilaria, Laura, Lilly, Matteo, Nilay, and Niraj are full of memories I cherish, and have been very important to help me pushing on. The sh(eep) cheese, tomatoes, Michael Phelps with braccialetti, sights and sounds, etc. The accomplices in the several late night activities: ping-pong, walks, board games, bakery, Marameo, Giorgino, Bonabao, San Michele, Franklin, Magellano, etc, are all I thank, and sincerely apologize to for not mentioning individually here. Barbara, Antonella, and colleagues at the PhD office, by being kind bureaucratic ninjas, have been very helpful right from the beginning in navigating the many official status changes.

I would like to thank Prof. Balaji and Prof. Sivadas at Amrita, Dr. Ajinkya Bhave at Siemens (LMS), Prof. Goele Pipeleers and Dr. Ruben Van Parys at KU Leuven, for having inspired me to pursue a PhD, and direct myself towards control systems.

Finally, I would like to thank my family for the unending support, for which I will remain eternally grateful. Their enormous investment into the education of my sister and I, and their utmost belief in us being responsible and wise for many years now, are the greatest gifts one can receive.

Vita

June 18, 1992	Born, Hyderabad, India
2013	Bachelor of Technology in Mechanical Engineering Final mark: 8.78/10 distinction Amrita School of Engineering, India
2017	Master of Science in Mechanical Engineering Final mark: 14.82/20 cum laude KU Leuven, Belgium

Publications

1. S. K. Mulagaleti, A. Bemporad and M. Zanon, ”Computation of Least-Conservative State-Constraint Sets for Decentralized MPC With Dynamic and Constraint Coupling,” in IEEE Control Systems Letters, 2021.
2. S. K. Mulagaleti, A. Bemporad and M. Zanon, ”Input Constraint Sets for Robust Regulation of Linear Systems,” in IEEE Transactions on Automatic Control, 2022.
3. S. K. Mulagaleti, A. Bemporad and M. Zanon, ”Data-Driven Synthesis of Robust Invariant Sets and Controllers,” in IEEE Control Systems Letters, 2022.
4. S. K. Mulagaleti, A. Bemporad and M. Zanon, ”Computation of Input Disturbance Sets for Constrained Output Reachability,” in IEEE Transactions on Automatic Control, 2023.
5. S. K. Mulagaleti, M. Zanon and A. Bemporad, ”Dynamic output disturbance models for robust model predictive control,” in Proceedings of the 21st IFAC World Congress, 2020.
6. S. K. Mulagaleti, R. V. Parys and G. Pipeleers, ”Distributed model predictive control of multiple vehicles transporting a flexible payload,” 2018 European Control Conference (ECC), 2018.

Submitted

1. S. K. Mulagaleti, M. Zanon and A. Bemporad, ”A Smoothening-based Algorithm for the Computation of Input Disturbance Sets Guaranteeing Constrained Output Reachability”.
2. S. K. Mulagaleti, M. Zanon and A. Bemporad, ”Computation of Safe Disturbance sets using Implicit RPI sets” .

Presentations

1. S. K. Mulagaleti, ”Data-Driven Synthesis of Robust Invariant Sets and Controllers,” 2022 American Control Conference.
2. S. K. Mulagaleti, ”Computation of Least-Conservative State-Constraint Sets for Decentralized MPC With Dynamic and Constraint Coupling,” 2020 IEEE Conference on Decision and Control.
3. S. K. Mulagaleti, ”Dynamic output disturbance models for robust model predictive control,” 2020 IFAC World Congress.
4. S. K. Mulagaleti, ”Distributed model predictive control of multiple vehicles transporting a flexible payload,” 2018 European Control Conference.

Abstract

This dissertation presents new methods to synthesize disturbance sets and input constraints set for constrained linear time-invariant systems. Broadly, we formulate and solve optimization problems that (a) compute disturbance sets such that the reachable set of outputs approximates an assigned set, and (b) compute input constraint sets guaranteeing the stabilizability of a given set of initial conditions. The proposed methods find application in the synthesis and analysis of several control schemes such as decentralized control, reduced-order control, etc., as well as in practical system design problems such as actuator selection, etc.

The key tools supporting the develpment of the aforementioned methods are Robust Positive Invariant (RPI) sets. In particular, the problems that we formulate are such that they co-synthesize disturbance/input constraint sets along with the associated RPI sets. This requires embedding existing techniques to compute RPI sets within an optimization problem framework, that we facilitate by developing new results related to properties of RPI sets, polytope representations, inclusion encoding techniques, etc.

In order to solve the resulting optimization problems, we develop specialized structure-exploiting solvers that we numerically demonstrate to outperform conventional solution methods. We also demonstrate several applications of the methods we propose for control design. Finally, we extend the methods to tackle data-driven control synthesis problems in an identification-for-control framework.

Notation

- $\mathbb{R}^n$ denotes the set of real vectors of dimension n, and $\mathbb{R}^{m \times n}$ denotes the set of matrices of dimension $m \times n$.
- L_i denotes row i, and L_{ij} denotes the element in row i and column j of matrix L
- $\mathrm{rank}(L)$, $\mathrm{Im}(L)$, and $\mathrm{null}(L)$ denote the rank, image-space and null-space of matrix L respectively
- $\rho(L)$ denotes the spectral radius of a square matrix L
- $L \leq M$ denotes elementwise inequality of two matrices L, M of the same dimensions
- $\mathbf{1}$, $\mathbf{0}$, and $\mathbf{I}$ denote all-ones, all-zeros and identity matrix respectively, with dimensions specified if the context is ambiguous
- $\mathbb{I}_m^n$ denotes the set of natural numbers between two integers m and n, i.e, $\mathbb{I}_m^n := \{m, \ldots, n\}$
- $\mathrm{diag}(x) \in \mathbb{R}^{n \times n}$ is a matrix constructed with diagonal elements x_i of the vector $x \in \mathbb{R}^n$

Chapter 1

Overview

1.1 Introduction

In order to design effective control strategies for real-world plants, it is important to be able to predict the behaviour of the plant under the influence of a given control strategy. This prediction can be performed using dynamical models, which describe the input-output relationship of the plant. Such dynamical models can be obtained from an understanding of the physics of the underlying plant, through system/parameter identification strategies, or a combination of both. Since the plant can be a very complex system, this approach however can encounter several practical limitations. Identifying a dynamical model of the plant typically involves making several choices regarding the parameterization, and a complex plant can require a complex model parameterization for accurate predictions. Such models might not be amenable to use in a control design framework. On the other hand, a dynamical model that is simple enough for controller design might pose the risk that during online operations, the plant can violate its safety constraints leading to potentially catastrophic results. A reasonable trade-off between model complexity and prediction accuracy is presented by uncertainty descriptions. These descriptions capture the plant-model mismatch, and are appended to the dynamical models such that they can predict a range of future plant behaviours. Using simple dynamical models in conjunction with corresponding uncertainty descriptions, robust control strategies ensure safe operation for all predicted future plant behaviours in the presence of uncertainty. Since the actual plant behaviour is expected to be one of the predicted behaviours, guarantees regarding safe closed-loop operation with the controller can be provided.

1.2 Robust Invariant sets

An important tool used in the analysis of uncertain systems and synthesis of robust control strategies is the Robust Invariant set, that we introduce in this section. We consider a discrete-time model of a dynamical system written as

$$x(t+1) = \mathbf{f}_{\mathrm{OL}}(x(t), u(t), w(t)), \tag{1.1}$$

with current state vector $x(t)$, current input $u(t)$, and current disturbance (or uncertainty) vector $w(t)$. Such a model is used to predict the next state $x(t+1)$ of the plant. We consider further that the system is subject to pointwise-in-time constraints of the form

$$y(t) = \mathbf{g}(x(t), u(t)) \in \boldsymbol{\mathcal{Y}}. \tag{1.2}$$

We now define a Robust Control Invariant (RCI) set for the system.

Definition 1.1. *Given a set of disturbances $\boldsymbol{\mathcal{W}}$, a set $\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{W}})$ in the state-space is an RCI set for System* (1.1)-(1.2) *if and only if for each $x \in \boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{W}})$, there exists some control input u such that $\mathbf{g}(x, u) \in \boldsymbol{\mathcal{Y}}$, and $\mathbf{g}(\mathbf{f}_{\mathrm{OL}}(x, u, w), u) \in \boldsymbol{\mathcal{Y}}$ for every $w \in \boldsymbol{\mathcal{W}}$.*

Hence, the system can be enforced to evolve ad infinitum inside an RCI set by appropriately selecting a control input $u(t)$ at the current state $x(t)$. In order to select the control input $u(t)$, typically a control strategy is designed (assuming that $w(t)$ is unmeasured) as

$$u(t) = \mathcal{K}(x(t)). \tag{1.3}$$

The resulting closed-loop system given by

$$x(t+1) = \mathbf{f}_{\mathrm{CL}}(x(t), w(t)) \tag{1.4}$$

is an autonomous system, where $\mathbf{f}_{\mathrm{CL}}(x, w) = \mathbf{f}_{\mathrm{OL}}(x, \mathcal{K}(x), w)$. For the autonomous system, we now define a Robust Positive Invariant (RPI) set.

Definition 1.2. *Given a set of disturbances $\boldsymbol{\mathcal{W}}$, a set $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ in the state-space is an RPI set for the autonomous system in* (1.4) *with feedback law in* (1.3) *and subject to constraints* (1.2) *if and only if for each $x \in \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$, $\mathbf{g}(x, \mathcal{K}(x)) \in \boldsymbol{\mathcal{Y}}$ holds, and $\mathbf{g}(\mathbf{f}_{\mathrm{CL}}(x, w), \mathcal{K}(x)) \in \boldsymbol{\mathcal{Y}}$ for all disturbances $w \in \boldsymbol{\mathcal{W}}$.*

According to the definition of the RPI set, if $x(t_0) \in \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ at some time instant t_0, then the autonomous system always satisfies its constraints, i.e., $\mathbf{g}(x(t), u(t)) \in \boldsymbol{\mathcal{Y}}$ holds for all $t \geq t_0$. This is a key property that is used in the design and analysis of robust control and estimation schemes, and also in the analysis and design of autonomous systems.

1.2.1 Applications of Robust Invariant sets

Since RCI sets define regions of the state-space in which the system can be enforced to evolve ad infinitum, they prove to be very useful in safety-critical applications. For example, RCI sets formulated as superlevel sets of *control barrier functions* are discussed in, e.g., [2, 3], and are used to project the system input onto a set of safe inputs that guarantees that the system state remains inside the RCI set. They are also used extensively as terminal sets in Model Predictive Control (MPC) schemes, e.g., [83, 70], etc, in order to guarantee recursive feasibility. Moreover, in the case of uncertain linear systems, contractive RCI sets can be used to define Control Lyapunov functions through the Minkowski functional [16]. They also find applications in designing local control laws in a *non-cooperative game* setup [113], generating risky scenarios for safety-ensuring controller synthesis [27], etc.

On the other hand, since RPI sets define regions of the state-space in which an autonomous system evolves ad infinitum, they are naturally linked to stability analysis. It is well known that level sets of Lyapunov functions are RPI sets, which provide sufficient conditions to establish stability of equilibrium points [60, 59, 16]. Hence, they are used to synthesize Lyapunov functions for stability analysis, e.g. [19, 92, 121]. In [14, 15], feedback laws are derived to enforce invariance of a given arbitrary set, and stability of the derived law is established by demonstrating that the Minkowski functional over the given set is a Lyapunov function. For systems with persistently exciting additive disturbances, Input-to-State (ISS) Lyapunov functions [57] that demonstrate stability to ultimate-bounded regions rather than isolated equilibrium points are synthesized [72]. In applications such as Model Predictive Control (MPC), RPI sets are used as terminal sets to provide recursive feasibility guarantees [58, 119, 67, 18]. Moreover, in robust MPC approaches such as tube-based Robust MPC [87, 118] they are used to bound the deviation of a perturbed trajectory from a nominal one. A similar usage can also be found in several reference governor schemes [40, 41]. Several state estimation schemes are designed around minimizing the volume of the region in which the estimation error is bounded, thus naturally using RPI sets in their synthesis [90, 153].

1.2.2 Computation of Robust Invariant sets

Since Robust Invariant sets are invaluable tools for system and control analysis and design, methods to synthesize such sets is a very active area of research. Regarding RCI sets, one of the most popular and general proposal follows from the set iterations presented in [11]. These iterations are developed for general dynamical systems, and have been specialized over time to several classes of systems. An extensively studied application involves the case of LTI systems with polytopic constraints [148], where the conditions for finite time determination are discussed. Broad variants of these set iterations involve the *outside-in* and *inside-out* procedures [58]. Inside-out procedures, e.g. [32] initialize the iterations at a known RCI set, and expand out with each iterate being an RCI set. On the other hand, outside-in procedures only synthesize an RCI set upon convergence. However, if they converge, outside-in procedures converge to the Maximal RCI (MRCI) set, which is the union of all RCI sets. Recently, conditions for the convergence of the inside-out procedure to the MRCI set were studied in [78]. In order to avoid expensive set-projection operations, several methodologies are studied to compute tight approximations of the MRCI set, e.g., [37, 5, 159]. In [125], methods to compute arbitrarily tight approximations of the MRCI set for an LTI system with constraint set given as a union of polytopes is studied. For details regarding approaches adopted for computing RCI sets for nonlinear systems, we refer to [129] and the references therein.

In this thesis, we focus for a majority part on RPI sets, with specific attention paid to LTI systems with additive disturbances and polytopic constraints. Hence, we now proceed with a brief summary of relevant literature regarding computational methods for RPI sets for LTI systems. Of particular interest are the Maximal and the minimal RPI sets (MRPI and mRPI respectively) [16]. The mRPI set is the smallest RPI contained in all RPI sets $\mathcal{X}(\mathcal{W})$, while the MRPI set is the largest RPI that is the union of all RPI sets. The problem of computing RPI sets for linear systems has been extensively studied. In [44, 63, 106], the MRPI set was exactly computed for such systems with additive disturbances and polytopic model uncertainty and polytopic constraints, and the methods have been implemented in the software package MAXIS-G [91]. Similar iterations were used to compute the MRPI set in the presence of quasi-smooth nonlinear constraints in [153]. Computation of this set in the presence of state-dependent and scaled disturbances was studied in [130]. While these approaches can compute the MRPI set exactly, they might not be suitable for high-order and/or very slow systems due to high computational requirements and complexity of the resulting MRPI set. Hence, several methods to approximate the MRPI sets with RPI sets of a fixed complexity/representation have been developed. For example, in [19, Chapter 5], ellipsoidal invariant sets are computed for linear systems as sublevel sets of quadratic Lyapunov function. In [124, 123], an RPI set parameterized as a polytope is computed for linear systems with polytopic uncertainty using LMI techniques. It is well understood from [16] that, while ellipsoidal sets can be representationally simple, they can be excessively conservative. On the other hand, while polytopic sets can be less conservative, they can be excessively complex. In [103], RPI sets with a semi-ellipsoidal representation were synthesized in order to strike a trade-off between complexity and conservativeness. In [1], a method was presented to construct polytopic RPI sets starting from a contractive ellipsoidal set. More recent techniques focus on co-synthesizing a feedback controller (1.3) along with a corresponding RPI set for the resulting autonomous system (1.4) by fixing the RPI set parameterization. For example, in [139], a linear feedback gain and corresponding RPI sets parameterized as ellipsoids and hyperboxes are computed. RPI sets parameterized as low-complexity polytopes, i.e., polytopes with twice the number of hyperplanes as the state dimension, were co-synthesized along with corresponding linear static feedback laws in [17] by iteratively solving a nonlinear program. Similar low-complexity polytopic RPI sets with linear feedback laws were synthesized for linear systems with norm-bounded uncertainty and additive disturbances in [138], that was later extended to compute full-complexity polytopes in [74, 75]. In [49, 48] low and full-complexity polytopic RPI sets with associated linear static feedback laws for linear systems with additive uncertainty as well as a rational parameter dependence were synthesized. While the MRPI set can be computed exactly in the case of linear systems subject to additive disturbances under restrictions of strict stability and the origin being included in the nonempty interior of the constraint sets, exact computation of the mRPI set, even if well defined, is impossible except under very restrictive assumptions [63, 16, 135]. Hence, typically outer (RPI) approximations of the mRPI set are sought. A popular technique in [112, 111] involves appropriately scaling the 0-reachable set to compute arbitrarily tight RPI approximations of the mRPI set, that was extended in [66] to also accomodate polytopic model uncertainty. Alternatively, tight RPI approximations can also be computed by shrinking an ellipsoidal RPI set obtained as the level set of an ISS-Lyapunov function, as shown in [85, 141]. Since these approaches can be excessively computationally intensive as they involve Minkowski sums, and the resulting approximations non-viable for online control synthesis, there exist many methods to compute tight RPI approximations of the mRPI set with a predefined complexity. In [115, 118, 144], the smallest polytope with predefined normal vectors to the hyperplanes was used to approximate the mRPI set. Low-complexity polytopic RPI approximations of the mRPI set, along with the associated invariance inducing linear feedback controller were synthesized in [140]. Moreover, the methods in [74, 75] can also be used to compute full-complexity polytopic RPI sets and associated feedback laws.

The problem of computing RPI sets for nonlinear systems is typically more involved. Methods to compute invariant sets based on backward reachable sets from piecewise affine (PWA) linear systems were presented in [114, 46]. Such approaches however can be very computationally intensive. Alternatively, exploiting the link between Lyapunov functions and RPI sets, methods to synthesize piecewise Lyapunov functions with piecewise level sets that are RPI sets were developed [22, 69, 25, 152]. A useful tool in this regard is Sum-of-Squares (SOS) programming [108], that allows for the reformulation of polynomial constraints into Linear Matrix Inequalities (LMIs) using which convex computation of polynomial Lyapunov functions can be performed. Such tools have also been used in the synthesis of invariant sets using occupancy measures [52], and based on solving a Bellman type equation [157, 158]. Some recent results on the computation of parameter-dependent RPI sets [50, 29] present attractive approaches for the synthesis of robust control algorithms with reduced conservativeness.

1.3 Thesis outline and contributions

As demonstrated in Section 1.2, most of the contributions regrading the computation of RPI sets are based on the assumption that the set of inputs/disturbances are known a priori. Moreover, they also assume that a model of the plant is known beforehand. While this assumption is reasonable in many applications, the control practitioner must suitably define the set of inputs/disturbances, and perform system identification in order to identify a model of the plant in question. In this thesis, we focus the developments by exclusively considering uncertain linear time-invariant (LTI) plant models of the form

$$x(t+1) = Ax(t) + Bu(t) + B_w w(t), \tag{1.5a}$$
$$y(t) = Cx(t) + Du(t) + D_w w(t), \tag{1.5b}$$

with state $x \in \mathbb{R}^{n_x}$, input $u \in \mathbb{R}^{n_u}$, output $y \in \mathbb{R}^{n_y}$ and unmeasured disturbance $w \in \mathbb{R}^{n_w}$. Considering that the system in (1.5) is subject to output constraints $y \in \mathcal{Y}$, we tackle the following questions:

1. Assuming that system (1.5) is autonomous because either (a) matrices B and D are all-zero matrices, or (b) a stabilizing linear feedback gain $u = Kx$ is designed a priori, how can we compute an *appropriate* disturbance set $\mathcal{W}$ such that under the action of arbitrary disturbances $w \in \mathcal{W}$, the set of reachable outputs approximates the assigned output constraint set $\mathcal{Y}$?

2. Given a set of initial conditions $\mathcal{X}_0$ of system (1.5), i.e., $x(0) \in \mathcal{X}_0$ and a set of persistently exciting disturbances $\mathcal{W}$, i.e., $w(t) \in \mathcal{W}$ for all $t \geq 0$, how can we compute an input constraint set $\mathcal{U}$ such that with inputs $u \in \mathcal{U}$, system (1.5) can be regulated while robustly respecting the constraints $y \in \mathcal{Y}$?

3. Given a dataset of state and input measurements from a plant, how can we identify an LTI model (1.5) along with synthesizing a robustly regulating controller with reduced conservativeness?

As we will demonstrate in this thesis, the aforementioned questions can be tackled by using RPI sets parameterized by the input/disturbance constraint sets. Hence, for each of the questions, we formulate and solve optimization problems that co-synthesize the sets of interest. In this thesis, we address the formulation of and the solution methods for these optimization problems, and is organized as follows:

- In Chapter 3, we address Question 1, i.e., the problem computing appropriate input disturbance sets for constrained output reachability. To this end, we first parameterize the disturbance set as a polytope with a priori known normal vectors to the hyperplanes. Then, we formulate an optimization problem to solve for the right-hand-side parameters of the polytope. In this problem, we enforce constraints that the set of reachable outputs formulated using the minimal RPI set approximates the assigned output constraint set $\mathcal{Y}$. Since obtaining an explicit representation of the minimal RPI set is generally impossible, we approximate it in the optimization problem formulation with an RPI set parameterized as a polytope with a priori fixed normal vectors. We show that the smallest RPI set (in terms of inclusions) with the chosen parameterization is uniquely defined, that allows us to bring the optimization formulation into an implementable form. In this form, the constraints are formulated using support functions over polytopes that can be difficult to resolve since they are nonsmooth. Hence, in the second part of the chapter, we develop an optimization algorithm to solve the formulated problem based on smoothening-techniques. We adopt notions of parametric optimization theory to treat the support functions as implicit functions, based on which we develop a sensitivity-based Primal-Dual Interior Point solver. Finally we demonstrate the efficacy of the proposed formulation and optimization algorithm to tackle Question 1 using numerical examples. The content of this chapter is based primarily on [97, 96].

- In Chapter 4, we apply the techniques developed in Chapter 3 to synthesize a decentralized MPC control scheme. In particular, we consider a system composed of dynamically coupled subsystems and subject to coupled constraints on the output. We decouple these dynamics and constraints by computing state-constraint sets on the individual subsystems, that satisfy the requirement that if the subsystems satisfy their local constraints, then the overall system constraints are satisfied. This chapter is based on [94].

- In Chapter 5, we reconsider Question 1, and present an alternative methodology to compute disurbance sets that guarantee constraint satisfaction. This method is based on using implicitly-parameterized

RPI sets to approximate the mRPI set within the formulation of the optimization problem, instead of using a RPI set parameterized as a polytope with prespecified normal vectors. The implicit RPI sets permit to specify a priori the RPI approximation error, thus overcoming the first drawback of the approach proposed in Chapter 3. Then, we present a novel disturbance set parameterization, that allows for encoding the set of feasible disturbance sets as a polytope, thus overcoming the second drawback of the approach in Chapter 3. We demonstrate the efficacy of the method to tackle Question 1, and compare its performance against the approach of Chapter 3. Finally, we show an application of the method to synthesize a reduced-order MPC scheme. This chapter is based on [98].

- In Chapter 6, we address Question 2, i.e., the problem of computing input constraint sets $\mathcal{U}$ that can be used to guarantee robust regulation of given set of initial conditions. To this end, we assume that the plant is equipped with a tube-based Robust MPC controller, the properties of which we exploit to formulate and solve an optimization problem to compute the *smallest* input constraint set $\mathcal{U}$ that guarantees recursive feasibility. We use a positively invariant terminal set parameterized by the input constraint set within the optimization problem, and solve the optimization problem akin to [44]. We demonstrate the efficacy of the proposed method using an actuator selection problem, in which the input constrant set $\mathcal{U}$ is parameterized using discrete variables, and the size of the set is related to an economic cost. This chapter is based on [100].

- In Chapter 7, we address Question 3, i.e., the problem of identifying an LTI model along with synthesizing a tube-based Robust MPC controller using a dataset of state and input measurements from a plant. To this end, we characterize a set of LTI models that can robustly represent the plant behaviour with formal guarantees. Then, we use this set to formulate and solve an optimization problem that, along with selecting an LTI model, also computes corresponding RPI sets and invariance-inducing feedback controllers. This problem is solved using a sequential convexification approach formulated through Linear Matrix Inequality (LMI) approximations. This chapter is based on [99].

Chapter 2

Preliminaries

In this chapter, we present some preliminary results that will be used in the developments in the sequel. The first part of this chapter is devoted to recalling some basic mathematical tools, that will be used in the second part to recall some results from literature regarding Robust Positive Invariant (RPI) sets.

2.1 Set operations and representations

Given two general sets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^n$, the set operations that we will most frequently encounter in this thesis are

- *Linear Transformation*: $L\mathcal{S}_1 := \{y \ : \ y = Lx, \ x \in \mathcal{S}_1\}$, $L \in \mathbb{R}^{m \times n}$
- *Minkowski sum*: $\mathcal{S}_1 \oplus \mathcal{S}_2 := \{z \ : \ z = x + y, \ x \in \mathcal{S}_1, \ y \in \mathcal{S}_2\}$
- *Minkowski difference*: $\mathcal{S}_1 \ominus \mathcal{S}_2 := \{z \ : \ z + y \in \mathcal{S}_1, \ \forall\, y \in \mathcal{S}_2\}$

We refer the reader to [63, 131] for details regarding the properties of these operations for general sets $\mathcal{S}_1$ and $\mathcal{S}_2$. A useful tool while analyzing properties of sets is the support function. The support function of a set $\mathcal{S} \subset \mathbb{R}^n$ evaluated at some $y \in \mathbb{R}^n$ is defined as

$$h_{\mathcal{S}}(y) := \sup_{x \in \mathcal{S}} y^\top x. \tag{2.1}$$

If $\mathcal{S}$ is a bounded set, then $h_{\mathcal{S}}(y)$ is bounded above for all $y \in \mathbb{R}^n$ such that $\sup$ operator can be replaced by the $\max$ operator. If the set $\mathcal{S}$ is compact and convex, then it is the intersection of its supporting halfspaces, i.e.,

$$\mathcal{S} = \bigcap_{y \in \mathbb{R}^n} \{x : y^\top x \leq h_{\mathcal{S}}(y)\}. \tag{2.2}$$

Moreover, the inclusion $\mathcal{S}_1 \subseteq \mathcal{S}_2$ holds if and only if

$$h_{\mathcal{S}_1}(y) \leq h_{\mathcal{S}_2}(y), \qquad \forall\, y \in \mathbb{R}^n. \tag{2.3}$$

For some compact sets $\mathcal{S}_1$ and $\mathcal{S}_2$ in $\mathbb{R}^n$ and some p-norm ball defining

$$\bar{d}_{\mathrm{H}}^p(\mathcal{S}_1, \mathcal{S}_2) := \min_{\epsilon}\{\epsilon \ : \ \mathcal{S}_2 \subseteq \mathcal{S}_1 \oplus \epsilon \mathcal{B}_p^n\}, \tag{2.4}$$

the p-norm induced Hausdorff distance between $\mathcal{S}_1$ and $\mathcal{S}_2$ is given by

$$d_{\mathrm{H}}^p(\mathcal{S}_1, \mathcal{S}_2) := \max\{\bar{d}_{\mathrm{H}}^p(\mathcal{S}_1, \mathcal{S}_2), \ \bar{d}_{\mathrm{H}}^p(\mathcal{S}_2, \mathcal{S}_1)\}. \tag{2.5}$$

If the inclusion $\mathcal{S}_1 \subseteq \mathcal{S}_2$ holds, then $d_{\mathrm{H}}^p(\mathcal{S}_1, \mathcal{S}_2) = \bar{d}_{\mathrm{H}}^p(\mathcal{S}_1, \mathcal{S}_2)$. In this thesis, we primarily use the ∞-norm induced Hausdorff distance. Hence, we denote $d_{\mathrm{H}}^\infty(\mathcal{S}_1, \mathcal{S}_2)$ by $d_{\mathrm{H}}(\mathcal{S}_1, \mathcal{S}_2)$ for simplicity of notation.

Polytopes

We focus our developments primarily using sets parameterized as polytopes. A polytope is a bounded polyhedron, and a polyhedron $\mathcal{S} \subset \mathbb{R}^n$ is an intersection of a finite-number of halfspaces $\mathcal{H}_i$, i.e.,

$$\mathcal{S} := \bigcap_{i=1}^{s} \mathcal{H}_i, \text{ where } \mathcal{H}_i := \{x \ : \ M_i x \leq q_i\} \text{ for some } M_i^\top \in \mathbb{R}^n, \ q_i \in \mathbb{R}.$$

The polyhedron $\mathcal{S}$ can be equivalently represented as

$$\mathcal{S} = \{x : Mx \leq q\},$$

where $\leq$ denotes the elementwise inequality. Every polytope is also uniquely characterized by its finite-number of vertices, i.e.,

$$\{x_{[i]}, i \in \mathbb{I}_1^{v_{\mathcal{S}}}\} = \mathrm{vert}(\mathcal{S}). \tag{2.6}$$

A polytope is said to be in a *minimal-representation* [6] if the removal of a halfspace $\mathcal{H}_i$ would not change it. Moreover if the vector $q \geq \mathbf{0}$, then the origin is included in the polytope, i.e., $\mathbf{0} \in \mathcal{S}$. If $q > 0$, then the origin is included in the nonempty interior $\text{int}(\mathcal{S}) \subset \mathcal{S}$, i.e., $\mathbf{0} \in \text{int}(\mathcal{S})$. A polytope is said to be full-dimensional if it includes a nonempty translated p-norm ball defined as $\mathcal{B}_p^n := \{x : \|x\|_p \leq 1\}$ in $\mathbb{R}^n$, i.e,

$$\exists\, x_0 \in \mathbb{R}^n, \epsilon > 0 : \{x_0\} \oplus \epsilon\mathcal{B}_p^n \subseteq \mathcal{S} \tag{2.7}$$

For polytopes $\mathcal{S}_1 = \{x : Mx \leq q\}$ and $\mathcal{S}_2 = \{x : Nx \leq r\}$, the Minkowski sum $\mathcal{S}_1 \oplus \mathcal{S}_2$ can be computed as

$$\begin{aligned}
\mathcal{S}_1 \oplus \mathcal{S}_2 &= \{z : z = x + y,\ Mx \leq q,\ Ny \leq r\} \\
&= \left\{ z : \exists\, x \ : \ \begin{bmatrix} M & \mathbf{0} \\ -N & N \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \leq \begin{bmatrix} q \\ r \end{bmatrix} \right\} \\
&= \Pi_z \left(\left\{ \begin{bmatrix} x \\ z \end{bmatrix} : \begin{bmatrix} M & \mathbf{0} \\ -N & N \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \leq \begin{bmatrix} q \\ r \end{bmatrix} \right\} \right),
\end{aligned}$$

where $\Pi_z(.)$ is the projection operator onto the components z.

Over a polytope $\mathcal{S} = \{x : Mx \leq q\}$, the support function defined in (2.1) can be computed by solving the Linear Program (LP)

$$h_{\mathcal{S}}(y) = \max_x \{y^\top x \quad \text{s.t.} \quad Mx \leq q\}. \tag{2.8}$$

We recall the following properties of support functions from [63] for any polytopes $\mathcal{S}, \mathcal{P} \subset \mathbb{R}^n$, vectors $y \in \mathbb{R}^n$ and $z \in \mathbb{R}^m$, scalar $\alpha \geq 0$ and matrix $L \in \mathbb{R}^{m \times n}$:

1. $h_{\alpha\mathcal{S}}(y) = h_{\mathcal{S}}(\alpha y) = \alpha h_{\mathcal{S}}(y)$
2. $h_{\mathcal{S}\oplus\mathcal{P}}(y) = h_{\mathcal{S}}(y) + h_{\mathcal{P}}(y)$
3. $h_{L\mathcal{S}}(z) = h_{\mathcal{S}}(L^\top z)$.

Using support functions, polytope $\mathcal{S}$ can equivalently be expressed as

$$\mathcal{S} = \{x : Mx \leq q\} = \bigcap_{i=1}^{s} \{x : M_i x \leq h_{\mathcal{S}}(M_i^\top)\} \tag{2.9}$$

following from (2.2), i.e., as an intersection of a finite number of halfspaces. Thus, in general the inequality $h_{\mathcal{S}}(M_i^\top) \le q_i$ holds for all components $i \in \mathbb{I}_1^s$, with strict inequality $h_{\mathcal{S}}(M_i^\top) < q_i$ implying that the halfspace $\mathcal{H}_i = \{x : M_i x \le q_i\}$ is redundant in the definition of $\mathcal{S}$.

The Minkowski difference between a polytope $\mathcal{S}$ and some set $\mathcal{P}$ can be computed using support functions as

$$\mathcal{S} \ominus \mathcal{P} := \bigcap_{i=1}^{s} \{x : M_i x \le q_i - h_{\mathcal{P}}(M_i^\top)\}. \tag{2.10}$$

Then if $\mathcal{P}$ is a polytope, then the Minkowski difference can be computed by solving r number of LPs from (2.8).

Given a polytope $\mathcal{S}_1 = \{x : Mx \le q\}$ and $\mathcal{S}_2 = \{x : Nx \le r\}$ with $q \in \mathbb{R}^{s_1}$ and $r \in \mathbb{R}^{s_2}$, the inclusion $\mathcal{S}_1 \subseteq \mathcal{S}_2$ holds if and only if

$$h_{\mathcal{S}_1}(N_i^\top) \le h_{\mathcal{S}_2}(N_i^\top) \le r_i, \qquad \forall\, i \in \mathbb{I}_1^{s_2}. \tag{2.11}$$

Then if $\mathcal{S}_2$ does not contain any redundant hyperplanes, the inclusion can be verified by solving s_2 number of LPs as

$$h_{\mathcal{S}_1}(N_i^\top) = \left\{ \begin{array}{ll} \max\limits_{x} & N_i x \\ \text{s.t.} & Mx \le q \end{array} \right\} \le r_i, \qquad \forall\, i \in \mathbb{I}_1^{s_2}. \tag{2.12}$$

Because of strong duality of LPs [132], the equality

$$\left\{ \begin{array}{ll} \max\limits_{x} & N_i x \\ \text{s.t.} & Mx \le q \end{array} \right\} = \left\{ \begin{array}{ll} \min\limits_{\lambda_i \ge \mathbf{0}} & \lambda_i q \\ \text{s.t.} & \lambda_i M = N_i, \end{array} \right\} \tag{2.13}$$

holds, where $\lambda_i \in \mathbb{R}^{1 \times s_1}$, and the LP on the right-hand-side is the dual LP corresponding to the LP defining the support function $h_{\mathcal{S}_1}(N_i^\top)$ on the left-hand-side. Defining the set

$$\mathbf{\Lambda} := \{\Lambda \in \mathbb{R}^{s_2 \times s_1} \; : \; \Lambda \ge \mathbf{0},\ \Lambda M = N\},$$

it then follows from (2.12) and (2.13) that the inclusion $\mathcal{S}_1 \subseteq \mathcal{S}_2$ holds if and only if

$$\exists\, \Lambda \in \mathbf{\Lambda} \; : \Lambda q \le r. \tag{2.14}$$

Finally, if the inclusion $\mathcal{S}_1 \subseteq \mathcal{S}_2$ holds, and the vertices $\{y_{[i]}, i \in \mathbb{I}_1^{v_{\mathcal{S}_2}}\}$ are known a priori such that

$$\mathcal{S}_2 = \mathrm{ConvHull}(y_{[i]}, i \in \mathbb{I}_1^{v_{\mathcal{S}_2}}) \tag{2.15}$$

where $\mathrm{ConvHull}(.)$ is the convex-hull operator, then the Hausdorff distance $d_{\mathrm{H}}(\mathcal{S}_1, \mathcal{S}_2)$ is given by the value of the LP

$$\begin{aligned} \min_{\epsilon, \{x_{[i]}, z_{[i]}, i \in \mathbb{I}_1^{v_{\mathcal{S}_2}}\}} \quad & \epsilon && \qquad (2.16) \\ \text{s.t.} \quad & x_{[i]} + z_{[i]} = y_{[i]}, && i \in \mathbb{I}_1^{v_{\mathcal{S}_2}}, \\ & x_{[i]} \in \mathcal{S}_1,\ z_{[i]} \in \epsilon \mathcal{B}_\infty^n, && i \in \mathbb{I}_1^{v_{\mathcal{S}_2}}. \end{aligned}$$

Having recalled these basic tools, we will now present some fundamental concepts regarding RPI sets for LTI systems of the form in Equation (1.5).

2.2 RPI sets for LTI systems

In this section, we recall some fundamental concepts regarding RPI sets for LTI systems of the form in (1.5). To this end, we assume that the system is equipped with a linear static feedback law of the form $u = Kx$, resulting in the closed-loop system

$$x(t+1) = (A + BK)x(t) + B_w w(t) \tag{2.17a}$$
$$y(t) = (C + DK)x(t) + D_w w(t). \tag{2.17b}$$

We denote the closed-loop matrices by

$$A_K := A + BK, \qquad C_K := C + DK$$

in the sequel. Assuming that the disturbance set $\boldsymbol{\mathcal{W}}$ is given, i.e., $w(t) \in \boldsymbol{\mathcal{W}}$ for all $t \geq 0$, a set $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \subset \mathbb{R}^{n_x}$ is RPI for this system if and only if it satisfies the inclusion

$$A_K \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \oplus B_w \boldsymbol{\mathcal{W}} \subseteq \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}). \tag{2.18}$$

If the sets $\boldsymbol{\mathcal{W}}$ and $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ are polytopes defined as

$$\boldsymbol{\mathcal{W}} := \{w \ : \ F_i w \leq \epsilon_i^w, \ \forall\, i \in \mathbb{I}_1^{m_W}\}, \tag{2.19a}$$
$$\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) := \{x \ : \ E_i x \leq \epsilon_i^x, \ \forall\, i \in \mathbb{I}_1^{m_X}\}, \tag{2.19b}$$

then exploting basic properties of support functions and (2.11), the RPI condition in (2.18) can equivalently be written as

$$\forall\, i \in \mathbb{I}_1^{m_X} \begin{cases} h_{A_K \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})}(E_i^\top) + h_{B_w \boldsymbol{\mathcal{W}}}(E_i^\top) \leq h_{\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})}(E_i^\top) \\ \qquad \Leftrightarrow h_{\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})}(A_K^\top E_i^\top) + h_{\boldsymbol{\mathcal{W}}}(B_w^\top E_i^\top) \leq h_{\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})}(E_i^\top). \end{cases}$$

Hence, given the sets in (2.19), the RPI inclusion in (2.18) can be verified by solving $3 \times m_X$ number of LPs. Moreover, if the set $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ does not contain any redundant hyperplanes, then $h_{\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})}(E_i^\top) = \epsilon_i^x$, such that the inclusion can be verified by solving $2 \times m_X$ number of LPs. Alternatively, by exploting the duality conditions in (2.14), the RPI inclusions can be verified if the set $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ does not contain any redundant hyperplanes by checking the existence of matrices nonnegative $\Lambda \geq \mathbf{0},\ M \geq \mathbf{0}$ satisfying

$$\Lambda E = E A_K,\ MF = EB_w,\ \Lambda \epsilon^x + M\epsilon^w \leq \epsilon^x. \tag{2.20}$$

Assuming that the output of the system in (2.17b) is subject to polytopic constraints $\boldsymbol{\mathcal{Y}}$ as

$$y \in \boldsymbol{\mathcal{Y}} := \{y \ : \ G_i y \leq g_i,\ \forall\, i \in \mathbb{I}_1^{m_\mathcal{Y}}\}, \tag{2.21}$$

the output inclusion

$$C_K \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \oplus D_w \boldsymbol{\mathcal{W}} \subseteq \boldsymbol{\mathcal{Y}} \tag{2.22}$$

can be verified using support functions as

$$\forall\, i \in \mathbb{I}_1^{m_Y} \begin{cases} h_{C_K \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})}(G_i^\top) + h_{D_w \boldsymbol{\mathcal{W}}}(G_i^\top) \leq g_i \\ \qquad \Leftrightarrow h_{\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})}(C_K^\top G_i^\top) + h_{\boldsymbol{\mathcal{W}}}(D_w^\top G_i^\top) \leq g_i. \end{cases} \tag{2.23}$$

Since $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ is an RPI set, if the initial state $x(0) \in \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ then it holds that $x(t) \in \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}),\ \forall\, t \geq 0$ for all future disturbance sequences $w(t) \in \boldsymbol{\mathcal{W}},\ \forall\, t \geq 0$. Then, if inclusion (2.22) is verified, it implies that the system always satisfies its constraints, i.e., $y(t) \in \boldsymbol{\mathcal{Y}},\ \forall\, t \geq 0$. This property permits $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ to be used as a terminal set in MPC techniques in order to provide recursive feasibility guarantees.

2.2.1 The minimal RPI set

What are the conditions that guarantee the existence of an RPI set $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ satisfying inclusion (2.22)*?*

Given some disturbance set $\boldsymbol{\mathcal{W}}$, the set of states that can be reached by the system from the origin in t-steps is given by

$$\boldsymbol{\mathcal{X}}_t(\boldsymbol{\mathcal{W}}) := \left\{ x(t) \ : \ x(t) = \sum_{k=0}^{t-1} A_K^{t-k-1} B_w w(k), \ \forall \ w(k) \in \boldsymbol{\mathcal{W}} \right\}, \quad (2.24)$$

the limit of which is referred to as the 0-reachable set $\boldsymbol{\mathcal{X}}_{\mathrm{m}}(\boldsymbol{\mathcal{W}})$, i.e.,

$$\boldsymbol{\mathcal{X}}_{\mathrm{m}}(\boldsymbol{\mathcal{W}}) := \lim_{t\to\infty} \boldsymbol{\mathcal{X}}_t(\boldsymbol{\mathcal{W}}). \quad (2.25)$$

If the following assumptions are satisfied:

1. The spectral radius of A_K is strictly less that 1, i.e., $\rho(A_K) < 1$;

2. The origin belongs to the compact and convex disturbance set $\boldsymbol{\mathcal{W}}$, i.e., $\mathbf{0} \in \boldsymbol{\mathcal{W}}$,

then the reachable sets satisfy the inclusions

$$\boldsymbol{\mathcal{X}}_t(\boldsymbol{\mathcal{W}}) \subseteq \boldsymbol{\mathcal{X}}_{t+1}(\boldsymbol{\mathcal{W}}) \subseteq \boldsymbol{\mathcal{X}}_{\mathrm{m}}(\boldsymbol{\mathcal{W}}), \ \ \forall t \geq 0,$$

and the 0-reachable set $\boldsymbol{\mathcal{X}}_{\mathrm{m}}(\boldsymbol{\mathcal{W}})$ is a uniquely-defined compact and convex set that contains the origin [63, Theorem 4.1]. Moreover, is satisfies

$$A_K \boldsymbol{\mathcal{X}}_{\mathrm{m}}(\boldsymbol{\mathcal{W}}) \oplus B_w \boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{X}}_{\mathrm{m}}(\boldsymbol{\mathcal{W}}),$$

from which it can be observed that it satisfies the RPI condition in (2.18). Hence, $\boldsymbol{\mathcal{X}}_{\mathrm{m}}(\boldsymbol{\mathcal{W}})$ is an RPI set. Infact, it was shown in [63, Corollary 4.2] that $\boldsymbol{\mathcal{X}}_{\mathrm{m}}(\boldsymbol{\mathcal{W}})$ is the *minimal* RPI (mRPI) set, i.e., it is the smallest (in an inclusion sense) RPI set of the system, such that

$$\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \text{ is RPI } \implies \boldsymbol{\mathcal{X}}_{\mathrm{m}}(\boldsymbol{\mathcal{W}}) \subseteq \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}). \quad (2.26)$$

In a set-theoretic notation, the mRPI set can be expressed using Minkowski sums as

$$\mathcal{X}_{\mathrm{m}}(\mathcal{W}) = \bigoplus_{k=0}^{\infty} A_K^k B_w \mathcal{W}. \tag{2.27}$$

Hence, from the inclusions in (2.26) and (2.22), we observe that there exists an RPI set $\mathcal{X}(\mathcal{W})$ satisfying the output-constraint set inclusion if and only if the mRPI set satisfies the inclusion as

$$C_K \mathcal{X}_{\mathrm{m}}(\mathcal{W}) \oplus D\mathcal{W} \subseteq \mathcal{Y}, \tag{2.28}$$

thus answering the aforementioned question. Unfortunately, from the infinite Minkowski sum in (2.27), we see that in general, the mRPI set $\mathcal{X}_{\mathrm{m}}(\mathcal{W})$ cannot be computed exactly. Hence, one typically attempts to compute RPI sets that tightly approximate the mRPI set. The characterization and computation of such RPI sets is an active research area, e.g., [112, 144], etc.

2.2.2 The maximal RPI set

What is the largest RPI set $\mathcal{X}(\mathcal{W})$ satisfying inclusion (2.22)*?*

We recall that inclusion (2.28) is necessary and sufficient for the existence of an RPI set satisfying inclusion (2.22). Since the origin belongs to the mRPI set $\mathcal{X}_{\mathrm{m}}(\mathcal{W})$, it is hence necessary that the origin belongs to the output-constraint set $\mathcal{Y}$ for inclusion (2.28) to be satisfied. Assuming that inclusion (2.28) is satisfied, the Maximal RPI (MRPI) set $\mathcal{X}_{\mathrm{M}}(\mathcal{W})$ satisfying inclusion (2.22) is defined as the union of all RPI sets satisfying inclusion (2.22), i.e.,

$$\mathcal{X}_{\mathrm{M}}(\mathcal{W}) := \bigcup \{\mathcal{X}(\mathcal{W}) : (2.18), (2.22)\}.$$

It was shown in [63] that the MRPI set $\mathcal{X}_{\mathrm{M}}(\mathcal{W})$ is the limit set of the iterations

$$\mathcal{O}_0(\mathcal{W}) := \{x \ : \ C_K x \in \mathcal{Y}_0(\mathcal{W})\} \tag{2.29}$$

$$\mathcal{O}_{t+1}(\mathcal{W}) := \mathcal{O}_t(\mathcal{W}) \cap \{x \ : \ C_K A_K^{t+1} x \in \mathcal{Y}_{t+1}(\mathcal{W})\}, \tag{2.30}$$

i.e.,

$$\boldsymbol{\mathcal{X}}_{\mathrm{M}}(\boldsymbol{\mathcal{W}}) = \boldsymbol{\mathcal{O}}_{\infty}(\boldsymbol{\mathcal{W}}) := \bigcap_{t \geq 0} \boldsymbol{\mathcal{O}}_t(\boldsymbol{\mathcal{W}}),$$

where the sets $\boldsymbol{\mathcal{Y}}_0(\boldsymbol{\mathcal{W}})$ and $\boldsymbol{\mathcal{Y}}_{t+1}(\boldsymbol{\mathcal{W}})$ are defined as

$$\begin{aligned} \boldsymbol{\mathcal{Y}}_0(\boldsymbol{\mathcal{W}}) &:= \boldsymbol{\mathcal{Y}} \ominus D_w \boldsymbol{\mathcal{W}}, \\ \boldsymbol{\mathcal{Y}}_{t+1}(\boldsymbol{\mathcal{W}}) &:= \boldsymbol{\mathcal{Y}}_t \ominus C_K A_K^t B_w \boldsymbol{\mathcal{W}}. \end{aligned}$$

Moreover, if the tighter inclusion

$$C_K \boldsymbol{\mathcal{X}}_{\mathrm{m}}(\boldsymbol{\mathcal{W}}) \oplus D\boldsymbol{\mathcal{W}} \subseteq \mathrm{int}(\boldsymbol{\mathcal{Y}}) \subset \boldsymbol{\mathcal{Y}} \tag{2.31}$$

holds, then there exists some $t^* \geq 1$ such that

$$\boldsymbol{\mathcal{O}}_{\infty}(\boldsymbol{\mathcal{W}}) = \boldsymbol{\mathcal{O}}_{t^*}(\boldsymbol{\mathcal{W}}) = \boldsymbol{\mathcal{O}}_{t^*+1}(\boldsymbol{\mathcal{W}}).$$

Such an index t^* is called the finite-determination index, and the set $\boldsymbol{\mathcal{O}}_{\infty}(\boldsymbol{\mathcal{W}})$ is said to be finitely-determined, i.e., it can be computed exactly in finite-time.

2.2.3 Tube-based Robust Model Predictive Control

In order to demonstrate some control design applications of the methods presented in this thesis, we use the tube-based Robust Model Predictive Control (RMPC) scheme presented in [87]. We recap here some basic design ideas of this scheme, and present its relevant properties. In the remaining chapters in this thesis, we recall the scheme, with notation specialized to the problem being tackled in the respective chapters.

We recall the LTI System (1.5) with dynamics

$$x(t+1) = Ax(t) + Bu(t) + B_w w(t), \tag{2.32a}$$
$$y(t) = Cx(t) + Du(t) + D_w w(t), \tag{2.32b}$$

that is subject to output constraints $y \in \boldsymbol{\mathcal{Y}}$ and persistent disturbances $w \in \boldsymbol{\mathcal{W}}$. The tube-based RMPC scheme in [87] can be used to robustly regulate the state of System (2.32), and is constructed as follows.

First, a feedback gain K is assumed to be known a priori such that $\rho(A_K) < 1$, where $A_K = A + BK$. Then, exploiting linearity the state and output are split into nominal and perturbed components as

$$x = \hat{x} + \Delta x, \qquad\qquad y = \hat{y} + \Delta y, \tag{2.33}$$

and the control input is parameterized as

$$u = \hat{u} + K\Delta x, \tag{2.34}$$

where $\hat{u}$ is the nominal control input. Then, denoting $C_K = C + DK$, the dynamics of the nominal and perturbed systems are written as

$$\text{Nominal system} \begin{cases} \hat{x}(t+1) = A\hat{x}(t) + B\hat{u}(t), \\ \hat{y}(t) = C\hat{x}(t) + D\hat{u}(t) \end{cases} \tag{2.35}$$

$$\text{Perturbed system} \begin{cases} \Delta\hat{x}(t+1) = A_K\hat{x}(t) + B_w w(t), \\ \Delta\hat{y}(t) = C_K\hat{x}(t) + D_w w(t) \end{cases} \tag{2.36}$$

For the perturbed system in (2.36), an RPI set $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ is constructed that satisfies the RPI inclusion

$$A_K\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \oplus B_w\boldsymbol{\mathcal{W}} \subseteq \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}). \tag{2.37}$$

Then, it can be observed that the state of System (2.32) belongs to a tube of cross-section $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ around the nominal system state $\hat{x}$, and the output of System (2.32) belongs to a tube of cross-section $C_K\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \oplus D_W\boldsymbol{\mathcal{W}}$ around the nominal system output $\hat{y}$, i.e.,

$$x(t) \in \hat{x}(t) \oplus \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \implies \begin{cases} x(t+1) \in \hat{x}(t+1) \oplus \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \\ y(t) \in \hat{y}(t) \oplus C_K\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \oplus D_w\boldsymbol{\mathcal{W}} \end{cases} \tag{2.38}$$

The RMPC scheme exploits this property to guarantee constraint satisfaction by enfocing the output constraint inclusion

$$\hat{y}(t) \oplus C_K\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \oplus D_w\boldsymbol{\mathcal{W}} \subseteq \boldsymbol{\mathcal{Y}}, \qquad \forall\, t \geq 0.$$

These constraints are enforced by solving, at each time-instant t the following Quadratic Program (QP) that is parameterized by the current state measurement $x(t)$ of the plant:

$$\begin{aligned}
\min_{\mathbf{z}} \ & \sum_{s=t}^{t+N-1} \|\hat{x}(s)\|_Q^2 + \|\hat{u}(s)\|_R^2 + \|\hat{x}(t+N)\|_P^2 && (2.39)\\
\text{s.t.} \ & x(t) \in \hat{x}(t) \oplus \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}), \\
& \hat{x}(s+1) = A\hat{x}(s) + B\hat{u}(s), && s \in \mathbb{I}_t^{t+N-1}, \\
& C\hat{x}(s) + D\hat{u}(s) \in \boldsymbol{\mathcal{Y}} \ominus (C_K \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \oplus D_w \boldsymbol{\mathcal{W}}), && s \in \mathbb{I}_{t+1}^{t+N-1}, \\
& \hat{x}(t+N) \in \boldsymbol{\mathcal{T}}(\boldsymbol{\mathcal{W}}),
\end{aligned}$$

where $\mathbf{z} := \{\hat{x}(t), \ldots, \hat{x}(t+N), \hat{u}(t), \ldots, \hat{u}(t+N-1)\}$. Uniquely, the initial nominal state $\hat{x}(t)$ is an optimization variable in the QP formulation, that is used to establish robust exponential stability of the RMPC scheme. If QP (2.39) is successfully solved, then the control input

$$u(t) = \hat{u}_*(t) + K(x(t) - \hat{x}_*(t))$$

is applied to the plant. Feasibility of the QP is ensured if the perturbation due to the disturbance is *not too large* with respect to the constraints, i.e.,

$$C_K \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \oplus D_w \boldsymbol{\mathcal{W}} \subseteq \operatorname{int}(\boldsymbol{\mathcal{Y}}).$$

Moreover, in order to guarantee recursive feasibility of the RMPC scheme, the terminal set $\boldsymbol{\mathcal{T}}(\boldsymbol{\mathcal{W}})$ is selected as a Positive invariant set for the nominal system controlled by the feedback law $\hat{u} = K\hat{x}$, i.e.,

$$A_K \boldsymbol{\mathcal{T}}(\boldsymbol{\mathcal{W}}) \subseteq \boldsymbol{\mathcal{T}}(\boldsymbol{\mathcal{W}}), \tag{2.40}$$

that is also constraint admissible, i.e.,

$$\boldsymbol{\mathcal{T}}(\boldsymbol{\mathcal{W}}) \subseteq \{\hat{x} \ : \ C_K \hat{x} \in \boldsymbol{\mathcal{Y}} \ominus (C_K \boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}}) \oplus D_w \boldsymbol{\mathcal{W}})\}. \tag{2.41}$$

Finally, the cost matrices (Q, R, P) along with the feedback gain K are assumed to satisfy the dissipativity condition

$$A_K^\top P A_K - P \preceq -(Q + K^\top R K)$$

to ensure robust exponential stability of the closed-loop system with the RMPC controller to the RPI set $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$.

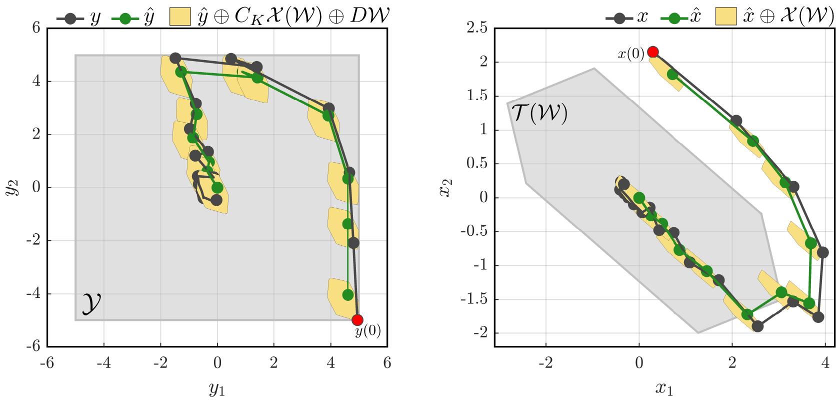


Figure 1: Closed-loop tube-based RMPC performance. The RPI set $\boldsymbol{\mathcal{X}}(\boldsymbol{\mathcal{W}})$ computed using methods from [144], and terminal set $\boldsymbol{\mathcal{T}}(\boldsymbol{\mathcal{W}})$ selected as the MPI set, computed using [44].

In Figure 1, we illustrate the controller performance on a randomly generated system with $\boldsymbol{\mathcal{Y}} = 5\mathcal{B}_\infty^2$. Further details of this system are given in Appendix A. The gray lines indicate the actual plant trajectories, and the green lines indicate the nominal trajectories. Observe that the nominal plant trajectories always belong in the RPI tube around the nominal trajectory. Thus, the RMPC scheme regulates the whole tube of trajectories instead of a single point to the RPI set instead of the origin, thus guaranteeing robust constraint satisfaction and stability.

Chapter 3

Computation of Input Disturbance Sets for Constrained Output Reachability

3.1 Introduction

The theory of set invariance plays an important role in the analysis of uncertain dynamical systems by providing the tools useful for the synthesis of robust controllers that can satisfy constraints in the presence of disturbances [16]. Of particular importance are RPI sets, the computation of which is a very active area of research as discussed in the introduction. These sets have successfuly been used in several applications such as Robust MPC [87, 116, 120], fault-tolerant control [104], state-observer design [90], etc. A key observation in these applications, however, is that they are developed under the assumption that *the disturbance set is known a priori*

In many practical cases, however, while the set of admissible states can be estimated from sensor measurements or pre-specified from given constraints to be satisfied, *the disturbance set is unknown*, leaving the designer the task of suitably defining it, especially in case one must satisfy a given set of constraints on the system, e.g., encoding known physical limitations, or undesired states. For example, in a decentralized MPC (DeMPC) application such as [120, 94], the dynamic coupling between subsystems is modeled as an additive disturbance. Then, the disturbance set for a given subsystem represents the state-constraint sets of the neighboring subsystems. Another example is presented in [39], in which the tracking references are modeled as disturbances acting on a system, such that a feasible disturbance set is the set of all feedforward tracking references guaranteeing constraint satisfaction. In both these cases, it is desirable to compute the *largest* feasible disturbance sets. In particular, a large disturbance set in the DeMPC case ensures that the region of attraction of the DeMPC scheme in which recursive feasibility and stability is guaranteed is maximized. Similarly, in the reference tracking case, a large disturbance set ensures that the operating range of the tracking control system is maximized.

In this chapter, we propose a method to tackle disturbance set computation problems such as those described above. In particular, we compute a set of disturbances such that the corresponding output reachable set approximately matches an assigned one. This method is centered on the formulation of an optimization problem, with the input disturbance set being the unknown and the approximation error between the obtained and assigned output set being the objective function to minimize. We propose the formulation of the optimization problem for stable linear systems and polytopic sets: since the construction of the output set requires the computation of an RPI set, we adopt the notions of [118, 144] to encode the computation of a parametrized RPI set within the problem. The chapter is organized as follows. We define the problem we solve in Problem 3.2. Then in Section 3.3, we present the main parameterizations we adopt, and present theoretical results that help us bring the problem into an implementable form. In Section 3.4, we develop a specialized smoothening-based interior-point solver for the resulting optimization problem.

3.2 Problem Definition

Consider the linear time-invariant discrete-time system

$$x(t+1) = Ax(t) + Bw(t), \tag{3.1a}$$
$$y(t) = Cx(t) + Dw(t), \tag{3.1b}$$

with state $x \in \mathbb{R}^{n_x}$, output $y \in \mathbb{R}^{n_y}$ and additive disturbance $w \in \mathbb{R}^{n_w}$. We assume that a set of output constraints is given:

$$\mathcal{Y} := \{y : Gy \le g\} \qquad \text{with } g \in \mathbb{R}^{m_Y}. \tag{3.2}$$

We define the reachable set of states from the origin, i.e., from $x(0) = \mathbf{0}$, under the action of disturbances $w(t) \in \mathcal{W}$ for all $t \ge 0$ in t-time steps as

$$\mathcal{X}(t, \mathcal{W}) := \left\{ x : x = \sum_{k=0}^{t-1} A^{t-k-1} Bw(k),\ \forall\, w(k) \in \mathcal{W} \right\},$$

and the corresponding set of t-step reachable outputs as

$$\mathcal{Y}(t, \mathcal{W}) := C\mathcal{X}(t, \mathcal{W}) \oplus D\mathcal{W}.$$

Observing that the reachable set of states satisfies the inclusion

$$\mathcal{X}(t, \mathcal{W}) \subseteq \mathcal{X}(t+1, \mathcal{W}), \qquad \forall\, t \ge 0,$$

if the disturbance set $\mathcal{W}$ is compact, convex, and contains the origin, we define the limit of reachable set of states as

$$\mathcal{X}_{\mathrm{m}}(\mathcal{W}) := \lim_{t \to \infty} \mathcal{X}(t, \mathcal{W}), \tag{3.3}$$

and the corresponding limit set of reachable outputs as

$$\mathcal{Y}_{\mathrm{m}}(\mathcal{W}) := C\mathcal{X}_{\mathrm{m}}(\mathcal{W}) \oplus D\mathcal{W}. \tag{3.4}$$

Then, our goal is to compute a disturbance set $\mathcal{W}$ that satisfies

$$\mathcal{Y}_{\mathrm{m}}(\mathcal{Y}) = \mathcal{Y}, \tag{3.5}$$

i.e., the reachable set of outputs is equal to the assigned set of outputs $\mathcal{Y}$. Unfortunately, satisfying the equality in (3.5) exactly might not be feasible in general. This is because the set $\mathcal{Y}$ is user-specified and hence can be of an arbitrary shape, while the reachable set of outputs $\mathcal{Y}_{\mathrm{m}}(\mathcal{Y})$ is defined by the system matrices (A, B, C, D) and the parameterization adopted to represent the disturbance set $\mathcal{W}$. Thus, we instead focus on computing a disturbance set $\mathcal{W}$ that satisfies the inclusion

$$\mathcal{Y}_{\mathrm{m}}(\mathcal{W}) \subseteq \mathcal{Y}, \tag{3.6}$$

and minimizes the distance between the reachable set of outputs $\mathcal{Y}_{\mathrm{m}}(\mathcal{W})$ and the assigned output set $\mathcal{Y}$. To this end, we tackle the optimization problem

$$\begin{aligned} \min_{\mathcal{W}} \ & \mathrm{d}_{\mathcal{Y}}(\mathcal{Y}_{\mathrm{m}}(\mathcal{W})) && \text{(3.7a)} \\ \text{s.t. } & \mathcal{Y}_{\mathrm{m}}(\mathcal{W}) \subseteq \mathcal{Y}, && \text{(3.7b)} \\ & \mathbf{0} \in \mathcal{W}, && \text{(3.7c)} \end{aligned}$$

where $\mathrm{d}_{\mathcal{Y}}(\mathcal{Y}_{\mathrm{m}}(\mathcal{W}))$ measures the disturbance between the sets $\mathcal{Y}$ and $\mathcal{Y}_{\mathrm{m}}(\mathcal{W})$, and Constraint (3.7b) enforces the desired inclusion in (3.6). Regarding the distance function $\mathrm{d}_{\mathcal{Y}}(\mathcal{Y}_{\mathrm{m}}(\mathcal{W}))$, a classical choice is to use the Hausdorff distance between the sets $\mathcal{Y}_{\mathrm{m}}(\mathcal{W})$ and $\mathcal{Y}$. In this chapter, we consider a slightly more general formulation with

$$\mathrm{d}_{\mathcal{Y}}(\mathcal{Y}_{\mathrm{m}}(\mathcal{W})) := \min_{\epsilon}\{\|\epsilon\|_1 : \mathcal{Y} \subseteq \mathcal{Y}_{\mathrm{m}}(\mathcal{W}) \oplus \mathbb{B}(\epsilon)\} \tag{3.8}$$

defined using the set $\mathbb{B}(\epsilon) := \{y : Hy \leq \epsilon\}$, in which normal vectors $\{H_i^\top, i \in \mathbb{I}_1^{n_B}\}$ are specified a priori by the user. Since $\mathrm{d}_{\mathcal{Y}}(\cdot)$ is monotonic, i.e, for all compact sets $\mathbf{S}_1, \mathbf{S}_2 \subseteq \mathcal{Y}$,

$$\mathbf{S}_1 \subseteq \mathbf{S}_2 \subseteq \mathcal{Y} \implies \mathrm{d}_{\mathcal{Y}}(\mathbf{S}_1) \geq \mathrm{d}_{\mathcal{Y}}(\mathbf{S}_2), \tag{3.9}$$

Problem (3.7) computes a disturbance set $\mathcal{W}$ that maximizes the coverage of $\mathcal{Y}$ by the reachable outputs while enforcing inclusion (3.6). Moreover, it prioritizes coverage of $\mathcal{Y}$ by the output reachable set $\mathcal{Y}_{\mathrm{m}}(\mathcal{W})$ in directions indicated by the normal vectors $H_i^\top$ to the set $\mathbb{B}(\epsilon)$.

Before tackling Problem (3.7), we observe that the reachable state set $\mathcal{X}_{\mathrm{m}}(\mathcal{W})$ in (3.3) is given by the infinite Minkowski sum

$$\mathcal{X}_{\mathrm{m}}(\mathcal{W}) = \bigoplus_{t=0}^{\infty} A^t B\mathcal{W}. \tag{3.10}$$

This implies that, except under very restrictive assumptions on $(A, B, \mathcal{W})$ [135], computing an exact finite-dimensional representation of the set $\mathcal{X}_{\mathrm{m}}(\mathcal{W})$, and hence $\mathcal{Y}_{\mathrm{m}}(\mathcal{W})$, is in general impossible. Thus, Problem (3.7) is in general impossible to solve exactly. This can, however, be ameliorated by adopting the notion of RPI sets, as we now explain. To this end, we make the following standing assumption of System (7.1).

Assumption 3.1. *System* (7.1) *is strictly stable, i.e.,* $\rho(A) < 1$. □

We know from [63] that under Assumption 3.1, if the disturbance set $\mathcal{W}$ is compact, convex and contains the origin, then the set $\mathcal{X}_{\mathrm{m}}(\mathcal{W})$ exists, is compact, convex, and contains the origin. By basic properties of the Minkowski sum, the set $\mathcal{Y}_{\mathrm{m}}(\mathcal{W})$ also inherits these properties. Furthermore, it was also shown in [63] that $\mathcal{X}_{\mathrm{m}}(\mathcal{W})$ is the smallest (in an inclusion sense) RPI set of System (3.1a), i.e., if a set $\mathcal{X}_{\mathrm{RPI}}(\mathcal{W})$ satisfies the RPI inclusion

$$A\mathcal{X}_{\mathrm{RPI}}(\mathcal{W}) \oplus B\mathcal{W} \subseteq \mathcal{X}_{\mathrm{RPI}}(\mathcal{W}), \tag{3.11}$$

then the set $\mathcal{X}_{\mathrm{m}}(\mathcal{W})$ is included in the RPI set, i.e.,

$$\mathcal{X}_{\mathrm{m}}(\mathcal{W}) \subseteq \mathcal{X}_{\mathrm{RPI}}(\mathcal{W}). \tag{3.12}$$

Hence, $\mathcal{X}_{\mathrm{m}}(\mathcal{W})$ is also referred to as the minimal RPI (mRPI) set. Then, defining a set of outputs corresponding to some RPI set $\mathcal{X}_{\mathrm{RPI}}(\mathcal{W})$ as

$$\mathcal{Y}_{\mathrm{RPI}}(\mathcal{W}) := C\mathcal{X}_{\mathrm{RPI}}(\mathcal{W}) \oplus D\mathcal{W} \supseteq \mathcal{Y}_{\mathrm{m}}(\mathcal{W}), \tag{3.13}$$

the desired output constraint inclusion in (3.6) formulating Constraint (3.7b) can be enforced through

$$\mathcal{Y}_{\mathrm{RPI}}(\mathcal{W}) \subseteq \mathcal{Y}. \tag{3.14}$$

Thus, we propose to tackle Problem (3.7) by replacing the output reachable set $\mathcal{Y}_{\mathrm{m}}(\mathcal{W})$ with some outer-approximating set $\mathcal{Y}_{\mathrm{RPI}}(\mathcal{W})$ defined using some suitable finite-dimensional RPI set $\mathcal{X}_{\mathrm{RPI}}(\mathcal{W})$.

From inclusion (3.13) and the monotonicity property of the distance function $\mathrm{d}_{\mathcal{Y}}(\cdot)$ in (3.9), it follows that

$$\mathrm{d}_{\mathcal{Y}}(\mathcal{Y}_{\mathrm{RPI}}(\mathcal{W})) \leq \mathrm{d}_{\mathcal{Y}}(\mathcal{Y}_{\mathrm{m}}(\mathcal{W})) \tag{3.15}$$

for any disturbance set $\mathcal{W}$ and finite-dimensional RPI set $\mathcal{X}_{\mathrm{RPI}}(\mathcal{W})$ satisfying inclusion (3.13). This implies that if we replace $\mathcal{Y}_{\mathrm{m}}(\mathcal{W})$ in the objective of Problem (3.7) with $\mathcal{Y}_{\mathrm{RPI}}(\mathcal{W})$, then we would be minimizing a lower-bound, which is undesirable. Hence, we propose to instead select some index $l > 0$, and minimize the distance between the l-step output reachable set and the output constraint set $\mathcal{Y}$. In other words, for some user-specified index $l > 0$, we define the l-step reachable set as

$$\mathcal{S}(l, \mathcal{W}) := \bigoplus_{t=0}^{l-1} CA^t B\mathcal{W} \oplus D\mathcal{W}. \tag{3.16}$$

Then observing that for any $l > 0$, by monotonicity the distance function satisfies

$$\mathrm{d}_{\mathcal{Y}}(\mathcal{Y}_{\mathrm{m}}(\mathcal{W})) \leq \mathrm{d}_{\mathcal{Y}}(\mathcal{S}(l, \mathcal{W})), \tag{3.17}$$

we propose to tackle the following optimization problem to minimize an upper-bound to Problem (3.7):

$$\min_{\mathcal{W}} \ \mathrm{d}_{\mathcal{Y}}(\mathcal{S}(l, \mathcal{W})) \tag{3.18a}$$

$$\text{s.t. } \mathcal{Y}_{\mathrm{RPI}}(\mathcal{W}) \subseteq \mathcal{Y}, \tag{3.18b}$$

$$\mathbf{0} \in \mathcal{W}. \tag{3.18c}$$

The main questions to tackle in order to solve Problem (3.18) are

1. How to choose a representation for the disturbance set $\mathcal{W}$?

2. How to choose a representation of the RPI set $\mathcal{X}_{\mathrm{RPI}}(\mathcal{W})$ required to formulate Constraint (3.18b)?

3. How to solve the resulting optimization problem?

In the rest of this chapter, we tackle these questions using an Explicit RPI framework, i.e., we introduce the RPI set $\mathcal{X}_{\mathrm{RPI}}(\mathcal{W})$ as a decision variable into Problem (3.18).

Remark 3.1. *Since the set $\mathcal{X}_{\mathrm{RPI}}(\mathcal{W})$ is RPI, with persistent disturbances $w(t) \in \mathcal{W}, \forall\, t \geq 0$ from any disturbance set $\mathcal{W}$ feasible for Problem* (3.18)*, we are guaranteed that the output of System* (7.1) *satisfies the constraints $y(t) \in \mathcal{Y}, \forall\, t \geq 0$ from any initial state $x(0) \in \mathcal{X}_{\mathrm{RPI}}(\mathcal{W})$, and is not restricted to $x(0) = \mathbf{0}$.* $\square$.

3.3 Explicit RPI approach to solve Problem (3.18)

In this approach, we focus on the computation of a disturbance set $\mathcal{W}$ parametrized as the polytope

$$\mathbb{W}(\epsilon^w) := \{w \ : \ F_t w \leq \epsilon^w_t, \ \ \forall\, t \in \mathbb{I}_1^{m_W}\} = \{w : Fw \leq \epsilon^w\}. \tag{3.19}$$

We assume that the normal vectors $\{F_t^\top \in \mathbb{R}^{n_w}, t \in \mathbb{I}_1^{m_W}\}$ to $\mathbb{W}(\epsilon^w)$ are given a priori, and restrict our attention to computing the vector ϵ^w. Given the disturbance set parameterization in (3.19), the constraint $\mathbf{0} \in \mathcal{W}$ can be enforced simply as $\epsilon^w \geq \mathbf{0}$.

Nonconvexity due to parameterization (3.19)

Before we proceed with formulating Problem (3.7) with the polytopic disturbance set parameterization in (3.19) and then approximating it as Problem (3.18), we present a brief discussion regarding the inherent nonconvexity associated with this parameterization. To this end, we first write out Constraint (3.7b) as

$$\mathcal{Y}_{\mathrm{m}}(\epsilon^w) = \bigoplus_{t=0}^{\infty} CA^t B\mathbb{W}(\epsilon^w) \oplus D\mathbb{W}(\epsilon^w) \tag{3.20}$$

by recalling the definition of the mRPI set from (3.10) and the output reachable set $\mathcal{Y}_{\mathrm{m}}(\cdot)$ from (3.4). Note that we slightly abuse notation with using $\mathcal{Y}_{\mathrm{m}}(\epsilon^w)$ instead of $\mathcal{Y}_{\mathrm{m}}(\mathbb{W}(\epsilon^w))$. This is because we assume that the matrix F is fixed a priori. Then, we define the set of all feasible disturbance set parameters ϵ^w as

$$\mathcal{O}_{\mathcal{Y}}(F) := \{\epsilon^w \geq \mathbf{0} : \mathcal{Y}_{\mathrm{m}}(\epsilon^w) \subseteq \mathcal{Y}\}.$$

By properties of support functions, $\epsilon^w \in \mathcal{O}_{\mathcal{Y}}(F)$ if and only if

$$\sum_{t=0}^{\infty} h_{\mathbb{W}(\epsilon^w)}((G_j C A^t B)^\top) + h_{\mathbb{W}(\epsilon^w)}((G_j D)^\top) \leq g_j, \qquad \forall\, j \in \mathbb{I}_1^{m_Y}. \quad (3.21)$$

Considering two feasible vectors $\epsilon^{w,1}, \epsilon^{w,2} \in \mathcal{O}_{\mathcal{Y}}(F)$, a scalar $\zeta \in [0,1]$, and the convex combination

$$\tilde{\epsilon}^w := \zeta \epsilon^{w,1} + (1-\zeta)\epsilon^{w,2},$$

the inclusion

$$\mathbb{W}(\zeta\epsilon^{w,1}) \oplus \mathbb{W}((1-\zeta)\epsilon^{w,2}) \subseteq \mathbb{W}(\tilde{\epsilon}^w)$$

holds for a general disturbance set parameterizing matrix F (from duality in Linear Programing (LP)). This implies that for any arbitrary vector $r \in \mathbb{R}^{n_w}$, the support function inequality

$$h_{\mathbb{W}(\zeta\epsilon^{w,1})}(r) + h_{\mathbb{W}((1-\zeta)\epsilon^{w,2})}(r) \leq h_{\mathbb{W}(\tilde{\epsilon}^w)}(r)$$

holds, such that $\tilde{\epsilon}^w$ does not necessarily belong to $\mathcal{O}_{\mathcal{Y}}(F)$. Hence, *Problem* (3.7) *is in general nonconvex*. However, as we show in the following result, there exist special cases of parametrization of F for which convexity of the problems holds.

Proposition 3.1. *Suppose that the disturbance set $\mathbb{W}(\epsilon^w)$ is parametrized with $F = [\tilde{F}^\top \;\; -\tilde{F}^\top]^\top$, where $\tilde{F} \in \mathbb{R}^{n_w \times n_w}$, and the matrix $\hat{F} := (\tilde{F}\tilde{F}^\top)^{-1}\tilde{F}$ is well-defined. Then, $\mathcal{O}_{\mathcal{Y}}(F)$ is convex.* □

Proof. We note that if $h_{\mathbb{W}(\zeta\epsilon^w)}(F_t^\top) = \zeta\epsilon_t^w$ for all $t \in \mathbb{I}_1^{m_W}$, then

$$\mathbb{W}(\zeta\epsilon^{w,1}) \oplus \mathbb{W}((1-\zeta)\epsilon^{w,2}) = \mathbb{W}(\tilde{\epsilon}^w)$$

holds for all $\epsilon^{w,1}, \epsilon^{w,2} \in \mathcal{O}_{\mathrm{y}}(F)$, $\zeta \in [0,1]$ and $\tilde{\epsilon}^w = \zeta\epsilon^{w,1} + (1-\zeta)\epsilon^{w,2}$, such that $\tilde{\epsilon}^w \in \mathcal{O}_{\mathrm{y}}(F)$.

Hence, we show in the sequel that the support function equality

$$h_{\mathbb{W}(\epsilon^w)}(F_t^\top) = \left\{ \begin{array}{ll} \max\limits_{w} & F_t w \\ \text{s.t.} & Fw \leq \epsilon^w \end{array} \right\} = \epsilon_t^w$$

holds for all $t \in \mathbb{I}_1^{mW}$ under the assumption on F. Since $h_{\mathbb{W}(\epsilon^w)}(F_t^\top) \leq \epsilon_t^w$ in general, it is sufficient to show the existence of a primal variable $\boldsymbol{w}$ satisfying $F_t\boldsymbol{w} = \epsilon_t^w$, along with dual variables $[\boldsymbol{\lambda}^\top \ \boldsymbol{\mu}^\top]^\top \geq \mathbf{0}_{2n_w}$ and slack variables $\boldsymbol{s} \geq \mathbf{0}_{2n_w}$ satisfying the LP optimality conditions [102]

$$\tilde{F}^\top \boldsymbol{\lambda} - \tilde{F}^\top \boldsymbol{\mu} = F_t^\top, \qquad F\boldsymbol{w} + \boldsymbol{s} = \epsilon^w, \qquad [\boldsymbol{\lambda}^\top \ \boldsymbol{\mu}^\top]^\top \circ \boldsymbol{s} = \mathbf{0}. \tag{3.22}$$

We first write the dual feasible condition as $\boldsymbol{\lambda} = \boldsymbol{\mu} + \hat{F}F_t^\top$ by multiplying $\hat{F}$ on both sides, and denote column t of $\mathbf{I}_{n_w}$ by $e_{[t]}$. Then, by definition of $\hat{F}$, and have $\hat{F}F_t^\top = e_{[t]}$ if $t \in \mathbb{I}_1^{n_w}$, and $\hat{F}F_t^\top = -e_{[t-n_w]}$ if $t \in \mathbb{I}_{n_w+1}^{2n_w}$. Then, setting the dual variables $\boldsymbol{\lambda} = e_{[t]}$ and $\boldsymbol{\mu} = \mathbf{0}$ for all $t \in \mathbb{I}_1^{n_w}$, and $\boldsymbol{\mu} = e_{[t-n_w]}$ and $\boldsymbol{\lambda} = \mathbf{0}$ for all $t \in \mathbb{I}_{n_w+1}^{2n_w}$, we note that $[\boldsymbol{\lambda}^\top \ \boldsymbol{\mu}^\top]^\top \circ \boldsymbol{s} = \mathbf{0}$ implies $\boldsymbol{s}_t = 0$, or equivalently that the primal variable $\boldsymbol{w}$ satisfies $F_t\boldsymbol{w} = \epsilon_t^w$. □

In the Explicit RPI approach, we focus on general polytopic parametrizations of $\mathbb{W}(\epsilon^w)$, for which Problem (3.7) is generally nonconvex. This is motivated primarily by the fact that parametrizations of $\mathbb{W}(\epsilon^w)$ that ensure convexity of Problem (3.7) might be excessively conservative in certain applications, e.g. decentralized MPC [94], in which the disturbance sets represent state-constraint sets of dynamically coupled subsystems.

3.3.1 Polytopic parameterization of the RPI set

We propose to use a RPI set $\mathcal{X}_{\text{RPI}}(\epsilon^w)$ parameterized as the polytope

$$\mathbb{X}(\epsilon^x) := \{x \ : \ E_i x \leq \epsilon_i^x, \ \forall\, i \in \mathbb{I}_1^{mX}\} = \{x : Ex \leq \epsilon^x\} \tag{3.23}$$

to formulate Problem (3.18), in which we assume that the normal vectors $\{E_i^\top \in \mathbb{R}^{n_x}, i \in \mathbb{I}_1^{mX}\}$ to $\mathbb{X}(\epsilon^x)$ are given a priori. In order to formulate Problem (3.18) to compute a disturbance set $\mathbb{W}(\epsilon^w)$ using the RPI set

$\mathbb{X}(\epsilon^x)$, we first introduce the support functions

$$\forall\, i \in \mathbb{I}_1^{mx}, \begin{cases} \boldsymbol{c}_i(\epsilon^x) & := h_{A\mathbb{X}(\epsilon^x)}(E_i^\top), \\ \boldsymbol{d}_i(\epsilon^w) & := h_{B\mathbb{W}(\epsilon^w)}(E_i^\top), \\ \boldsymbol{b}_i(\epsilon^x) & := h_{\mathbb{X}(\epsilon^x)}(E_i^\top). \end{cases}$$

We then recall that the set $\mathbb{X}(\epsilon^x)$ is RPI for a given disturbance set $\mathbb{W}(\epsilon^w)$ if and only if it verifies the inclusion

$$A\mathbb{X}(\epsilon^x) \oplus B\mathbb{W}(\epsilon^w) \subseteq \mathbb{X}(\epsilon^x), \tag{3.24}$$

which can be written using the support functions introduced above as

$$\boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) \leq \boldsymbol{b}(\epsilon^x). \tag{3.25}$$

We then define the set of all vectors ϵ^x characterizing an RPI set $\mathbb{X}(\epsilon^x)$ for a given disturbance set parameter $\epsilon^w \geq \mathbf{0}$ as

$$\mathcal{E}(\boldsymbol{d}(\epsilon^w)) := \{\epsilon^x \geq \mathbf{0} : \boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) \leq \boldsymbol{b}(\epsilon^x)\}. \tag{3.26}$$

In the definition of $\mathcal{E}(\boldsymbol{d}(\epsilon^w))$, we enforce $\epsilon^x \geq \mathbf{0}$ since every RPI set $\mathbb{X}(\epsilon^x)$ corresponding to a disturbance set $\mathbb{W}(\epsilon^w)$ with $\epsilon^w \geq \mathbf{0}$ contains the origin, such that $\epsilon^x \geq \mathbf{0}$. Using the set $\mathcal{E}(\boldsymbol{d}(\epsilon^w))$, we define

$$\begin{aligned} \epsilon^x(\epsilon^w) := \arg\min_{\underline{\epsilon}^x} \quad & \mathrm{d_H}(\mathcal{X}_\mathrm{m}(\epsilon^w), \mathbb{X}(\underline{\epsilon}^x)) \\ \text{s.t.} \quad & \underline{\epsilon}^x \in \mathcal{E}(\boldsymbol{d}(\epsilon^w)), \end{aligned} \tag{3.27}$$

where $\mathrm{d_H}(\mathcal{X}_\mathrm{m}(\epsilon^w), \mathbb{X}(\epsilon^x))$ is the Hausdorff distance between the mRPI set $\mathcal{X}_\mathrm{m}(\epsilon^w)$ and the set $\mathbb{X}(\epsilon^x)$, such that for a given disturbance set parameter $\epsilon^w \geq \mathbf{0}$, the set $\mathbb{X}(\epsilon^x(\epsilon^w))$ is the tightest RPI approximation parameterized as the polytope $\mathbb{X}(\epsilon^x)$ to the mRPI set $\mathcal{X}_\mathrm{m}(\epsilon^w)$. We then propose to use the set $\mathbb{X}(\epsilon^x(\epsilon^w))$ as the RPI set $\mathcal{X}_\mathrm{RPI}(\mathbb{W}(\epsilon^w))$ in the formulation of Problem (3.18).

Then, in order to encode Constraint (3.18b) for the disturbance set $\mathbb{W}(\epsilon^w)$ and RPI set $\mathbb{X}(\epsilon^x(\epsilon^w))$, we first define the associated output set as

$$\mathbb{Y}(\epsilon^x(\epsilon^w), \epsilon^w) := C\mathbb{X}(\epsilon^x(\epsilon^w)) \oplus D\mathbb{W}(\epsilon^w), \tag{3.28}$$

that is equivalent to the set $\mathcal{Y}_{\mathrm{RPI}}(\epsilon^w)$ in the definition of Problem (3.18). Then, we define the support functions

$$\forall\, k \in \mathbb{I}_1^{m_Y},\ \begin{cases} \boldsymbol{l}_k(\epsilon^x) := h_{C\mathbb{X}(\epsilon^x)}(G_k^\top), \\ \boldsymbol{m}_k(\epsilon^w) := h_{D\mathbb{W}(\epsilon^w)}(G_k^\top). \end{cases}$$

using which we encode the constraint $\mathbb{Y}(\epsilon^x(\epsilon^w), \epsilon^w) \subseteq \mathcal{Y}$ as

$$\boldsymbol{l}(\epsilon^x(\epsilon^w)) + \boldsymbol{m}(\epsilon^w) \le g. \tag{3.29}$$

Hence, for the disturbance parameterization $\mathbb{W}(\epsilon^w)$ and RPI set parameterization $\mathbb{X}(\epsilon^x)$, we propose to formulate Problem (3.18) as the bilevel optimization problem

$$\begin{aligned} \min_{\epsilon^w}\quad & \mathrm{d}_{\mathcal{Y}}(\mathcal{S}(l, \mathbb{W}(\epsilon^w))) && \text{(3.30a)} \\ \text{s.t.}\quad & \boldsymbol{l}(\epsilon^x) + \boldsymbol{m}(\epsilon^w) \le g, && \text{(3.30b)} \\ & \epsilon^w \ge \mathbf{0}, && \text{(3.30c)} \\ & \epsilon^x := \arg\min_{\underline{\epsilon}^x}\ \mathrm{d}_{\mathrm{H}}(\mathcal{X}_{\mathrm{m}}(\epsilon^w), \mathbb{X}(\underline{\epsilon}^x)) \quad \text{s.t.} \quad \underline{\epsilon}^x \in \mathcal{E}(\boldsymbol{d}(\epsilon^w)). && \text{(3.30d)} \end{aligned}$$

In the definition of Problem (3.30), we drop the dependence of ϵ^x on ϵ^w for simplicity of notation. Since we explicitly compute the vector ϵ^x in Problem (3.30) that computing an explicit representation of the RPI set $\mathbb{X}(\epsilon^x)$, we refer to the current approach as an Explicit RPI (ERPI) approach.

In the rest of this section, we focus on the development of methods to solve Problem (3.30). Before we do so, we discuss the rationale behind the formulation of Problem (3.30). To this end, we first note that a conventional formulation of Problem (3.18) follows for a disturbance set $\mathbb{W}(\epsilon^w)$ and an RPI set $\mathbb{X}(\epsilon^x)$ as

$$\begin{aligned} \min_{\epsilon^x, \epsilon^w}\quad & \mathrm{d}_{\mathcal{Y}}(\mathcal{S}(l, \mathbb{W}(\epsilon^w))) && \text{(3.31a)} \\ \text{s.t.}\quad & \boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) \le \boldsymbol{b}(\epsilon^x), && \text{(3.31b)} \\ & \boldsymbol{l}(\epsilon^x) + \boldsymbol{m}(\epsilon^w) \le g, && \text{(3.31c)} \\ & \epsilon^w \ge \mathbf{0}, && \text{(3.31d)} \end{aligned}$$

in which instead of Constraint (3.30d), Constraint (3.31b) simply enforces that $\mathbb{X}(\epsilon^x)$ is an RPI set corresponding to $\mathbb{W}(\epsilon^w)$. Then, the formulation of Problem (3.18) is justified only if it is not more conservative that Problem (3.31). To this end, we recall from [118] that given some $\epsilon^w \geq \mathbf{0}$, the set $\mathbb{X}(\epsilon^x(\epsilon^w))$ is the smallest RPI set contained in all RPI sets $\mathbb{X}(\epsilon^x)$ parameterized with fixed normal vectors defining the matrix E, i.e.,

$$\mathbb{X}(\epsilon^x(\epsilon^w)) \subseteq \mathbb{X}(\epsilon^x), \qquad\qquad \forall\, \epsilon^x \in \mathcal{E}(\boldsymbol{d}(\epsilon^w)).$$

By basic properties of Minkowski sums and monotonicity of support functions, it follows that

$$\boldsymbol{l}(\epsilon^x(\epsilon^w)) \oplus \boldsymbol{m}(\epsilon^w) \leq \boldsymbol{l}(\epsilon^x) \oplus \boldsymbol{m}(\epsilon^w), \qquad \forall\, \epsilon^x \in \mathcal{E}(\boldsymbol{d}(\epsilon^w)).$$

This implies any solution ϵ^w of Problem (3.31) is feasible for Problem (3.30), such that the optimal value of Problem (3.30) is no worse than that of Problem (3.31). On the other hand, any solution $(\epsilon^x(\epsilon^w), \epsilon^w)$ of Problem (3.30) is feasible for Problem (3.31), since $\epsilon^x = \epsilon^x(\epsilon^w)$ satisfies the RPI condition. However, the constraint sets of Problems (3.30) and (3.31) are nonconvex since support functions are concave in their argument. Hence, it is justified to directly use $\epsilon^x = \epsilon^x(\epsilon^w)$ to while formulating the problem, since it implies that we optimize directly using the RPI set $\mathbb{X}(\epsilon^x)$ that will be the least conservative for a given $\mathbb{W}(\epsilon^w)$. As we will show in the sequel, this is also advantageous from an implementation viewpoint.

3.3.2 Existence conditions for a polytopic RPI set

Since ϵ^w is an optimization variable in Problem (3.30), we must ensure that for every feasible $\epsilon^w \geq \mathbf{0}$, there exists an RPI set parameterized as $\mathbb{X}(\epsilon^x)$. In other words, we must ensure that the set $\mathcal{E}(\boldsymbol{d}(\epsilon^w))$ characterizing the vectors ϵ^x defining RPI sets $\mathbb{X}(\epsilon^x)$ for a given disturbance set $\mathbb{W}(\epsilon^w)$ is nonempty for all feasible $\epsilon^w \geq \mathbf{0}$. Hence, we now formulate the requirements that matrix E must satisfy to ensure that $\mathcal{E}(\boldsymbol{d}(\epsilon^w))$ is nonempty for every $\epsilon^w \geq \mathbf{0}$.

Assumption 3.2. *Matrix E is chosen such $\boldsymbol{b}(\mathbf{1}) = \mathbf{1}$, and there exists an $\hat{\epsilon}^x \geq \mathbf{0}$ satisfying the inequality $\boldsymbol{c}(\hat{\epsilon}^x) + \mathbf{1} \leq \boldsymbol{b}(\hat{\epsilon}^x)$.* □

Assumption 3.2 implies that there exists an RPI set $\mathbb{X}(\hat{\epsilon}^x)$ for the system $x(t+1) = Ax(t) + \tilde{w}(t)$ with $\tilde{w} \in \mathbb{X}(\mathbf{1})$. In the following result, we show that there always exists an RPI set $\mathbb{X}(\epsilon^x)$ for system (3.1a) with the disturbance set $\mathbb{W}(\epsilon^w)$ under Assumption 3.2.

Proposition 3.2. *If Assumption 3.2 holds, then there always exists an $\epsilon^x \geq \mathbf{0}$ satisfying $\boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) \leq \boldsymbol{b}(\epsilon^x)$ for all $\epsilon^w \geq \mathbf{0}$.* □

Proof. Under Assumption 3.2, there exist nonnegative multipliers variables $\hat{\Lambda}_{\mathbf{c}} \in \mathbb{R}^{m_X \times m_X}$ and $\hat{\Lambda}_{\mathbf{b}} \in \mathbb{R}^{m_X \times m_X}$ satisfying

$$\hat{\Lambda}_{\mathbf{c}}^\top \hat{\epsilon}_x + \mathbf{1} \leq \hat{\Lambda}_{\mathbf{b}}^\top \hat{\epsilon}_x, \qquad \hat{\Lambda}_{\mathbf{c}}^\top E = EA, \qquad \hat{\Lambda}_{\mathbf{b}}^\top E = E,$$

by LP duality and Farkas' lemma [16]. There exists an $\epsilon^x \geq \mathbf{0}$ satisfying the RPI condition

$$\boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) \leq \boldsymbol{b}(\epsilon^x)$$

for any given $\epsilon^w \geq \mathbf{0}$ if and only if there exist nonnegative multiplier variables $\Lambda_{\mathbf{c}}$ and $\Lambda_{\mathbf{b}}$ satisfying

$$\Lambda_{\mathbf{c}}^\top \epsilon_x + \boldsymbol{d}(\epsilon^w) \leq \Lambda_{\mathbf{b}}^\top \epsilon_x, \qquad \Lambda_{\mathbf{c}}^\top E = EA, \qquad \Lambda_{\mathbf{b}}^\top E = E.$$

The proof is concluded by noting that $\epsilon^x = ||\boldsymbol{d}(\epsilon^w)||_\infty \hat{\epsilon}^x$, $\Lambda_{\mathbf{c}} = \hat{\Lambda}_{\mathbf{c}}$, and $\Lambda_{\mathbf{b}} = \hat{\Lambda}_{\mathbf{b}}$ satisfy these conditions. □

Remark 3.2. *Assumption 3.2 can be verified by checking the boundedness of LP (8) in [144]. An iterative procedure to obtain a matrix E that verifies Assumption 3.2 was presented in [81].* □

3.3.3 Elimination of $\mathcal{X}_{\mathrm{m}}(\epsilon^w)$ from (3.30d)

Having established that $\mathcal{E}(\boldsymbol{d}(\epsilon^w))$ is nonempty for any $\epsilon^w \geq \mathbf{0}$ under Assumption 3.2, we will now eliminate the mRPI set $\mathcal{X}_{\mathrm{m}}(\epsilon^w)$ from Problem (3.30d). To this end, we recall the following results from [118] (specialized to the case of an autonomous stable LTI system), which state that the solution of Problem (3.30d) can be obtained using fixed-point iterations for a given $\epsilon^w \geq \mathbf{0}$. We denote $\boldsymbol{d}(\epsilon^w)$ by $\boldsymbol{d}$ for ease of notation.

Lemma 3.1. *[118, Theorems 1 and 2, Corollary 1] Suppose Assumption 3.2 holds, and*

$$\mathcal{H}(\boldsymbol{d}) := \{\epsilon^x : \mathbf{0} \leq \epsilon^x \leq \|\boldsymbol{d}\|_\infty \hat{\epsilon}^x\}.$$

Then, for any $\epsilon^w \geq \mathbf{0}$, the following results hold:

1. *For all* $\underline{\epsilon}^x \in \mathcal{H}(\boldsymbol{d})$*, it holds that* $\boldsymbol{c}(\underline{\epsilon}^x)+\boldsymbol{d} \in \mathcal{H}(\boldsymbol{d})$*, and there exists atleast one solution* $\epsilon_*^x(\boldsymbol{d}) \in \mathcal{H}(\boldsymbol{d})$ *for the fixed-point equations*

$$\boldsymbol{c}(\epsilon_*^x(\boldsymbol{d})) + \boldsymbol{d} = \boldsymbol{b}(\epsilon_*^x(\boldsymbol{d})), \quad \boldsymbol{b}(\epsilon_*^x(\boldsymbol{d})) = \epsilon_*^x(\boldsymbol{d}).$$

   Hence, the set of all fixed-point solutions

$$\mathcal{R}(\boldsymbol{d}) := \{\epsilon^x \in \mathcal{H}(\boldsymbol{d}) : \boldsymbol{c}(\epsilon^x) + \boldsymbol{d} = \boldsymbol{b}(\epsilon^x),\ \boldsymbol{b}(\epsilon^x) = \epsilon^x\}$$

   is nonempty;

2. *Starting from the initial-condition* $\epsilon_{[0]}^x = \boldsymbol{0}$*, the sequence generated by the iterations* $\epsilon_{[k+1]}^x := \boldsymbol{c}(\epsilon_{[k]}^x) + \boldsymbol{d}$ *converges to a fixed-point solution*

$$\lim_{k\to\infty} \epsilon_{[k]}^x := \epsilon_*^x(\boldsymbol{0}, \boldsymbol{d}) \in \mathcal{R}(\boldsymbol{d}).$$

   Moreover, $\epsilon_*^x(\boldsymbol{0}, \boldsymbol{d})$ *is the minimal fixed-point, i.e.,*

$$\epsilon_*^x(\boldsymbol{0}, \boldsymbol{d}) \leq \underline{\epsilon}^x,\ \forall\, \underline{\epsilon}^x \in \mathcal{R}(\boldsymbol{d}) \subseteq \mathcal{E}(\boldsymbol{d}).$$

   Consequently, the set $\mathbb{X}(\epsilon_*^x(\boldsymbol{0}, \boldsymbol{d}))$ *satisfies satisfies*

$$\mathcal{X}_{\mathrm{m}}(\epsilon^w) \subseteq \mathbb{X}(\epsilon_*^x(\boldsymbol{0}, \boldsymbol{d})) = \bigcap_{\underline{\epsilon}^x \in \mathcal{E}(\boldsymbol{d})} \mathbb{X}(\underline{\epsilon}^x),$$

   and hence is the minimal parametrized RPI set.

□

From Lemma 3.1.2, we see that $\epsilon_*^x(\boldsymbol{0}, \boldsymbol{d}(\epsilon^w))$ is the solution of the Problem (3.30d), since the RPI set $\mathbb{X}(\epsilon_*^x(\boldsymbol{0}, \boldsymbol{d}(\epsilon^w)))$ satisfies the inequality

$$d_{\mathrm{H}}(\mathcal{X}_{\mathrm{m}}(\epsilon^w), \mathbb{X}(\epsilon_*^x(\boldsymbol{0}, \boldsymbol{d}(\epsilon^w)))) \leq d_{\mathrm{H}}(\mathcal{X}_{\mathrm{m}}(\epsilon^w), \mathbb{X}(\underline{\epsilon}^x))$$

over all $\underline{\epsilon}^x \in \mathcal{E}(\boldsymbol{d}(\epsilon^w))$. Since this solution also satisfies $\epsilon_*^x(\boldsymbol{0}, \boldsymbol{d}(\epsilon^w)) \leq \underline{\epsilon}^x$ over all feasible $\underline{\epsilon}^x \in \mathcal{E}(\boldsymbol{d}(\epsilon^w))$, it has the smallest norm-1 value over all feasible $\underline{\epsilon}^x \in \mathcal{E}(\boldsymbol{d}(\epsilon^w))$. Hence, we write Problem (3.30d) equivalently as

$$\begin{aligned} \epsilon^x = \arg\min_{\underline{\epsilon}^x} \quad & \|\underline{\epsilon}^x\|_1 \\ \text{s.t.} \quad & \underline{\epsilon}^x \in \mathcal{E}(\boldsymbol{d}(\epsilon^w)). \end{aligned} \tag{3.32}$$

Thus, in the rest of this section, we tackle Problem (3.30) formulated with constraint (3.30d) replaced by (3.32) that is independent of the mRPI set $\mathcal{X}_{\mathrm{m}}(\epsilon^w)$. This results in Problem (3.30) being equivalent to

$$\min_{\epsilon^w} \quad \mathrm{d}_{\mathcal{Y}}(\mathcal{S}(l, \mathbb{W}(\epsilon^w))) \tag{3.33a}$$

$$\text{s.t.} \quad \boldsymbol{l}(\epsilon^x) + \boldsymbol{m}(\epsilon^w) \leq g, \tag{3.33b}$$

$$\epsilon^w \geq \mathbf{0}, \tag{3.33c}$$

$$\epsilon^x := \arg\min_{\underline{\epsilon}^x} \quad \|\underline{\epsilon}^x\|_1 \quad \text{s.t.} \quad \underline{\epsilon}^x \in \mathcal{E}(\boldsymbol{d}(\epsilon^w)). \tag{3.33d}$$

In the sequel, we transform Problem (3.33) into an implementable form under the following feasibility assumption on output constraint set $\mathcal{Y}$, which results in $\epsilon^w = 0$ being a feasible solution.

Assumption 3.3. *The origin belongs to the output constraint set $\mathcal{Y}$, such that $g \geq \mathbf{0}$.* □

3.3.4 Characterization of RPI Constraints

In this subsection, we show that the minimal parametrized RPI constraint (3.32) in Problem (3.33) can be replaced by the equality $\boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) = \epsilon^x$, i.e., the equivalence

$$(3.32) \iff \boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) = \epsilon^x \tag{3.34}$$

holds. For simplicity, we denote $\boldsymbol{d}(\epsilon^w)$ by $\boldsymbol{d}$ in the sequel, since the results are presented for a fixed $\epsilon^w \geq \mathbf{0}$. We recall from Lemma 3.1 that the fixed-point solution

$$\epsilon^x_*(\mathbf{0}, \boldsymbol{d}) = \arg\min_{\epsilon^x} \quad \|\underline{\epsilon}^x\|_1 \quad \text{s.t. } \underline{\epsilon}^x \in \mathcal{E}(\boldsymbol{d})\},$$

exists, and satisfies the fixed-point equations

$$\boldsymbol{c}(\epsilon^x_*(\mathbf{0}, \boldsymbol{d})) + \boldsymbol{d} = \boldsymbol{b}(\epsilon^x_*(\mathbf{0}, \boldsymbol{d})) = \epsilon^x_*(\mathbf{0}, \boldsymbol{d}).$$

We recall further that $\mathcal{R}(\boldsymbol{d}) \subseteq \mathcal{E}(\boldsymbol{d})$ is the set of all fixed-points, i.e.,

$$\mathcal{R}(\boldsymbol{d}) := \{\epsilon^x \in \mathcal{H}(\boldsymbol{d}) : \boldsymbol{c}(\epsilon^x) + \boldsymbol{d} = \boldsymbol{b}(\epsilon^x),\ \boldsymbol{b}(\epsilon^x) = \epsilon^x\}.$$

Then, if there exists a unique fixed-point $\epsilon^x_\#(\boldsymbol{d}) \in \mathcal{R}(\boldsymbol{d})$, we will then have that $\epsilon^x_*(\mathbf{0}, \boldsymbol{d}) = \epsilon^x_\#(\boldsymbol{d})$. Moreover, since we know that

$$\boldsymbol{b}(\boldsymbol{c}(\epsilon^x) + \boldsymbol{d}) = \boldsymbol{c}(\epsilon^x) + \boldsymbol{d}$$

for every $\epsilon^x \in \mathcal{E}(\boldsymbol{d})$ from [118, Proposition 1], every $\epsilon^x \in \mathcal{E}(\boldsymbol{d})$ that satisfies $\boldsymbol{c}(\epsilon^x) + \boldsymbol{d} = \epsilon^x$ satisfies $\boldsymbol{b}(\epsilon^x) = \epsilon^x$. Hence, the existence of a unique fixed-point $\epsilon^x_\#(\boldsymbol{d}) \in \mathcal{R}(\boldsymbol{d})$ implies that we can replace constraint (3.32) by $\boldsymbol{c}(\epsilon^x) + \boldsymbol{d} = \epsilon^x$. In the following result, uniqueness of $\epsilon^x_\#(\boldsymbol{d})$ was shown under a slightly more restrictive assumption.

Lemma 3.2. *[144, Theorem 3] Suppose Assumption 3.2 holds and $\boldsymbol{d} > \mathbf{0}$, then there exists a unique fixed-point $\epsilon^x_\#(\boldsymbol{d}) \in \mathcal{R}(\boldsymbol{d})$.* □

We now present a brief discussion regarding the restrictions imposed by the assumption $\boldsymbol{d} > \mathbf{0}$: recalling the definition of the support function

$$\begin{aligned} \boldsymbol{d}_i = \max_{w} &\;\; E_i B w \\ \text{s.t.} &\;\; F w \leq \epsilon^w, \end{aligned}$$

we see that $\boldsymbol{d}_i > 0$ for all $i \in \mathbb{I}_1^{m_X}$ only if $E_i B \neq \mathbf{0}$ for each $i \in \mathbb{I}_1^{m_X}$ and $\epsilon^w > \mathbf{0}$. While the condition $\epsilon^w > \mathbf{0}$ can be enforced easily through a linear constraint in Problems (3.33), the former condition holds only if the additional assumption $E_i^\top \notin \mathrm{null}(B^\top)$ (or the stronger assumption $\mathrm{rank}(B) = n_x$) is satisfied: these assumptions restrict the class of systems and RPI set parametrizations that are often encountered. Moreover, they lead to excessively conservative RPI set parametrizations. For example, an uncontrollable system would require an RPI set that always includes the origin within its interior.

We prove next that there exists a unique fixed-point $\epsilon^x_\#(\boldsymbol{d}) \in \mathcal{R}(\boldsymbol{d})$ if $\boldsymbol{d} \geq \mathbf{0}$ (rather than $\boldsymbol{d} > \mathbf{0}$). To this end, we first characterize the fixed-points using the following LP, similarly to [144]:

$$\max_{\boldsymbol{c}, \mathbf{x} := \{\mathbf{x}^i, i \in \mathbb{I}_1^{m_X}\}} \;\; \sum_{i=1}^{m_X} \boldsymbol{c}_i \tag{3.35a}$$

$$\text{s.t.} \;\; \boldsymbol{c}_i - E_i A \mathbf{x}^i \leq 0, \quad i \in \mathbb{I}_1^{m_X}, \tag{3.35b}$$

$$E \mathbf{x}^i \leq \boldsymbol{c} + \boldsymbol{d}, \quad i \in \mathbb{I}_1^{m_X}, \tag{3.35c}$$

and we denote the set of all optimizers $(\boldsymbol{c}^*, \mathbf{x}^*)$ of LP (3.35) as $\mathcal{S}$.

Proposition 3.3. *Suppose Assumption 3.2 holds. Then if $\bar{\epsilon}^x \in \mathcal{R}(\boldsymbol{d})$ there exists a $(\bar{\boldsymbol{c}}, \bar{\mathbf{x}}) \in \mathcal{S}$ such that $\bar{\boldsymbol{c}}_i = E_i A \bar{\mathbf{x}}^i$ and $\bar{\epsilon}^x = \bar{\boldsymbol{c}} + \boldsymbol{d}$.* □

Proof. If Assumption 3.2 holds, Lemma 3.1.1 entails that $\mathcal{R}(\boldsymbol{d})$ is nonempty for every $\boldsymbol{d} \geq \mathbf{0}$. At every fixed-point solution $\bar{\epsilon}^x \in \mathcal{R}(\boldsymbol{d})$, $\bar{\epsilon}^x = \boldsymbol{c}(\bar{\epsilon}^x) + \boldsymbol{d}$ holds. Define

$$\begin{aligned} \bar{\mathbf{x}}^i := \arg\max_{\mathbf{x}^i} \ & E_i A \mathbf{x}^i \\ \text{s.t.} \ & E \mathbf{x}^i \leq \bar{\epsilon}^x, \end{aligned}$$

and $\bar{\boldsymbol{c}}_i := E_i A \bar{\mathbf{x}}^i$. By definition of $\boldsymbol{c}(\cdot)$ we have $\bar{\epsilon}^x_i = \bar{\boldsymbol{c}}_i + \boldsymbol{d}_i$ for each $i \in \mathbb{I}_1^{m_X}$. We combine the LPs defining $\bar{\mathbf{x}}^i$ into a single LP by defining

$$\bar{\mathbf{x}} := \{\bar{\mathbf{x}}^i, i \in \mathbb{I}_1^{m_X}\},$$

and adopting an epigraph form [20] by introducing variables $\boldsymbol{c}_i$ to obtain

$$\max_{\boldsymbol{c}, \mathbf{x}} \quad \sum_{i=1}^{m_X} \mathbf{c}_i \tag{3.36}$$

$$\text{s.t.} \quad \mathbf{c}_i - E_i A \mathbf{x}^i \leq 0, \qquad i \in \mathbb{I}_1^{m_X} \tag{3.37}$$

$$E \mathbf{x}^i \leq \bar{\boldsymbol{c}} + \boldsymbol{d}, \qquad i \in \mathbb{I}_1^{m_X}, \tag{3.38}$$

in which we write $\bar{\epsilon}^x = \bar{\boldsymbol{c}} + \boldsymbol{d}$.

Since $(\boldsymbol{c}, \mathbf{x}) = (\bar{\boldsymbol{c}}, \bar{\mathbf{x}})$ is feasible for LP (3.36), and the optimal value is $\sum_{i=1}^{m_X} \bar{\boldsymbol{c}}_i$, we can replace $\bar{\mathbf{c}}$ by $\mathbf{c}$ to obtain LP (3.35), and $(\bar{\boldsymbol{c}}, \bar{\mathbf{x}})$ will be one of the optimizers. □

Proposition 3.3 entails that every fixed-point $\bar{\epsilon}^x \in \mathcal{R}(\boldsymbol{d})$ can be expressed as $\bar{\epsilon}^x = \bar{\boldsymbol{c}} + \boldsymbol{d}$ for some $\bar{\boldsymbol{c}} \in \Pi_{\boldsymbol{c}^*} \mathcal{S}$ (Note that, for now, $\Pi_{\boldsymbol{c}^*} \mathcal{S}$ need not be singleton). In Theorem 3.1, we exploit this property to show that the fixed-point is unique. To this end, we first present the following general result that we use later to establish uniqueness.

Lemma 3.3. *Let $M \in \mathbb{R}^{p \times p}$ be a matrix with $M_{ij} \geq 0, \forall\, i, j \in \mathbb{I}_1^p$, and*

$$N := M(\mathbf{I} + \operatorname{diag}(M\mathbf{1}))^{-1} \quad \text{, i.e.,} \quad N_{ij} = \frac{M_{ij}}{1 + \sum_{k=1}^p M_{jk}}.$$

Then, it holds that

1. *$Z := \mathbf{I} - N^\top$ is invertible;*

2. $\rho(N^\top) < 1$.

$\square$

Proof.

1. Matrix Z is invertible if and only if $Z^\top$ is invertible. Suppose there exists some $q \in \mathbb{R}^p$ satisfying

$$Nq + \mathbf{1} \leq q, \qquad\qquad q \geq \mathbf{0}, \tag{3.39}$$

such that $Nq < q$ holds. Then, it follows that

$$(\mathbf{I} - N)q > \mathbf{0}, \qquad \text{and} \qquad q \geq \mathbf{0},$$

which, by [45, Theorems 4.1, 4.6], implies that $\mathbf{I} - N$ is invertible. This is because Z is a $\mathcal{Z}$-matrix [45, Definition 1], since the fact that $N_{ii} \leq 1$ implies that the diagonal elements $Z_{ii} \geq 0$ for all $i \in \mathbb{I}_1^p$, and all the off-diagonal elements $Z_{ij} \leq 0, \forall\, i, j \in \mathbb{I}_1^p$.

We show next that indeed there exists some $q \in \mathbb{R}^p$ satisfying (3.39). To this end, we introduce a slack variable $s \in \mathbb{R}^p$ in the aforementioned formulation, and write (3.39) equivalently as

$$\begin{bmatrix} N - \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix} = -\mathbf{1}, \qquad\qquad \begin{bmatrix} q \\ s \end{bmatrix} \geq \mathbf{0}. \tag{3.40}$$

By Farkas' lemma [132, Corollary 7.1d], there exist $[q^\top \; s^\top]^\top$ satisfying (3.40) if and only if

$$\zeta^\top \mathbf{1} \leq \mathbf{0}, \qquad \forall\, \zeta \in \mathcal{T} := \{\zeta \;:\; N^\top \zeta \geq \zeta,\; \zeta \geq \mathbf{0}\}.$$

Since $\zeta \geq \mathbf{0}$ for every $\zeta \in \mathcal{T}$, $\zeta^\top \mathbf{1} \leq \mathbf{0}$ holds if and only if the only ζ that satisfies $N^\top \zeta \geq \zeta$ is $\zeta = \mathbf{0}$, i.e., $\mathcal{T} = \{\mathbf{0}\}$.

To show $\mathcal{T} = \{\mathbf{0}\}$, we rewrite $N^\top \zeta \geq \zeta$ as

$$(\mathbf{I} + \mathrm{diag}(M\mathbf{1}))^{-\top} M^\top \zeta \geq \zeta$$

(using the definition of N), and multiply both sides by the positive diagonal matrix $(\mathbf{I} + \mathrm{diag}(M\mathbf{1}))$ to obtain

$$M^\top \zeta \geq \zeta + \mathrm{diag}(M\mathbf{1})\zeta \Leftrightarrow \begin{cases} \sum_{i=1}^p M_{i1}\zeta_i \geq \zeta_1 + \sum_{k=1}^p M_{1k}\zeta_1, \\ \sum_{i=1}^p M_{i2}\zeta_i \geq \zeta_2 + \sum_{k=1}^p M_{2k}\zeta_2, \\ \vdots \\ \sum_{i=1}^p M_{ip}\zeta_i \geq \zeta_p + \sum_{k=1}^p M_{pk}\zeta_p. \end{cases}$$

We further manipulate these inequalities as

$$\sum_{i=1}^{p} M_{i1}\zeta_i \geq \zeta_1 + M_{11}\zeta_1 + M_{12}\zeta_1 + \cdots + M_{1p}\zeta_1 \quad \to \text{Row } 1$$

$$M_{12}\zeta_1 \geq \zeta_2 + \sum_{k=1}^{p} M_{2k}\zeta_2 - \sum_{i=2}^{p} M_{i2}\zeta_i \quad \to \text{Row } 2$$

$$\vdots$$

$$M_{1p}\zeta_1 \geq \zeta_p + \sum_{k=1}^{p} M_{pk}\zeta_p - \sum_{i=2}^{p} M_{ip}\zeta_i \quad \to \text{Row } p$$

Substituting Rows 2-p in Row 1 to replace $M_{1i}\zeta_1$ terms, we obtain

$$\sum_{i=1}^{p} M_{i1}\zeta_i \geq \sum_{l=1}^{p} \zeta_l + \sum_{j=1}^{p} M_{j1}\zeta_j + \sum_{j=2}^{p}\sum_{k=2}^{p} M_{jk}\zeta_j - \sum_{j=2}^{p}\sum_{i=2}^{p} M_{ij}\zeta_i,$$

which, after elementary operations, yields

$$\sum_{l=1}^{p} \zeta_l \leq 0.$$

Hence, the set

$$\mathcal{T} = \left\{ \zeta : \sum_{l=1}^{p} \zeta_l \leq 0, \ \zeta \geq \mathbf{0} \right\} = \{\mathbf{0}\},$$

such that

$$\zeta^\top \mathbf{1} \leq \mathbf{0}, \qquad \forall\, \zeta \in \mathcal{T}.$$

Thus, there exists some $q \in \mathbb{R}^p$ satisfying (3.39), concluding the proof of the first claim.

2. Since

$$(\mathbf{I} - N^\top)^{-1} = \sum_{k=0}^{\infty} (N^\top)^k$$

is well-defined, it implies $\lim_{k\to\infty}(N^\top)^k = \mathbf{0}$, or, equivalently, that the spectral radius $\rho(N^\top) < 1$.

$\square$

Theorem 3.1. *Suppose that Assumption 3.2 holds and $\boldsymbol{d} \geq \mathbf{0}$, then there exists a unique fixed-point $\epsilon^x_\#(\boldsymbol{d}) \in \mathcal{R}(\boldsymbol{d})$.* □

Proof. By Assumption 3.2, Lemma 3.1 entails $\mathcal{R}(\boldsymbol{d}) \neq \emptyset$, and the fixed-point $\bar{\epsilon}^x = \epsilon^x_*(\mathbf{0}, \boldsymbol{d})$ reached from $\epsilon^x_{[0]} = \mathbf{0}$ with the iterations $\epsilon^x_{[k+1]} = \boldsymbol{c}(\epsilon^x_{[k]}) + \boldsymbol{d}$ is the minimal fixed-point, i.e.,

$$\bar{\epsilon}^x \leq \tilde{\epsilon}^x, \qquad \forall\, \tilde{\epsilon}^x \in \mathcal{R}(\boldsymbol{d}). \tag{3.41}$$

In order to show uniqueness of this fixed-point, we show that the iterations $\epsilon^x_{[k+1]} = \boldsymbol{c}(\epsilon^x_{[k]}) + \boldsymbol{d}$ starting from any initial-condition $\epsilon^x_{[0]} \geq \bar{\epsilon}^x$ converge to $\bar{\epsilon}^x$. Since this initial condition can be any other fixed-point $\tilde{\epsilon}^x \in \mathcal{R}(\boldsymbol{d}) \setminus \{\bar{\epsilon}^x\}$, we will conclude the proof by noting that iterations with $\epsilon^x_{[0]} = \tilde{\epsilon}^x$ converging to $\bar{\epsilon}^x$ implies $\tilde{\epsilon}^x = \bar{\epsilon}^x$.

To this end, we observe that Proposition 3.3 entails that there exists some optimizer $\bar{\boldsymbol{c}} \in \Pi_{\boldsymbol{c}^*}\mathcal{S}$ of LP (3.35) such that $\bar{\epsilon}^x = \bar{\boldsymbol{c}} + \boldsymbol{d}$. Then, we write the dual LP of LP (3.35) as

$$\min_{(\boldsymbol{\lambda},\boldsymbol{\eta}):=\{\lambda_i,\eta^i,i\in\mathbb{I}_1^{m_X}\}} \sum_{i=1}^{m_X} \eta^{i^\top} \boldsymbol{d} \tag{3.42a}$$

$$\text{s.t.} \quad \lambda_i = 1 + \sum_{j=1}^{m_X} \eta_i^j, \qquad i \in \mathbb{I}_1^{m_X}, \tag{3.42b}$$

$$\eta^{i^\top} E = \lambda_i E_i A, \qquad i \in \mathbb{I}_1^{m_X}, \tag{3.42c}$$

$$\lambda_i \geq 0, \eta^i \geq \mathbf{0}_{m_X}, \qquad i \in \mathbb{I}_1^{m_X} \tag{3.42d}$$

where λ_i and η^i are the dual variables associated to constraints (3.35b) and (3.35c) respectively. We denote the optimal dual variables corresponding to $\bar{\boldsymbol{c}}$ as λ_i^* and η^{i*}, and define matrix Θ^* with rows

$$\Theta_i^* := \frac{\eta^{i*^\top}}{\lambda_i^*},$$

where $\lambda_i^* \geq 1$ by (3.42b). We recall that

$$\bar{\boldsymbol{c}} = \Theta^*(\bar{\boldsymbol{c}} + \boldsymbol{d}) = \Theta^* \bar{\epsilon}^x,$$

since $\bar{\boldsymbol{c}}$ optimizes LP (3.35) ([144, Theorem 4]).

Then we apply Lemma 3.3 with

$$M = [\eta^{1*} \ \cdots \ \eta^{m_X*}],$$

such that $N = \Theta^*$. Hence, $\rho(\Theta^*) < 1$ from Lemma 3.3(b).

For any $\epsilon^x \in \mathcal{H}(\boldsymbol{d})$, it follows that

$$\boldsymbol{c}_i(\epsilon^x) = \left\{ \begin{array}{ll} \max\limits_{x} & E_i A x \\ \text{s.t.} & E x \leq \epsilon^x \end{array} \right\} = \left\{ \begin{array}{ll} \min\limits_{\gamma \geq \mathbf{0}} & \gamma^\top \epsilon^x \\ \text{s.t.} & \gamma^\top E = E_i A, \end{array} \right\} \leq \Theta_i^* \epsilon^x,$$

where the second equality follows from strong duality for LPs, and the inequality follows since $\gamma^\top = \Theta_i^*$ is feasible for the dual LP. This implies

$$\boldsymbol{c}(\epsilon^x) \leq \Theta^* \epsilon^x, \qquad \forall \quad \epsilon^x \in \mathcal{H}(\boldsymbol{d}).$$

Hence, for the iterations $\epsilon^x_{[k+1]} = \boldsymbol{c}(\epsilon^x_{[k]}) + \boldsymbol{d}$ from any $\epsilon^x_{[0]} \in \mathcal{H}(\boldsymbol{d})$, we obtain

$$\epsilon^x_{[k+1]} \leq \Theta^* \epsilon^x_{[k]} + \boldsymbol{d}.$$

Subtracting by $\bar{\epsilon}^x = \Theta^* \bar{\epsilon}^x + \boldsymbol{d}$, the inequality

$$\epsilon^x_{[k+1]} - \bar{\epsilon}^x \leq \Theta^* (\epsilon^x_{[k]} - \bar{\epsilon}^x)$$

follows. Applying recursively, the inequality

$$\epsilon^x_{[k]} - \bar{\epsilon}^x \leq (\Theta^*)^k (\epsilon^x_{[0]} - \bar{\epsilon}^x)$$

holds. If $\bar{\epsilon}^x \leq \epsilon^x_{[0]}$, then $\bar{\epsilon}^x \leq \epsilon^x_{[k]}$ for all $k \geq 0$ by monotonicity of $\boldsymbol{c}(\cdot)$, and definition of $\bar{\epsilon}^x$. Then, $\rho(\Theta^*) < 1$ implies $(\Theta^*)^k \to \mathbf{0}$ as $k \to \infty$, such that

$$\forall\, \delta > 0,\ \exists\, k < \infty \ : \ \epsilon^x_{[k]} - \bar{\epsilon} \leq \delta \mathbf{1}. \tag{3.43}$$

If the initial condition

$$\epsilon^x_{[0]} = \tilde{\epsilon}^x \in \mathcal{R}(\boldsymbol{d}) \setminus \{\bar{\epsilon}^x\},$$

i.e., the iterations start at some fixed-point that is not the minimal fixed-point $\bar{\epsilon}^x$, then $\epsilon^x_{[k]} = \tilde{\epsilon}^x$ for all $k \geq 1$, since $\tilde{\epsilon}^x$ is a fixed-point.

From (3.43), this implies $\tilde{\epsilon}^x \leq \bar{\epsilon}^x + \delta \mathbf{1}$ for every $\delta > 0$. From (3.41), we know that $\bar{\epsilon}^x \leq \tilde{\epsilon}^x$. Suppose there exist some index $i \in \mathbb{I}_1^{m_X}$ such that $\bar{\epsilon}^x_i < \tilde{\epsilon}^x_i$. Then, for every arbitrary scalar $\beta \in (0, \tilde{\epsilon}^x_i - \bar{\epsilon}^x_i)$, $\tilde{\epsilon}^x \not\leq \bar{\epsilon}^x + \beta \mathbf{1}$ holds, which contradicts (3.43) with $\epsilon^x_{[k]} = \tilde{\epsilon}^x$. Hence,

$$\epsilon^x_{\#}(\boldsymbol{d}) = \bar{\epsilon}^x = \tilde{\epsilon}^x,$$

which concludes the proof. □

Remark 3.3. *We note that $\rho(\Theta^*) \in [\rho(A), 1)$: Let (α, κ_α) be an eigenpair of A, such that $A\kappa_\alpha = \alpha\kappa_\alpha$. Multiplying by E, we obtain*

$$\Theta^*(E\kappa_\alpha) = \alpha(E\kappa_\alpha)$$

since $\Theta^* E = EA$ from (3.42b)-(3.42c). *Hence, the eigenvalues of A are a subset of the eigenvalues of Θ^*.* □

This theorem validates (3.34) and allows us to replace constraint (3.32) by the equivalent functional equality $\boldsymbol{c}(\epsilon^x) + \boldsymbol{d} = \epsilon^x$ in Problem (3.33)

Remark 3.4. *While we assume that $\mathbf{0} \in \mathbb{W}(\epsilon^w)$, there exist cases where this is not known a priori. Such cases can be accommodated in Problems* (3.33) *by considering the disturbance set parametrization $\{\bar{w}\} \oplus \mathbb{W}(\epsilon^w)$, where $\mathbf{0} \in \mathbb{W}(\epsilon^w)$ if $\epsilon^w \geq \mathbf{0}$, and $\bar{w}$ represents the origin offset. Then, an RPI set parametrized as $\{\bar{x}\} \oplus \mathbb{X}(\epsilon^x)$ satisfies*

$$\{A\bar{x} + B\bar{w}\} \oplus A\mathbb{X}(\epsilon^x) \oplus B\mathbb{W}(\epsilon^w) \subseteq \{\bar{x}\} \oplus \mathbb{X}(\epsilon^x),$$

or equivalently $EA\bar{x} + EB\bar{w} - E\bar{x} + \boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) \leq \boldsymbol{b}(\epsilon^x)$, the first part of which can be eliminated by using the state offset $\bar{x} = (\mathbf{I} - A)^{-1}B\bar{w}$. □

3.3.5 Implementation of Problem (3.33)

Using the result in Theorem 3.1 and recalling the definition of the distance function $\mathrm{d}_{\mathcal{Y}}(l, \mathbb{W}(\epsilon^w))$ from (3.8), we write Problem (3.33) equivalently as

$$\begin{aligned}
\min_{\epsilon^x, \epsilon^w, \epsilon} \quad & \|\epsilon\|_1 && (3.44a)\\
\text{s.t.} \quad & \boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) = \epsilon^x, && (3.44b)\\
& \boldsymbol{l}(\epsilon^x) + \boldsymbol{m}(\epsilon^w) \leq g, && (3.44c)\\
& \mathcal{Y} \subseteq \bigoplus_{t=0}^{l-1} CA^t B\mathbb{W}(\epsilon^w) \oplus D\mathbb{W}(\epsilon^w) \oplus \mathbb{B}(\epsilon), && (3.44d)\\
& \epsilon^w \geq \mathbf{0}. && (3.44e)
\end{aligned}$$

In order to implement Constraint (3.44d), we assume to know the vertices of the output constraint set $\mathcal{Y}$.

Assumption 3.4. *The vertices $\{\mathrm{y}_{[p]}, p \in \mathbb{I}_1^{v_{\mathcal{Y}}}\} = \mathrm{vert}(\mathcal{Y})$ are known a priori.* □

Then, inclusion (3.44d) verifies if and only if $(\epsilon^w, \epsilon) \in \Xi$, where

$$\Xi := \left\{ (\epsilon^w, \epsilon) : \begin{array}{c} \forall\, p \in \mathbb{I}_1^{v\mathcal{Y}}, \exists \{\mathrm{w}_{[pt]},\ t \in \mathbb{I}_0^l\} \in \mathbb{W}(\epsilon^w), \mathrm{b}_{[p]} \in \mathbb{B}(\epsilon) : \\ \mathrm{y}_{[p]} = \sum_{t=0}^{l-1} CA^{l-1-t}B\mathrm{w}_{[pt]} + D\mathrm{w}_{[pl]} + \mathrm{b}_{[p]} \end{array} \right\},$$

Defining the variables $z := \{\mathrm{w}_{[pt]}, \mathrm{b}_{[p]},\ p \in \mathbb{I}_1^{v\mathcal{Y}},\ t \in \mathbb{I}_0^l\}$, Problem (3.44) can hence be implemented as

$$\begin{aligned} \min_{\epsilon^x, \epsilon^w, \epsilon, z} \quad & \|\epsilon\|_1 && (3.45a) \\ \text{s.t.} \quad & \boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) = \epsilon^x, && (3.45b) \\ & \boldsymbol{l}(\epsilon^x) + \boldsymbol{m}(\epsilon^w) \leq g, && (3.45c) \\ & (\epsilon^w, \epsilon) \in \Xi, && (3.45d) \\ & \epsilon^w \geq \mathbf{0}, && (3.45e) \end{aligned}$$

Remark 3.5. *If the vertices $\{\mathrm{y}_{[p]}, p \in \mathbb{I}_1^{v\mathcal{Y}}\}$ of the output constraint set $\mathcal{Y}$ are not known, then Constraint* (3.44d) *can be encoded as a set of linear constraints directly in terms of the hyperplane notation of $\mathcal{Y}$ using the sufficient conditions for polytopic inclusions presented in [127, Theorem 1].* □

3.4 Numerical Optimization

In this section, we develop a numerical optimization algorithm in order to solve Problem (3.45).

3.4.1 Related literature and Problem setup

The central difficulty in solving Problem (3.45) arises from the fact that the constraints are nonsmooth and nonconvex, since they are defined using support functions over polytopes. A typical approach to tackle such problems is to resort to the Karush-Kuhn-Tucker (KKT) optimality conditions [102]: Since the support functions are defined using Linear Programs (LPs), they can be replaced by their corresponding optimality conditions. The resulting lifted reformulation [7] is a Mathematical Program with Complementarity Conditions (MPCC) [61] that can be tackled by mixed-integer programming [161]. However, this approach exhibits an exponential increase in the number of binary variables with the system dimension. An alternative is to use smoothening-based approaches, in which the nonsmooth complementarity KKT condition is replaced by a smooth approximation [109, 77], that can then be solved with an off-the-shelf nonlinear programming solver (NLP) like IPOPT [151]. Smooth formulations can also be derived using approaches based on zeroing the duality-gap [4, 154]. Such approaches were previously considered for invariant-set design in [68]. These reformulations too, however, suffer from the curse of dimensionality.

In this section, we present an approach based on implicit functions [7], in which the support functions formulating the optimization problem are treated as implicit functions of the parameters of the disturbance and RPI sets, and sensitivities of these functions are calculated using parametric optimization theory [23]. We introduce a smoothening-based approach

Nonunique representations of the set $\mathbb{W}(\epsilon^w)$

We note that, in general, there exist infinitely many values of ϵ^w characterizing a given set $\mathbb{W}(\epsilon^w)$ because of redundant hyperplanes, i.e., there can exist $\epsilon^{w,1}, \epsilon^{w,2}$ such that

$$\epsilon^{w,1} \neq \epsilon^{w,2}, \qquad \text{with} \qquad \mathbb{W}(\epsilon^{w,1}) = \mathbb{W}(\epsilon^{w,2}).$$

This nonuniqueness can negatively affect our optimization procedure. In order to tackle it, we define the support functions

$$\boldsymbol{q}_t(\epsilon^w) := h_{\mathbb{W}(\epsilon^w)}(F_t^\top), \qquad \forall\, t \in \mathbb{I}_1^{m_W}.$$

We then note that

$$\mathbb{W}(\epsilon^{w,1}) = \mathbb{W}(\epsilon^{w,2}) \iff \boldsymbol{q}(\epsilon^{w,1}) = \boldsymbol{q}(\epsilon^{w,2}),$$

and there exists a unique ϵ^w such that $\mathbb{W}(\epsilon^w) = \mathbb{W}(\epsilon^{w,1}) = \mathbb{W}(\epsilon^{w,2})$ and $\epsilon^w = \boldsymbol{q}(\epsilon^w)$. In $\mathbb{W}(\epsilon^w)$, the redundant hyperplanes are tangent to the set. Hence, we encourage the computation of such an unique ϵ^w by appending $\frac{1}{2}\sigma \left\|\epsilon^w - \boldsymbol{q}(\epsilon^w)\right\|_2^2$ to the objective function in Problem (3.45), where

$\sigma > 0$ is some user-defined scalar, resulting in

$$
\begin{aligned}
\min_{\mathrm{z}} \quad & \|\epsilon\|_1 + \frac{1}{2}\sigma \left\|\epsilon^w - \boldsymbol{q}(\epsilon^w)\right\|_2^2 && \text{(3.46a)}\\
\text{s.t.} \quad & \boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) = \epsilon^x, && \text{(3.46b)}\\
& A^{\mathrm{E}}\mathrm{z} = b^{\mathrm{E}}, && \text{(3.46c)}\\
& \boldsymbol{l}(\epsilon^x) + \boldsymbol{m}(\epsilon^w) \le g, && \text{(3.46d)}\\
& A^{\mathrm{I}}\mathrm{z} \le b^{\mathrm{I}}, && \text{(3.46e)}
\end{aligned}
$$

where $\mathrm{z} := \{\epsilon^w, \epsilon^x, \epsilon, z\}$ is the optimization vector, and the linear constraint set

$$\{\mathrm{z} : A^{\mathrm{E}}\mathrm{z} = b^{\mathrm{E}}, A^{\mathrm{I}}\mathrm{z} \le b^{\mathrm{I}}\}$$

captures the constraints $\epsilon^w \ge \mathbf{0}$ and $(\epsilon^x, \epsilon^w, \epsilon) \in \Xi$. While any $\sigma > 0$ is suitable, we observed that large values can slow down the convergence rate of the NLP approach we propose in the sequel to solve Problem (3.46).

The rest of this section is devoted to the development of an optimization algorithm to solve Problem (3.46). Firstly, we introduce the proposed smoothening procedure and discuss the feasibility of the resulting smoothened problem. The smoothened approximation is then used later to develop a PDIP solver.

3.4.2 Smooth approximation

In order to solve Problem (3.46), we note that given (ϵ^x, ϵ^w), the support functions $\boldsymbol{c}_i(\epsilon^x), \boldsymbol{d}_i(\epsilon^w), \boldsymbol{l}_k(\epsilon^x), \boldsymbol{m}_k(\epsilon^w), \boldsymbol{q}_t(\epsilon^w)$ can be evaluated by solving the corresponding linear programs. In order to clarify this point and aid further developments, we introduce the following parametric LP with parameter $\boldsymbol{\epsilon} \in \mathbb{R}^{\boldsymbol{m}}$:

$$\max_{\mathbf{z} \in \mathbb{R}^{\boldsymbol{n}}} \ \mathbf{r}^\top \mathbf{z} \quad \text{s.t.} \quad \mathbf{P}\mathbf{z} \le \boldsymbol{\epsilon}. \tag{3.47}$$

We label the primal-dual solution pair of Problem (3.47) as $(\{\mathbf{z}^*\}, \{\boldsymbol{\lambda}^*\})$ (we omit the dependence of $\{\mathbf{z}^*\}, \{\boldsymbol{\lambda}^*\}$ on $\boldsymbol{\epsilon}$ for simplicity), and the value function as

$$\boldsymbol{v}(\boldsymbol{\epsilon}) := \mathbf{r}^\top \mathbf{z}^*.$$

$\boldsymbol{v}(\boldsymbol{\epsilon})$	$\mathbf{r}^\top$	$\mathbf{P}$	$\boldsymbol{\epsilon}$	$\mathbf{z}^*$	$\boldsymbol{\lambda}^*$	$\mathbf{s}^*$	$\boldsymbol{n}$	$\boldsymbol{m}$
$\boldsymbol{c}_i(\epsilon^x)$	E_iA	E	ϵ^x	$\mathrm{z}^{\boldsymbol{c}_i}$	$\lambda^{\boldsymbol{c}_i}$	$s^{\boldsymbol{c}_i}$	n_x	m_X
$\boldsymbol{d}_i(\epsilon^w)$	E_iB	F	ϵ^w	$\mathrm{z}^{\boldsymbol{d}_i}$	$\lambda^{\boldsymbol{d}_i}$	$s^{\boldsymbol{d}_i}$	n_w	m_W
$\boldsymbol{l}_k(\epsilon^x)$	G_kC	E	ϵ^x	$\mathrm{z}^{\boldsymbol{l}_k}$	$\lambda^{\boldsymbol{l}_k}$	$s^{\boldsymbol{l}_k}$	n_x	m_X
$\boldsymbol{m}_k(\epsilon^w)$	G_kD	F	ϵ^w	$\mathrm{z}^{\boldsymbol{m}_k}$	$\lambda^{\boldsymbol{m}_k}$	$s^{\boldsymbol{m}_k}$	n_w	m_W
$\boldsymbol{q}_t(\epsilon^w)$	F_t	F	ϵ^w	$\mathrm{z}^{\boldsymbol{q}_t}$	$\lambda^{\boldsymbol{q}_t}$	$s^{\boldsymbol{q}_t}$	n_w	m_W

Table 1: Support functions

We use $\boldsymbol{v}(\boldsymbol{\epsilon})$ to represent any of the support functions involved in the formulation of Problem (3.46), as described in Table 1. In other words, at a given (ϵ^x, ϵ^w), the support functions are defined as

$$\begin{aligned}
\boldsymbol{c}_i(\epsilon^x) &:= E_iA\mathrm{z}^{\boldsymbol{c}_i},\\
\boldsymbol{d}_i(\epsilon^w) &:= E_iB\mathrm{z}^{\boldsymbol{d}_i},\\
\boldsymbol{l}_k(\epsilon^x) &:= G_kC\mathrm{z}^{\boldsymbol{l}_k},\\
\boldsymbol{m}_k(\epsilon^w) &:= G_kD\mathrm{z}^{\boldsymbol{m}_k},\\
\boldsymbol{q}_t(\epsilon^w) &:= F_t\mathrm{z}^{\boldsymbol{q}_t}.
\end{aligned}$$

Problem (3.47) can be equivalently written using slack variables as

$$\max_{\mathbf{z}\in\mathbb{R}^{\boldsymbol{n}},\mathbf{s}\in\mathbb{R}^{\boldsymbol{m}}} \ \mathbf{r}^\top\mathbf{z} \quad \text{s.t.} \quad \mathbf{P}\mathbf{z}+\mathbf{s}=\boldsymbol{\epsilon},\ \mathbf{s}\geq\mathbf{0}, \tag{3.48}$$

with the primal-dual solution pair $(\{\mathbf{z}^*, \mathbf{s}^*\}, \{\boldsymbol{\lambda}^*\})$. This solution satisfies the KKT conditions of Problem (3.48), i.e.,

$$\mathbf{P}^\top\boldsymbol{\lambda}^* = \mathbf{r}^*, \qquad \mathbf{P}\mathbf{z}^*+\mathbf{s}^*=\boldsymbol{\epsilon}, \qquad \boldsymbol{\lambda}^*\circ\mathbf{s}^*=\mathbf{0}, \tag{3.49}$$

along with $\boldsymbol{\lambda}^*, \mathbf{s}^* \geq \mathbf{0}$.

A popular approach to solve Problem (3.46) involves a KKT-based reformulation. This reformulation involves introducing the optimization variables $(\mathbf{z}^*, \boldsymbol{\lambda}^*, \mathbf{s}^*)$ into Problem (3.46), replacing $\boldsymbol{v}(\boldsymbol{\epsilon})$ with $\mathbf{r}^\top\mathbf{z}^*$, and appending the KKT conditions (3.49) as constraints on Problem (3.46). However, by this approach the problem dimension increases quadratically with the number of hyperplanes defining the sets $\mathbb{X}(\epsilon^x)$, $\mathbb{W}(\epsilon^w)$, and $\mathcal{Y}$ through the variables $\boldsymbol{\lambda}^*$ and $\mathbf{s}^*$. For example, the constraint

$\boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) = \epsilon^x$ requires $m_X^2 + m_X m_W$ additional optimization variables $\{\lambda^{\boldsymbol{c}_i}, \lambda^{\boldsymbol{d}_i}, i \in \mathbb{I}_1^{m_X}\}$, thus increasing the problem complexity. Hence, we resort to using an implicit-function approach, in which the support functions $\boldsymbol{v}(\boldsymbol{\epsilon})$ are treated as implicit functions of $\boldsymbol{\epsilon}$. It is well-known that these functions are in general nonsmooth in their parameter $\boldsymbol{\epsilon}$ [7]. This implies that standard derivative-based NLP solvers cannot be applied to solve Problem (3.46). In the sequel, we present a suitable smoothening-based approximation of $\boldsymbol{v}(\boldsymbol{\epsilon})$, and a corresponding smoothened approximate formulation of Problem (3.46) that can then be solved by introducing a minor modification by any standard NLP solver.

Regularization and smooth formulation

We now focus on computing the first and second order sensitivities of $\boldsymbol{v}(\boldsymbol{\epsilon})$

$$\nabla_{\boldsymbol{\epsilon}} \boldsymbol{v}(\boldsymbol{\epsilon}) = \mathbf{r}^\top \frac{\partial \mathbf{z}^*}{\partial \boldsymbol{\epsilon}}, \qquad \nabla^2_{\boldsymbol{\epsilon}\boldsymbol{\epsilon}} \boldsymbol{v}(\boldsymbol{\epsilon}) = \sum_{l=1}^{\boldsymbol{n}} \mathbf{r}_l \frac{\partial^2 \mathbf{z}_l^*}{\partial \boldsymbol{\epsilon}^2}, \tag{3.50}$$

for a given value of $\boldsymbol{\epsilon} \geq \mathbf{0}$. We know from [38, Theorem 3.2.2] that if the linear independence constraint qualification (LICQ) condition, second-order sufficient conditions (SOSC), and strict complementary slackness (SCS) are satisfied at the solution of the LP (3.48)[1], then the solution $(\mathbf{z}^*, \boldsymbol{\lambda}^*, \mathrm{s}^*)$ is a continuously differentiable function of $\boldsymbol{\epsilon}$ in a neighborhood of $(\mathbf{z}^*, \boldsymbol{\lambda}^*, \mathrm{s}^*)$. However, for an arbitrary $\boldsymbol{\epsilon} \geq \mathbf{0}$ these conditions might be violated, and hence we propose the following approximation. Following [128], let us first regularize the LP (3.47) using primal and dual regularization parameters $\kappa := (\kappa_\mathrm{p}, \kappa_\mathrm{d}) > 0$ as

$$\begin{aligned} \max_{\mathbf{z},\theta} \quad & \mathbf{r}^\top \mathbf{z} - 0.5(\|\kappa_\mathrm{p} \mathbf{z}\|_2^2 + \|\theta\|_2^2) \\ \text{s.t.} \quad & \mathbf{P}\mathbf{z} + \kappa_\mathrm{d}\theta \leq \boldsymbol{\epsilon}. \end{aligned} \tag{3.51}$$

We label the primal-dual solution of Problem (3.51) as $(\{\mathbf{z}^\kappa, \theta^\kappa\}, \{\boldsymbol{\lambda}^\kappa\})$, and denote

$$\boldsymbol{v}^\kappa(\boldsymbol{\epsilon}) := \mathbf{r}^\top \mathbf{z}^\kappa.$$

[1]For precise definitions of LICQ, SOSC and SCS, we refer to [102].

The ℓ_2-regularization term in (3.51) ensures that the feasible set of the Quadratic Program (QP) (3.51) contains a nonempty interior and satisfies LICQ and SOSC at the unique solution $(\{\mathbf{z}^{\kappa}, \theta^{\kappa}\}, \{\boldsymbol{\lambda}^{\kappa}\})$.

Proposition 3.4. *$\boldsymbol{v}^{\kappa}(\boldsymbol{\epsilon}) \to \boldsymbol{v}(\boldsymbol{\epsilon})$ quadratically as $\kappa \to \mathbf{0}$.* □

Proof.

- **(Part 1)** For some $\kappa_{\mathrm{d}} > 0$, consider the QP

$$\begin{aligned} \max_{\mathbf{z},\theta} \quad & \mathbf{r}^{\top}\mathbf{z} - 0.5\,\|\theta\|_2^2 \\ \text{s.t.} \quad & \mathbf{P}\mathbf{z} + \kappa_{\mathrm{d}}\theta \leq \boldsymbol{\epsilon} \end{aligned} \tag{3.52}$$

  and let $(\{\mathbf{z}^{\kappa_{\mathrm{d}}}, \theta^{\kappa_{\mathrm{d}}}\}, \{\boldsymbol{\lambda}^{\kappa_{\mathrm{d}}}\})$ be the corresponding primal-dual solution. Problem (3.51) is a perturbed version of QP (3.52), with perturbation parameter κ_{p} over variables $\mathbf{z}$. Then, since Problem (3.52) satisfies LICQ as the matrix $[\mathbf{P}\ \kappa_{\mathrm{d}}\mathbf{I}]$ is full rank for all $\kappa_{\mathrm{d}} > 0$, we know from [84, Theorem 2] that there exists some $\bar{\kappa}_{\mathrm{p}}$ such that if $\kappa_{\mathrm{p}} \in [0, \bar{\kappa}_{\mathrm{p}}]$, the unique primal solution of Problem (3.51) is also a primal solution of Problem (3.52), i.e., $(\mathbf{z}^{\kappa_{\mathrm{d}}}, \theta^{\kappa_{\mathrm{d}}}) = (\mathbf{z}^{\kappa}, \theta^{\kappa})$. Hence,

$$\boldsymbol{v}^{\kappa}(\boldsymbol{\epsilon}) = \boldsymbol{v}^{\kappa_{\mathrm{d}}}(\boldsymbol{\epsilon}) := \mathbf{r}^{\top}\mathbf{z}^{\kappa_{\mathrm{d}}}, \qquad \forall\, \kappa_{\mathrm{p}} \in [0, \bar{\kappa}_{\mathrm{p}}]. \tag{3.53}$$

- **(Part 2)** Let us show now that as $\kappa_{\mathrm{d}} \to 0$, $\boldsymbol{v}^{\kappa_{\mathrm{d}}}(\boldsymbol{\epsilon}) \to \boldsymbol{v}(\boldsymbol{\epsilon})$ quadratically. The dual problem of (3.47) can be rewritten as

$$\min_{\boldsymbol{\lambda}} \quad \boldsymbol{\epsilon}^{\top}\boldsymbol{\lambda} \tag{3.54a}$$
$$\text{s.t.} \quad \mathbf{P}^{\top}\boldsymbol{\lambda} = \mathbf{r}, \tag{3.54b}$$
$$\boldsymbol{\lambda} \geq \mathbf{0}. \tag{3.54c}$$

  Let $(\{\boldsymbol{\lambda}^*\}, \{\alpha^*, \zeta^*\})$ denote the corresponding primal-dual solution, where α^* and ζ^* are the optimal dual variables corresponding to constraints (3.54b) and (3.54c) respectively. We also write the dual problem corresponding to Problem (3.52) as

$$\min_{\boldsymbol{\lambda}} \quad \boldsymbol{\epsilon}^{\top}\boldsymbol{\lambda} + 0.5\,\|\kappa_{\mathrm{d}}\boldsymbol{\lambda}\|_2^2 \tag{3.55a}$$
$$\text{s.t.} \quad \mathbf{P}^{\top}\boldsymbol{\lambda} = \mathbf{r}, \tag{3.55b}$$
$$\boldsymbol{\lambda} \geq \mathbf{0}, \tag{3.55c}$$

with corresponding primal-dual solution $(\{\boldsymbol{\lambda}^{\kappa_\mathrm{d}}\}, \{\alpha^{\kappa_\mathrm{d}}, \zeta^{\kappa_\mathrm{d}}\})$, where $\alpha^{\kappa_\mathrm{d}}$ and $\zeta^{\kappa_\mathrm{d}}$ are the optimal dual variables corresponding to constraints (3.55b) and (3.55c) respectively.

Problem (3.55) is a perturbed version of the LP (3.54), with perturbation parameter κ_d. Then, from [84, Theorem 1] we know that there exists some $\bar{\kappa}_\mathrm{d}$ such that if $\kappa_\mathrm{d} \in [0, \bar{\kappa}_\mathrm{d}]$, the unique optimal primal solution of Problem (3.55) is also an optimal primal solution of Problem (3.54), i.e., $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}^{\kappa_\mathrm{d}}$. Then, since Problems (3.47) and (3.54) have a zero duality-gap due to strong duality, we have $\boldsymbol{v}(\boldsymbol{\epsilon}) = \mathbf{r}^\top \mathbf{z}^* = \boldsymbol{\epsilon}^\top \boldsymbol{\lambda}^*$. However, since $\boldsymbol{\lambda}^{\kappa_\mathrm{d}}$ is an optimal dual solution of Problem (3.47) (equivalently an optimal primal solution of Problem (3.54)) for all $\kappa_\mathrm{d} \in [0, \bar{\kappa}_\mathrm{d}]$, we have

$$\boldsymbol{v}(\boldsymbol{\epsilon}) = \boldsymbol{\epsilon}^\top \boldsymbol{\lambda}^{\kappa_\mathrm{d}}, \qquad \forall\, \kappa_\mathrm{d} \in [0, \bar{\kappa}_\mathrm{d}]. \tag{3.56}$$

- **(Part 3)** Since Problems (3.52) and (3.55) also have a zero duality gap,
$$\boldsymbol{v}^{\kappa_\mathrm{d}}(\boldsymbol{\epsilon}) - 0.5\,\|\theta^{\kappa_\mathrm{d}}\|_2^2 = \boldsymbol{\epsilon}^\top \boldsymbol{\lambda}^{\kappa_\mathrm{d}} + 0.5\,\|\kappa_\mathrm{d} \boldsymbol{\lambda}^{\kappa_\mathrm{d}}\|_2^2$$
holds. From the KKT conditions of Problem (3.52), $\theta^{\kappa_\mathrm{d}} = -\kappa_\mathrm{d} \boldsymbol{\lambda}^{\kappa_\mathrm{d}}$ follows, such that
$$\boldsymbol{v}^{\kappa_\mathrm{d}}(\boldsymbol{\epsilon}) = \boldsymbol{\epsilon}^\top \boldsymbol{\lambda}^{\kappa_\mathrm{d}} + \|\kappa_\mathrm{d} \boldsymbol{\lambda}^{\kappa_\mathrm{d}}\|_2^2\,.$$
Then, it follows from (3.56) that
$$\boldsymbol{v}^{\kappa_\mathrm{d}}(\boldsymbol{\epsilon}) = \boldsymbol{v}(\boldsymbol{\epsilon}) + \|\kappa_\mathrm{d} \boldsymbol{\lambda}^{\kappa_\mathrm{d}}\|_2^2 \geq 0, \qquad \forall\, \kappa_\mathrm{d} \in [0, \bar{\kappa}_\mathrm{d}]. \tag{3.57}$$
Moreover, we have that for every $\kappa_{\mathrm{d},1}$, $\kappa_{\mathrm{d},2}$ such that $0 \leq \kappa_{\mathrm{d},1} < \kappa_{\mathrm{d},2} \leq \bar{\kappa}_\mathrm{d}$,
$$\boldsymbol{v}^{\kappa_{\mathrm{d},1}}(\boldsymbol{\epsilon}) - \boldsymbol{v}^{\kappa_{\mathrm{d},2}}(\boldsymbol{\epsilon}) = (\kappa_{\mathrm{d},1}^2 - \kappa_{\mathrm{d},2}^2)\,\|\boldsymbol{\lambda}^{\kappa_\mathrm{d}}\|_2^2 < 0.$$
Thus, $\boldsymbol{v}^{\kappa_\mathrm{d}}(\boldsymbol{\epsilon})$ is strictly-increasing in $\kappa_\mathrm{d} \in [0, \bar{\kappa}_\mathrm{d}]$, and is lower-bounded by $\boldsymbol{v}(\boldsymbol{\epsilon})$ so that
$$\boldsymbol{v}^{\kappa_\mathrm{d}}(\boldsymbol{\epsilon}) \to \boldsymbol{v}(\boldsymbol{\epsilon}) \text{ as } \kappa_\mathrm{d} \to 0. \tag{3.58}$$
Hence, for $\kappa_\mathrm{p} \in [0, \bar{\kappa}_\mathrm{p}]$ and $\kappa_\mathrm{d} \in [0, \bar{\kappa}_\mathrm{d}]$, $\boldsymbol{v}^{\kappa}(\boldsymbol{\epsilon})$ converges to $\boldsymbol{v}(\boldsymbol{\epsilon})$ quadratically in κ_d from (3.53) and (3.58).

$\square$

While the value of $\boldsymbol{v}^{\kappa}(\boldsymbol{\epsilon})$ of Problem (3.51) converges to the optimal value $\boldsymbol{v}(\boldsymbol{\epsilon})$ of Problem (3.47), SCS might still be violated at the optimizer $\{\mathbf{z}^{\kappa}, \theta^{\kappa}\}$ of Problem (3.51), since there might exist weakly active constraints at the solution. We resolve this issue by eliminating the inequality constraints in Problem (3.51) using a log-barrier formulation with some barrier parameter $\mu > 0$ as

$$\begin{aligned} \max_{\mathbf{z},\theta,\mathbf{s}} \quad & \mathbf{r}^{\top}\mathbf{z} - 0.5(\|\kappa_{\mathrm{p}}\mathbf{z}\|_2^2 + \|\theta\|_2^2) + \mu \sum_{i=1}^{\boldsymbol{m}} \log(\mathbf{s}_i) \\ \text{s.t.} \quad & \mathbf{P}\mathbf{z} + \kappa_{\mathrm{d}}\theta + \mathbf{s} = \boldsymbol{\epsilon}, \end{aligned} \tag{3.59}$$

where $\pi := (\kappa_{\mathrm{p}}, \kappa_{\mathrm{d}}, \mu)$. We label the primal-dual solution pair of Problem (3.59) as $(\{\mathbf{z}^{\pi}, \theta^{\pi}, \mathbf{s}^{\pi}\}, \{\boldsymbol{\lambda}^{\pi}\})$, and define

$$\boldsymbol{v}^{\pi}(\boldsymbol{\epsilon}) := \mathbf{r}^{\top}\mathbf{z}^{\pi}.$$

Since Problem (3.51) satisfies LICQ and SOSC, we known from [89, Proposition 8.2] that

$$(\mathbf{z}^{\pi}, \theta^{\pi}, \boldsymbol{\lambda}^{\pi}) \to (\mathbf{z}^{\kappa}, \theta^{\kappa}, \boldsymbol{\lambda}^{\kappa}) \text{ as } \mu \to 0, \tag{3.60}$$

such that from Theorem 3.4 and (3.60), we have

$$\boldsymbol{v}^{\pi}(\boldsymbol{\epsilon}) \to \boldsymbol{v}(\boldsymbol{\epsilon}) \quad \text{as } \pi \to \mathbf{0}. \tag{3.61}$$

Hence, $\boldsymbol{v}^{\pi}(\boldsymbol{\epsilon})$ is a smooth approximation of $\boldsymbol{v}(\boldsymbol{\epsilon})$, based on which we propose to approximate Problem (3.46) as

$$\begin{aligned} \min_{\mathrm{z}} \quad & \|\epsilon\|_1 + 0.5\sigma \,\|\epsilon^w - \boldsymbol{q}^{\pi}(\epsilon^w)\|_2^2 && \text{(3.62a)} \\ \text{s.t.} \quad & \boldsymbol{c}^{\pi}(\epsilon^x) + \boldsymbol{d}^{\pi}(\epsilon^w) - \epsilon^x = \mathbf{0}, && \text{(3.62b)} \\ & A^{\mathrm{E}}\mathrm{z} = b^{\mathrm{E}}, && \text{(3.62c)} \\ & \boldsymbol{l}^{\pi}(\epsilon^x) + \boldsymbol{m}^{\pi}(\epsilon^w) \le g, && \text{(3.62d)} \\ & A^{\mathrm{I}}\mathrm{z} \le b^{\mathrm{I}}. && \text{(3.62e)} \end{aligned}$$

for some $\pi > 0$, and solve Problem (3.62) for reducing values of π such that we solve Problem (3.46) at termination.

To solve Problem (3.46) for a given π, we propose to use an NLP approach, that requires the evaluation of values and sensitivities of $\boldsymbol{v}^{\pi}(\boldsymbol{\epsilon})$ for any given $\boldsymbol{\epsilon} \geq \mathbf{0}$. For a given $\boldsymbol{\epsilon} \geq \mathbf{0}$, $\boldsymbol{v}^{\pi}(\boldsymbol{\epsilon}) = \mathbf{r}^{\top}\mathbf{z}^{\pi}$ can be evaluated by solving the KKT conditions of Problem (3.59) as

$$\mathcal{R}(\mathbf{z}^{\pi}, \boldsymbol{\lambda}^{\pi}, \mathrm{s}^{\pi}, \boldsymbol{\epsilon}) := \begin{bmatrix} \mathbf{P}^{\top}\boldsymbol{\lambda}^{\pi} + \kappa_{\mathrm{p}}^{2}\mathbf{z}^{\pi} - \mathbf{r} \\ \mathbf{P}\mathbf{z}^{\pi} - \kappa_{\mathrm{d}}^{2}\boldsymbol{\lambda}^{\pi} + \mathrm{s}^{\pi} - \boldsymbol{\epsilon} \\ \boldsymbol{\lambda}^{\pi} \circ \mathrm{s}^{\pi} - \mu\mathbf{1} \end{bmatrix} = \mathbf{0}, \tag{3.63}$$

along with $\boldsymbol{\lambda}^{\pi}, \mathrm{s}^{\pi} > \mathbf{0}$, in which $\theta^{\pi} = -\kappa_{\mathrm{d}}\boldsymbol{\lambda}^{\pi}$ is used to eliminate θ^{π}, and s^{π} is the optimal slack variable. These conditions can be solved by Algorithm 1, where

$$\begin{array}{r} \mathbf{S}^{\pi} := \mathrm{diag}(\mathbf{s}^{\pi}), \\ \boldsymbol{\Lambda}^{\pi} := \mathrm{diag}(\boldsymbol{\lambda}^{\pi}), \end{array} \quad \partial\mathcal{R}(\boldsymbol{\lambda}^{\pi}, \mathrm{s}^{\pi}) := \begin{bmatrix} \kappa_{\mathrm{p}}^{2}\mathbf{I} & \mathbf{P}^{\top} & \mathbf{0} \\ \mathbf{P} & -\kappa_{\mathrm{d}}^{2}\mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{S}^{\pi} & \boldsymbol{\Lambda}^{\pi} \end{bmatrix}.$$

At the solution $(\mathbf{z}^{\pi}, \boldsymbol{\lambda}^{\pi}, \mathrm{s}^{\pi})$, the sensitivities, i.e.,

$$\nabla_{\boldsymbol{\epsilon}}\boldsymbol{v}^{\pi}(\boldsymbol{\epsilon}) = \mathbf{r}^{\top}\frac{\partial\mathbf{z}^{\pi}}{\partial\boldsymbol{\epsilon}}, \qquad \nabla_{\boldsymbol{\epsilon}\boldsymbol{\epsilon}}^{2}\boldsymbol{v}^{\pi}(\boldsymbol{\epsilon}) = \sum_{l=1}^{\boldsymbol{n}}\mathbf{r}_{l}\frac{\partial^{2}\mathbf{z}_{l}^{\pi}}{\partial\boldsymbol{\epsilon}^{2}}. \tag{3.64}$$

can then be evaluated using the IFT [38]. The first-order sensitivities can be obtained by solving the linear system

$$\partial\mathcal{R}(\boldsymbol{\lambda}^{\pi}, \mathrm{s}^{\pi}) \begin{bmatrix} \partial\mathbf{z}^{\pi}/\partial\boldsymbol{\epsilon} \\ \partial\boldsymbol{\lambda}^{\pi}/\partial\boldsymbol{\epsilon} \\ \partial\mathrm{s}^{\pi}/\partial\boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix}, \tag{3.65}$$

and the second-order sensitivities can be obtained by solving the following linear system [23] for each component $j \in \mathbb{I}_{1}^{\boldsymbol{m}}$:

$$\partial\mathcal{R}(\boldsymbol{\lambda}^{\pi}, \mathrm{s}^{\pi}) \begin{bmatrix} \frac{\partial^{2}\mathbf{z}^{\pi}}{\partial\boldsymbol{\epsilon}\boldsymbol{\epsilon}_{j}} \\ \frac{\partial^{2}\boldsymbol{\lambda}^{\pi}}{\partial\boldsymbol{\epsilon}\boldsymbol{\epsilon}_{j}} \\ \frac{\partial^{2}\mathrm{s}^{\pi}}{\partial\boldsymbol{\epsilon}\boldsymbol{\epsilon}_{j}} \end{bmatrix} = -\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathrm{diag}(\frac{\partial\mathbf{S}^{\pi}}{\partial\boldsymbol{\epsilon}_{j}})\frac{\partial\boldsymbol{\lambda}^{\pi}}{\partial\boldsymbol{\epsilon}} + \mathrm{diag}(\frac{\partial\boldsymbol{\Lambda}^{\pi}}{\partial\boldsymbol{\epsilon}_{j}})\frac{\partial\mathbf{s}^{\pi}}{\partial\boldsymbol{\epsilon}} \end{bmatrix}. \tag{3.66}$$

In the following, we argue that as $\pi \to 0$, the sensitivities of $\boldsymbol{v}^{\pi}(\boldsymbol{\epsilon})$ computed as in (3.65) converge to the generalized gradient of $\boldsymbol{v}(\boldsymbol{\epsilon})$. We focus on an intuitive definition of these objects in the sequel, referring the reader to [156] for a formal treatment. Using (3.63), (3.64) and (3.65), we write

$$\begin{aligned}
\nabla_{\boldsymbol{\epsilon}} \boldsymbol{v}^{\pi}(\boldsymbol{\epsilon}) &= \boldsymbol{\lambda}^{\pi\top} \mathbf{P} \frac{\partial \mathbf{z}^{\pi}}{\partial \boldsymbol{\epsilon}} + \kappa_{\mathrm{p}}^2 \mathbf{z}^{\pi\top} \frac{\partial \mathbf{z}^{\pi}}{\partial \boldsymbol{\epsilon}} \\
&= \boldsymbol{\lambda}^{\pi\top} - \boldsymbol{\lambda}^{\pi\top} \frac{\partial \mathbf{s}^{\pi}}{\partial \boldsymbol{\epsilon}} + \left(\kappa_{\mathrm{p}}^2 \mathbf{z}^{\pi\top} \frac{\partial \mathbf{z}^{\pi}}{\partial \boldsymbol{\epsilon}} + \kappa_{\mathrm{d}}^2 \boldsymbol{\lambda}^{\pi\top} \frac{\partial \boldsymbol{\lambda}^{\pi}}{\partial \boldsymbol{\epsilon}} \right) \\
&= \boldsymbol{\lambda}^{\pi\top} - \mu \mathbf{1}^{\top} (\boldsymbol{\Lambda}^{\pi})^{-1} \frac{\partial \boldsymbol{\lambda}^{\pi}}{\partial \boldsymbol{\epsilon}} + \left(\kappa_{\mathrm{p}}^2 \mathbf{z}^{\pi\top} \frac{\partial \mathbf{z}^{\pi}}{\partial \boldsymbol{\epsilon}} + \kappa_{\mathrm{d}}^2 \boldsymbol{\lambda}^{\pi\top} \frac{\partial \boldsymbol{\lambda}^{\pi}}{\partial \boldsymbol{\epsilon}} \right).
\end{aligned} \tag{3.67}$$

For every $\boldsymbol{\epsilon} \geq \mathbf{0}$ and $\pi = (\kappa_{\mathrm{p}}, \kappa_{\mathrm{d}}, \mu) > 0$, the values of $(\mathbf{z}^{\pi}, \boldsymbol{\lambda}^{\pi}, \mathbf{s}^{\pi})$ solving (3.63) and their sensitivities with respect to $\boldsymbol{\epsilon}$ solving (3.65) are well-defined. Hence, from (3.67), we get

$$\nabla_{\boldsymbol{\epsilon}} \boldsymbol{v}^{\pi}(\boldsymbol{\epsilon}) \to \boldsymbol{\lambda}^{\pi\top} \text{ as } \pi \to 0. \tag{3.68}$$

We recall from (3.60) that $\boldsymbol{\lambda}^{\pi} \to \boldsymbol{\lambda}^{\kappa}$ as $\mu \to \mathbf{0}$. From **Part 1** in Proposition 3.4 and [84, Theorem 2], it can be shown that $\boldsymbol{\lambda}^{\kappa} \to \boldsymbol{\lambda}^{\kappa_{\mathrm{d}}}$ as $\kappa_{\mathrm{p}} \to \mathbf{0}$. Moreover, $\boldsymbol{\lambda}^{*} = \boldsymbol{\lambda}^{\kappa_{\mathrm{d}}}$ for any $\kappa_{\mathrm{d}} \in [0, \bar{\kappa}_{\mathrm{d}}]$ from **Part 2** in Proposition 3.4. Hence, $\boldsymbol{\lambda}^{\pi} \to \boldsymbol{\lambda}^{*}$ as $\pi \to 0$. This implies that $\nabla_{\boldsymbol{\epsilon}} \boldsymbol{v}^{\pi}(\boldsymbol{\epsilon}) \to \boldsymbol{\lambda}^{*\top}$ as $\pi \to 0$ from (3.68). We know from parametric linear programming [12] that every optimal dual variable $\boldsymbol{\lambda}^{*}$ of Problem (3.47) belongs to the generalized gradient of $\boldsymbol{v}(\boldsymbol{\epsilon})$ with respect to $\boldsymbol{\epsilon}$. Hence, as the approximation parameter $\pi \to 0$, the gradients of the proposed smoothening approximation $\boldsymbol{v}^{\pi}(\boldsymbol{\epsilon})$ converge to the generalized gradient of the nonsmooth function $\boldsymbol{v}(\boldsymbol{\epsilon})$, such that the gradient consistency property discussed in [156, Definition 2.6] is satisfied. This property ensures that by solving Problem (3.62) as an NLP for reducing values of $\pi \to 0$, with sensitivities evaluated as in (3.65) and (3.66), we approach a stationary point of the original nonsmooth Problem (3.46).

Feasibility of Problem (3.62)

We must ensure that Problem (3.62) is feasible for all $\pi > 0$. To this end, we prove the following result.

Proposition 3.5. *Given some $\pi = (\kappa_{\mathrm{p}}, \kappa_{\mathrm{d}}, \mu) > 0$, if there exists some $\epsilon^{w} \geq \mathbf{0}$ such that $\boldsymbol{d}^{\pi}(\epsilon^{w}) \geq \mathbf{0}$, then there exists an ϵ^{x} satisfying* (3.62b), *i.e.,*

$$\boldsymbol{c}^{\pi}(\epsilon^{x}) + \boldsymbol{d}^{\pi}(\epsilon^{w}) = \epsilon^{x}.$$

Algorithm 1: Solve KKT conditions in (3.63)

Result: Return $(\mathbf{z}^\pi, \boldsymbol{\lambda}^\pi, \mathbf{s}^\pi)$;
Input: $\pi > 0$, $\boldsymbol{\epsilon} \geq \mathbf{0}$, Initial guess $(\mathbf{z}^\pi, \boldsymbol{\lambda}^\pi > \mathbf{0}, \mathbf{s}^\pi > \mathbf{0})$;
while $\|\mathcal{R}(\mathbf{z}^\pi, \boldsymbol{\lambda}^\pi, \mathbf{s}^\pi, \boldsymbol{\epsilon})\|_\infty > \delta_{tol}$ **do**

1. Solve $\partial\mathcal{R}(\boldsymbol{\lambda}^\pi, \mathbf{s}^\pi)\begin{bmatrix}\Delta\mathbf{z}\\ \Delta\boldsymbol{\lambda}\\ \Delta\mathbf{s}\end{bmatrix} = -\mathcal{R}(\mathbf{z}^\pi, \boldsymbol{\lambda}^\pi, \mathbf{s}^\pi, \boldsymbol{\epsilon})$;
2. Compute largest $\alpha \in (0, 1]$ by backtracking such that $\boldsymbol{\lambda}^\pi + \alpha\Delta\boldsymbol{\lambda} > \mathbf{0}$ and $\mathbf{s}^\pi + \alpha\Delta\mathbf{s} > \mathbf{0}$;
3. Update $\mathbf{z}^\pi \leftarrow \mathbf{z}^\pi + \alpha\Delta\mathbf{z}$, $\boldsymbol{\lambda}^\pi \leftarrow \boldsymbol{\lambda}^\pi + \alpha\Delta\boldsymbol{\lambda}$, $\mathbf{s}^\pi \leftarrow \mathbf{s}^\pi + \alpha\Delta\mathbf{s}$;

end

□

Proof. Given some $\epsilon^w \geq \mathbf{0}$ such that $\boldsymbol{d}^\pi(\epsilon^w) \geq \mathbf{0}$, we observe from (3.63) that (3.62b) is feasible if and only if there exist variables

$$\{\epsilon^x, \mathrm{z}^{\boldsymbol{c}_i,\pi}, \lambda^{\boldsymbol{c}_i,\pi}, s^{\boldsymbol{c}_i,\pi}, \forall i \in \mathbb{I}_1^{m_X}\}$$

solving the set of equations formulating the KKT conditions as

$$E_i A\mathrm{z}^{\boldsymbol{c}_i,\pi} + \boldsymbol{d}_i^\pi(\epsilon^w) - \epsilon_i^x = 0, \tag{3.69a}$$
$$E^\top\lambda^{\boldsymbol{c}_i,\pi} + \kappa_\mathrm{p}^2\mathrm{z}^{\boldsymbol{c}_i,\pi} - A^\top E_i^\top = \mathbf{0}, \tag{3.69b}$$
$$E\mathrm{z}^{\boldsymbol{c}_i,\pi} - \kappa_\mathrm{d}^2\lambda^{\boldsymbol{c}_i,\pi} + s^{\boldsymbol{c}_i,\pi} - \epsilon^x = \mathbf{0}, \tag{3.69c}$$
$$\lambda^{\boldsymbol{c}_i,\pi} \circ s^{\boldsymbol{c}_i,\pi} - \mu\mathbf{1} = \mathbf{0}, \tag{3.69d}$$

along with $\lambda^{\boldsymbol{c}_i,\pi}, s^{\boldsymbol{c}_i,\pi} > \mathbf{0}$ for all $i \in \mathbb{I}_1^{m_X}$. In the sequel, we show that such variables exist for every $\pi > \mathbf{0}$. For brevity, we denote by $\boldsymbol{d}, z^i, \lambda^i$, and s^i, respectively, $\boldsymbol{d}^\pi(\epsilon^w), \mathrm{z}^{\boldsymbol{c}_i,\pi}, \lambda^{\boldsymbol{c}_i,\pi}$, and $s^{\boldsymbol{c}_i,\pi}$. Introducing $\boldsymbol{c}_i = E_i A z^i$ into (3.69) and eliminating (3.69a), we equivalently obtain

$$\boldsymbol{c}_i - E_i A z^i = 0, \tag{3.70a}$$
$$E^\top\lambda^i + \kappa_\mathrm{p}^2 z^i - A^\top E_i^\top = \mathbf{0}, \tag{3.70b}$$
$$E z^i - \kappa_\mathrm{d}^2\lambda^i + s^i - \boldsymbol{c} - \boldsymbol{d} = \mathbf{0}, \tag{3.70c}$$
$$\lambda^i \circ s^i - \mu\mathbf{1} = \mathbf{0}. \tag{3.70d}$$

Hence, we focus on demonstrating the existence of variables

$$\{\boldsymbol{c}, z^i, \lambda^i, s^i, \forall i \in \mathbb{I}_1^{m_X}\}$$

solving (3.70) for any $\pi > 0$. To this end, we consider the following QP

$$\max_{z^i, \boldsymbol{c}, \theta^i} \sum_{i=1}^{m_X} \mathcal{J}(\boldsymbol{c}_i, z^i, \theta^i, \kappa_{\mathrm{p}_i}) \tag{3.71a}$$

$$\text{s.t. } \boldsymbol{c}_i - E_i A z^i = \boldsymbol{0}, \qquad \forall i \in \mathbb{I}_1^{m_X}, \tag{3.71b}$$

$$E z^i + \kappa_{\mathrm{d}_i} \theta^i - \boldsymbol{c} \leq \boldsymbol{d}, \qquad \forall i \in \mathbb{I}_1^{m_X}, \tag{3.71c}$$

where the objective function

$$\mathcal{J}(\boldsymbol{c}_i, z^i, \theta^i, \kappa_{\mathrm{p}_i}) := \boldsymbol{c}_i - 0.5 \left(\left\| \kappa_{\mathrm{p}_i} z^i \right\|_2^2 + \left\| \theta^i \right\|_2^2 \right),$$

and $\kappa_{\mathrm{p}_i}, \kappa_{\mathrm{d}_i} > 0$ are individual regularization constants for each $i \in \mathbb{I}_1^{m_X}$. Problem (3.71) can be approximated as

$$\max_{z^i, \boldsymbol{c}, \theta^i, s^i} \sum_{i=1}^{m_X} \left(\mathcal{J}(\boldsymbol{c}_i, z^i, \theta^i, \kappa_{\mathrm{p}_i}) + \mu \omega_i \sum_{j=1}^{m_X} \log(s_j^i) \right) \tag{3.72a}$$

$$\text{s.t. } \boldsymbol{c}_i - E_i A z^i = \boldsymbol{0}, \qquad \forall i \in \mathbb{I}_1^{m_X}, \tag{3.72b}$$

$$E z^i + \kappa_{\mathrm{d}_i} \theta^i - \boldsymbol{c} + s^i = \boldsymbol{d}, \qquad \forall i \in \mathbb{I}_1^{m_X}, \tag{3.72c}$$

where $\omega_i > 0$ are individual logarithmic-barrier weights selected separately for each $i \in \mathbb{I}_1^{m_X}$. Since Problem (3.71) is feasible (with all variables set to **0**), bounded (since it is a strongly convex quadratic program), and the feasible set contains a nonempty interior, the primal-dual solution

$$\{\{\boldsymbol{c}^*, (z^{i*}, \theta^{i*}, s^{i*})\}, \{\alpha_i^*, \beta^{i*}\}, i \in \mathbb{I}_1^{m_X}\}$$

of Problem (3.72) exists and is unique for every $(\kappa_{\mathrm{p}_i}, \kappa_{\mathrm{d}_i}, \omega_i) > 0$, where α^{i*}, β^{i*} are the optimal dual variables associated to constraints (3.72b) and (3.72c) respectively [89, Theorem 8.1]. These variables solve the KKT conditions of (3.72), written after eliminating $\theta^{i*} = -\kappa_{\mathrm{d}_i} \beta^{i*}$ as

$$\alpha_i^* - 1 - \sum_{k=1}^{m_X} \beta_k^{i*} = 0, \tag{3.73a}$$

$$\boldsymbol{c}_i^* - E_i A z^{i*} = 0, \tag{3.73b}$$

$$E^\top \beta^{i*} + \kappa_{\mathrm{p}_i}^2 z^{i*} - \alpha_i^* A^\top E_i^\top = \boldsymbol{0}, \tag{3.73c}$$

$$E z^{i*} - \kappa_{\mathrm{d}_i}^2 \beta^{i*} + s^{i*} - \boldsymbol{c}^* - \boldsymbol{d} = \boldsymbol{0}, \tag{3.73d}$$

$$\beta^{i*} \circ s^{i*} - \omega_i \mu \boldsymbol{1} = \boldsymbol{0}, \tag{3.73e}$$

along with $\beta^{i*}, s^{i*} > \mathbf{0}$, for each $i \in \mathbb{I}_1^{m_X}$. Since $\alpha_i^* \geq 1$ from (3.73a), we can select the regularization and barrier parameters

$$\kappa_{\mathrm{p}_i} = \sqrt{\alpha_i^*}\kappa_{\mathrm{p}}, \qquad \kappa_{\mathrm{d}_i} = \kappa_{\mathrm{d}}/\sqrt{\alpha_i^*}, \qquad \omega_i = \alpha_i^*. \tag{3.74}$$

Then, Equations (3.73b)-(3.73e) are such that (3.70) is solved by

$$\{\boldsymbol{c}, z^i, \lambda^i, s^i\} = \{\boldsymbol{c}^*, z^{i*}, \beta^{i*}/\alpha_i^*, s^{i*}\},$$

thus concluding the proof. □

Remark 3.6. *The result in Proposition 3.5 is independent of Assumption 3.2, i.e., there exists an ϵ^x satisfying $\boldsymbol{c}^\pi(\epsilon^x) + \boldsymbol{d}^\pi(\epsilon^w) = \epsilon^x$ for any $\pi > 0$, even if there exists no finite ϵ^x satisfying the RPI condition*

$$\boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) \leq \boldsymbol{b}(\epsilon^x).$$

This is a consequence of the regularization parameters $\kappa_{\mathrm{p}_i}, \kappa_{\mathrm{d}_i} > 0$ that guarantee that Problem (3.71) *is bounded. If Assumption 3.2 does not hold, we know from [144] that Problem* (3.71) *with $(\kappa_{\mathrm{p}_i}, \kappa_{\mathrm{d}_i}) = 0$ is unbounded above. In this case,*

$$\sum_{i=1}^{m_X} \epsilon_i^x \to \infty, \quad \textit{as} \quad (\kappa_{\mathrm{p}_i}, \kappa_{\mathrm{d}_i}) \to 0$$

by continuity of the optimal value of Problem (3.71). □

3.4.3 Solution algorithm

Since Problem (3.62) is smooth, it can be solved by using standard NLP techniques such as Sequential Quadratic Programming and PDIP methods. In this section, we present an algorithm based on the PDIP method [102] to solve Problem (3.46), by approximately solving Problem (3.62) for reducing values of π. In order to recall the PDIP method, we write Problem (3.62) for simplicity as

$$\begin{aligned} \min_{\mathrm{z}} \quad & f^\pi(\mathrm{z}) \\ \text{s.t.} \quad & h_{\mathrm{E}}^\pi(\mathrm{z}) = \mathbf{0}, \\ & h_{\mathrm{I}}^\pi(\mathrm{z}) \leq \mathbf{0}, \end{aligned} \tag{3.75}$$

where $f^{\pi}(\mathrm{z})$ denotes the objective (3.62a), $h^{\pi}_{\mathrm{E}}(\mathrm{z}) = \mathbf{0}$ denotes the equality constraints (3.62b) and (3.62c), and $h^{\pi}_{\mathrm{I}}(\mathrm{z}) \leq \mathbf{0}$ denotes the inequality constraints (3.62d) and (3.62e). Let $\gamma := [\gamma_{\mathrm{E}}^{\top}\ \gamma_{\mathrm{I}}^{\top}]^{\top}$ with γ_{E} and γ_{I} denote the dual variables associated with constraints $h^{\pi}_{\mathrm{E}}(\mathrm{z}) = \mathbf{0}$ and $h^{\pi}_{\mathrm{I}}(\mathrm{z}) \leq \mathbf{0}$ respectively. The Lagrangian of Problem (3.75) is

$$\mathcal{L}^{\pi}(\mathrm{z}, \gamma) := f^{\pi}(\mathrm{z}) + \gamma_{\mathrm{E}}^{\top} h^{\pi}_{\mathrm{E}}(\mathrm{z}) + \gamma_{\mathrm{I}}^{\top} h^{\pi}_{\mathrm{I}}(\mathrm{z}).$$

PDIP methods aim at finding stationary points that satisfy SOSC of the PDIP-KKT conditions of Problem (3.75) written as

$$\mathcal{K}^{\tau,\pi}(\mathrm{z}, \gamma, \zeta) := \begin{bmatrix} \nabla_{\mathrm{z}} \mathcal{L}^{\pi}(\mathrm{z}, \gamma) \\ h^{\pi}_{\mathrm{E}}(\mathrm{z}) \\ h^{\pi}_{\mathrm{I}}(\mathrm{z}) + \zeta \\ \zeta \circ \gamma_{\mathrm{I}} - \tau \mathbf{1} \end{bmatrix} = \mathbf{0}, \tag{3.76}$$

along with $\zeta, \gamma_{\mathrm{I}} > \mathbf{0}$, for some barrier parameter $\tau > 0$ using suitably adapted Newton iterates. It is useful to observe that the equations in (3.76) can be reformulated as the KKT conditions of the primal interpretation of the log-barrier approach written as

$$\begin{aligned} \min_{\mathrm{z},\zeta} \quad & f^{\pi}(\mathrm{z}) - \tau \sum_{n=1}^{n_{\mathrm{I}}} \log(\zeta_n) \\ \text{s.t.} \quad & h^{\pi}_{\mathrm{E}}(\mathrm{z}) = \mathbf{0}, \\ & h^{\pi}_{\mathrm{I}}(\mathrm{z}) + \zeta = \mathbf{0}. \end{aligned} \tag{3.77}$$

As $\tau \to 0$, the solution of Problem (3.77) approaches that of Problem (3.75). Hence, PDIP methods progressively reduce the barrier parameter τ, such that at convergence, Problem (3.75), i.e., Problem (3.62) is solved. In our approach, along with reducing the value of τ, we also reduce π such that, at convergence, Problem (3.46) is solved.

Algorithm 2 summarizes the proposed PDIP solution method. In Step 1, the functions $\boldsymbol{v}^{\pi}(\boldsymbol{\epsilon})$ and their sensitivities are evaluated, and Step 2 builds the necessary vectors, gradients, and the Hessian of the Lagrangian $\nabla^2_{\mathrm{zz}} \mathcal{L}^{\pi}$. In Step 3, $\nabla^2_{\mathrm{zz}} \mathcal{L}^{\pi}$ is regularized to ensure that it is positive definite in the nullspace of $\nabla_{\mathrm{z}} h^{\pi}_{\mathrm{E}}$, thus guaranteeing descent. In Step 4, a linear system is solved to compute a Newton direction. Then in Step

5, an upper-bound to the step length $\bar{\alpha}$ is computed to ensure positivity of ζ and γ_{I}. In Step 6, a globalization method is employed to compute a steplength $\alpha \in (0, \bar{\alpha}]$, e.g., a line-search or filter method [102], and the variables are finally updated in Step 7. In our solver, along with updating the barrier parameter τ using some $\theta_\tau \in (0,1)$, we also update the smoothening parameter π using some $\theta_\pi \in (0,1)$. In Steps 1 and 7, we always warm-start the variables $(\mathbf{z}^\pi, \boldsymbol{\lambda}^\pi, \mathbf{s}^\pi)$ using previously computed values to evaluate $\boldsymbol{v}^\pi(\boldsymbol{\epsilon})$ using Algorithm 1.

We make the following assumptions to ensure that the smoothening parameters $\pi = (\kappa_{\mathrm{p}}, \kappa_{\mathrm{d}}, \mu) > 0$ are chosen such that Problem (3.75) is feasible, thus guaranteeing that the multipliers γ computed over the iterations of Algorithm 2 are bounded.

Assumption 3.5. *The values of the smoothening parameter $\pi > 0$ formulating Problem* (3.75) *and chosen in Algorithm 2 are such that: (a) there exists an $\epsilon^w \geq \mathbf{0}$ such that $\boldsymbol{d}^\pi(\epsilon^w) \geq \mathbf{0}$; (b) the pair (ϵ^x, ϵ^w) satisfying* (3.62b)*, i.e., $\boldsymbol{c}^\pi(\epsilon^x) + \boldsymbol{d}^\pi(\epsilon^w) = \epsilon^x$ also satisfies* (3.62d)*, i.e., $\boldsymbol{l}^\pi(\epsilon^x) + \boldsymbol{m}^\pi(\epsilon^w) \leq g$.* $\square$

Assumption 3.5(a) is necessary to satisfy the hypothesis of Proposition 3.5, and can typically be satisfied by choosing a small $\mu > 0$: Given some $\epsilon^w \geq \mathbf{0}$ and $\pi > 0$, let $(\mathbf{z}^{\boldsymbol{d}_i,\pi}, \boldsymbol{\lambda}^{\boldsymbol{d}_i,\pi}, \mathbf{s}^{\boldsymbol{d}_i,\pi})$ be the optimal primal, dual, and slack variables solving the KKT conditions in (3.63). Since $\boldsymbol{d}_i^\pi(\epsilon^w) = E_i B \mathbf{z}^{\boldsymbol{d}_i,\pi}$, it follows that

$$\boldsymbol{d}_i^\pi(\epsilon^w) = {\boldsymbol{\lambda}^{\boldsymbol{d}_i,\pi}}^\top \epsilon^w + \kappa_{\mathrm{p}}^2 ||\mathbf{z}^{\boldsymbol{d}_i,\pi}||_2^2 + \kappa_{\mathrm{d}}^2 ||\boldsymbol{\lambda}^{\boldsymbol{d}_i,\pi}||_2^2 - \mu m_X.$$

Since $\boldsymbol{\lambda}^{\boldsymbol{d}_i,\pi} > 0$ for every $\pi > 0$ and $\epsilon^w \geq \mathbf{0}$, it holds that

$${\boldsymbol{\lambda}^{\boldsymbol{d}_i,\pi}}^\top \epsilon^w + \kappa_{\mathrm{p}}^2 ||\mathbf{z}^{\boldsymbol{d}_i,\pi}||_2^2 + \kappa_{\mathrm{d}}^2 ||\boldsymbol{\lambda}^{\boldsymbol{d}_i,\pi}||_2^2 > 0.$$

Then, a small $\mu > 0$ can ensure $\boldsymbol{d}_i^\pi(\epsilon^w) \geq 0$. While a similar reasoning can be applied to guarantee the Assumption 3.5(b) is satisfied, a characterization of the set of values π satisfying Assumption 3.5(b) is left for future research. We observed in our numerical examples that Assumption 3.5(b) was never violated.

As we will demonstrate in the examples, using Algorithm 2 to solve Problem (3.62) can help in reaching a solution using a reduced number of iterations, as compared to approaches based on the KKT-reformulation. The main drawbacks of the proposed method are the following:

- A procedure to obtain a feasible initial point is not defined. This is because, for a given $\pi > 0$ and $\epsilon^w \geq \mathbf{0}$ satisfying Assumption 3.5, we cannot compute an ϵ^x satisfying (3.62b), i.e.,

$$\boldsymbol{c}^\pi(\epsilon^x) + \boldsymbol{d}^\pi(\epsilon^w) = \epsilon^x,$$

  using the result in Proposition 3.5. The reason is that the value of the optimal dual variables α_i^* required to select the parameters in (3.74) for given π are not known a priori, since α_i^* depends on $\{\kappa_{\mathrm{p}_i}, \kappa_{\mathrm{d}_i}, \omega_i\}$ and vice-versa. Hence, in our implementation, we use a scaling-based procedure described in [133, 97] to (infeasibly) initialize ϵ^w and ϵ^x;.

- While a pre-factorized KKT matrix of $\boldsymbol{v}^\pi(\boldsymbol{\epsilon})$ results in almost-free computation of the sensitivities in (3.65), building the Hessian matrix $\nabla^2_{\boldsymbol{\epsilon\epsilon}}\boldsymbol{v}^\pi(\boldsymbol{\epsilon})$ with these derivatives can be expensive with respect to memory requirements.

3.5 Numerical examples

3.5.1 Computation of safe reference-sets for supervisory control

We consider the system

$$z(t+1) = \begin{bmatrix} 1.1 & 0.2 \\ -0.3 & 0.4 \end{bmatrix} z(t) + \begin{bmatrix} 1 & 0 \\ 0.1 & 1 \end{bmatrix} u(t)$$

with input-constraints $u \in \hat{\mathbf{U}} := \{u : |u| \leq [2\ 1.5]^\top\}$, and equipped with an LQI-tracking controller such that z tracks a reference signal w: an integral-action state q with dynamics $q(t+1) = q(t) + z(t) - w(t)$ is appended, and the state $x = [z^\top\ q^\top]^\top$ is introduced. Then, an LQI feedback gain

$$K = \begin{bmatrix} -1.19 & -0.1439 & -0.3154 & 0.0213 \\ 0.2777 & -0.6497 & -0.0037 & -0.3724 \end{bmatrix}$$

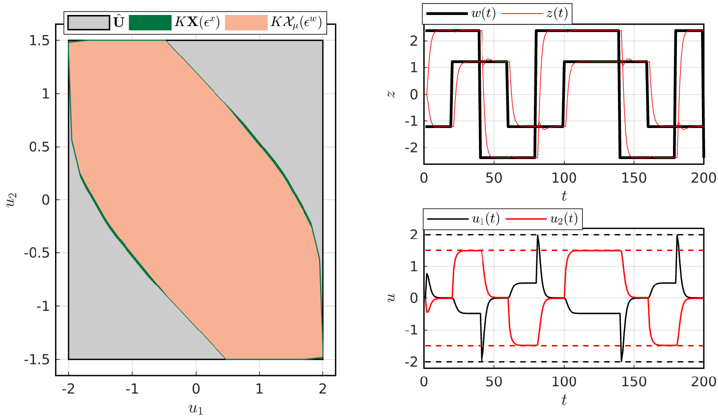


Figure 2: Results of solving Problem (3.46). Tight RPI set $\mathcal{X}_\mu(\epsilon^w)$ is computed with $\mu = 10^{-6}$. Top-right plot shows the tracking performance with w sampled from the vertices of $\mathbb{W}(\epsilon^w)$. Bottom-right plot shows resulting closed-loop inputs.

is computed corresponding to matrices $Q = \text{diag}(\mathbf{I}, 0.5\mathbf{I})$ and $R = \mathbf{I}$. The resulting closed-loop system with $u = Kx$ has the dynamics

$$x(t+1) = \begin{bmatrix} -0.09 & 0.0561 & -0.3154 & 0.0213 \\ -0.1413 & -0.2641 & -0.0353 & -0.3702 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} w(t).$$

We aim to design a supervisory controller for this system that saturates the references as

$$w \in \mathbb{W}(\epsilon^w) = \{w : |w| \leq \bar{\epsilon}^w\}$$

such that $u \in \hat{\mathbf{U}}$ always holds. *We assume that the supervisory controller cannot access the state $x(t)$ of the system, such that $\mathbb{W}(\epsilon^w)$ should guarantee input-constraint satisfaction for all reachable x.*

Since the mRPI set $\mathcal{X}_\text{m}(\epsilon^w)$ is the set of set of all reachable x, the constraint $u \in \hat{\mathbf{U}}$ is equivalent to

$$K\mathcal{X}_\text{m}(\epsilon^w) \subseteq \hat{\mathbf{U}}.$$

Hence, we solve Problem (3.46) with the output equation (3.1b) formulated using $C = K, D = 0$, output-constraint set $\mathcal{Y} = \hat{\mathbf{U}}$, and the mRPI set $\mathcal{X}_{\mathrm{m}}(\epsilon^w)$ approximated using the RPI set $\mathbb{X}(\epsilon^x) = \{x : Ex \leq \epsilon^x\}$, where the matrix E is composed of hyperplanes defining the set $\oplus_{t=0}^{4} A^t B \mathbb{W}(\mathbf{1})$ (A,B denote the matrices of the closed-loop system). This choice results in $m_X = 112$. The result of solving this problem using Algorithm 2 is shown in Figure 2 (Plotted using the MPT-toolbox [53]). The computed saturation bounds are $\bar{w}_1 = 2.3819$, $\bar{w}_2 = 1.2108$.

We also plot the set $K\mathcal{X}_{\mu}(\epsilon^w)$, where $\mathcal{X}_{\mu}(\epsilon^w)$ is an RPI set satisfying

$$\mathcal{X}_{\mu}(\epsilon^w) \subseteq \mathcal{X}_{\mathrm{m}}(\epsilon^w) \oplus \mu\mathcal{B}_{\infty}^{n_x}.$$

This set is computed using the method in [111] for $\mu = 10^{-6}$. Using this set and the triangle inequality, we compute $d_{\mathrm{H}}(\mathcal{X}_{\mathrm{m}}(\epsilon^w), \mathbb{X}(\epsilon^x)) \leq 0.3007$, indicating that $\mathbb{X}(\epsilon^x)$ is a fairly tight approximation of the mRPI set. Closed-loop trajectories are plotted with references w sampled from the vertices of $\mathbb{W}(\epsilon^w)$, for which the input response satisfies the input-constraints. Hence, if $x(0) \in \mathbb{X}(\epsilon^x)$, the supervisory controller can command any reference $w \in \mathbb{W}(\epsilon^w)$ with guaranteed input-constraint satisfaction.

Remark 3.7. *The mRPI set is suitable to formulate the problem in Example A since we do not have access to the state $x(t)$. If this limitation is overcome, then a reference governor scheme [40] is more suitable to design the supervisory controller, which uses control invariant sets to guarantee constraint satisfaction.* $\square$

3.5.2 Comparison of Algorithm 2 with a lifted KKT-based approach

In this subsection, we employ Algorithm 2 to solve 16 randomly generated instances of Problem (3.46). The dimensions of these instances are reported in Table 2. In our implementation, we set: $\sigma = 10^{-6}$ in objective (3.46a); initialize $\mu = 10^{-3}$, κ_{p}, $\kappa_{\mathrm{d}} = 10^{-2}$, $\tau = 10^{-2}$; update with $\theta_\pi, \theta_\tau = 0.1$; set tolerances for τ and μ as 10^{-7} and 10^{-8}, respectively; update $\kappa_{\mathrm{p}} \leftarrow \theta_\pi \kappa_{\mathrm{p}}$ and $\kappa_{\mathrm{d}} \leftarrow \theta_\pi \kappa_{\mathrm{d}}$ only when $\tau = 10^{-7}$ and $\mu = 10^{-8}$; set tolerance for $\kappa_{\mathrm{p}}, \kappa_{\mathrm{d}}$ equal to 10^{-4}; use Hessian regularization based on eigenvalue modification of the reduced Hessian [102, Page 50] in Step

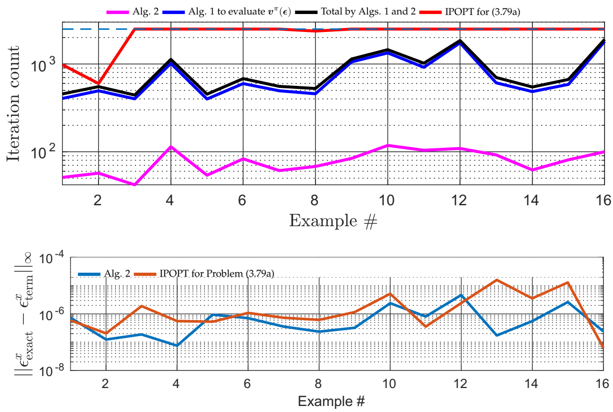


Figure 3: (Top) Comparison of the number of iterations; (Bottom) Error in RPI constraint (3.44b) satisfaction.

3; use the merit function proposed in [107] within the line-search procedure in Step 6. We terminate the algorithm with parameters $\tau = 10^{-7}$, $\mu = 10^{-8}$, $\kappa_{\text{p}}, \kappa_{\text{d}} = 10^{-4}$ to ensure that $\boldsymbol{v}^\pi(\boldsymbol{\epsilon})$ are close to $\boldsymbol{v}(\boldsymbol{\epsilon})$ at termination. Our computational experience suggests that maintaining $\mu < \tau$ facilitates faster convergence: This choice implies that $\boldsymbol{v}^\pi(\boldsymbol{\epsilon})$ is close $\boldsymbol{v}(\boldsymbol{\epsilon})$ while tracking the central path of Problem (3.62). A formalization of this reasoning is left for future study. When updating κ_{p}, κ_{d}, we start reducing the regularization parameters $\kappa_{\text{p}}^2, \kappa_{\text{d}}^2$ from 10^{-4} to 10^{-8} close to the solution, i.e., when $\tau = 10^{-7}$ and $\mu = 10^{-8}$, in order to avoid numerical difficulties resulting in inaccurate sensitivity evaluations. At termination of Algorithm 1, we compare the values $\boldsymbol{v}(\boldsymbol{\epsilon})$ and $\boldsymbol{v}^\pi(\boldsymbol{\epsilon})$ in order to examine the impact of the smoothening parameter $\pi > 0$. We obtain $\|\boldsymbol{v}(\boldsymbol{\epsilon}) - \boldsymbol{v}^\pi(\boldsymbol{\epsilon})\|_\infty = 4.127.10^{-6}$ for the presented examples, with the norm taken over all the support functions $\boldsymbol{v} = \{\boldsymbol{c}_i, \boldsymbol{d}_i, \boldsymbol{l}_k, \boldsymbol{m}_k, \boldsymbol{q}_t\}$ in all 16 examples.

We compare the performance of Algorithm 2 with the KKT-based approach presented in [4]. In this approach, the LPs formulating Problem (3.46) are replaced by their KKT-optimality conditions, resulting in a lifted formulation. For numerical robustness, we consider the regularized LP (QP) in (3.51) in lieu of (3.47), such that $\boldsymbol{v}(\boldsymbol{\epsilon}) = \mathbf{r}^\top \mathbf{z}^*$ is replaced by $\boldsymbol{v}^\kappa(\boldsymbol{\epsilon}) = \mathbf{r}^\top \mathbf{z}^\kappa$. The KKT optimality conditions of Problem (3.51) can be written as

$$\begin{gathered}\mathbf{P}^\top \mathbf{z}^\kappa + \kappa_\mathrm{p}^2 \mathbf{z}^\kappa = \mathbf{r}, \qquad \mathbf{P}\mathbf{z}^\kappa - \kappa_\mathrm{d}^2 \boldsymbol{\lambda}^\kappa \leq \boldsymbol{\epsilon}, \qquad \boldsymbol{\lambda}^\kappa \geq \mathbf{0}, \\ \boldsymbol{\lambda}^{\kappa\top} \boldsymbol{\epsilon} - \mathbf{r}^\top \mathbf{z}^\kappa + \kappa_\mathrm{p}^2 \left\|\mathbf{z}^\kappa\right\|_2^2 + \kappa_\mathrm{d}^2 \left\|\boldsymbol{\lambda}^\kappa\right\|_2^2 = 0,\end{gathered} \tag{3.78}$$

in which the fourth condition is the zero-duality gap condition that is equivalent to the strict-complementarity condition. Substituting the optimality conditions in (3.78) for each $\boldsymbol{v}^\pi(\boldsymbol{\epsilon})$ in Problem (3.46), we obtain the NLP

$$\min_{\mathrm{z},\mathrm{z}^*,\lambda^* \geq \mathbf{0}} \quad \|\epsilon\|_1 + \sigma \sum_{t=1}^{m_W} (\epsilon_t^w - F_t \mathrm{z}^{\boldsymbol{q}_t})^2 \tag{3.79a}$$

$$\begin{aligned}
\text{s.t. } & E_i A \mathrm{z}^{\boldsymbol{c}_i} + E_i B \mathrm{z}^{\boldsymbol{d}_i} = \epsilon_i^x, \\
& E \mathrm{z}^{\boldsymbol{c}_i} - \kappa_\mathrm{d}^2 \lambda^{\boldsymbol{c}_i} \leq \epsilon^x, \; F \mathrm{z}^{\boldsymbol{d}_i} - \kappa_\mathrm{d}^2 \lambda^{\boldsymbol{d}_i} \leq \epsilon^w, \\
& \begin{bmatrix} E^\top \lambda^{\boldsymbol{c}_i} + \kappa_\mathrm{p}^2 \mathrm{z}^{\boldsymbol{c}_i} \\ F^\top \lambda^{\boldsymbol{d}_i} + \kappa_\mathrm{p}^2 \mathrm{z}^{\boldsymbol{d}_i} \end{bmatrix} = \begin{bmatrix} A^\top E_i^\top \\ B^\top E_i^\top \end{bmatrix}, \\
& \begin{bmatrix} {\epsilon^x}^\top \lambda^{\boldsymbol{c}_i} - E_i A \mathrm{z}^{\boldsymbol{c}_i} + \kappa_\mathrm{p}^2 \left\|\mathrm{z}^{\boldsymbol{c}_i}\right\|_2^2 + \kappa_\mathrm{d}^2 \left\|\lambda^{\boldsymbol{c}_i}\right\|_2^2 \\ {\epsilon^w}^\top \lambda^{\boldsymbol{d}_i} - E_i B \mathrm{z}^{\boldsymbol{d}_i} + \kappa_\mathrm{p}^2 \left\|\mathrm{z}^{\boldsymbol{d}_i}\right\|_2^2 + \kappa_\mathrm{d}^2 \left\|\lambda^{\boldsymbol{d}_i}\right\|_2^2 \end{bmatrix} = \mathbf{0}, \\
& G_k C \mathrm{z}^{\boldsymbol{l}_k} + G_k D \mathrm{z}^{\boldsymbol{m}_k} \leq g_k, \\
& E \mathrm{z}^{\boldsymbol{l}_k} - \kappa_\mathrm{d}^2 \lambda^{\boldsymbol{l}_k} \leq \epsilon^x, \; F \mathrm{z}^{\boldsymbol{m}_k} - \kappa_\mathrm{d}^2 \lambda^{\boldsymbol{m}_k} \leq \epsilon^w, \\
& \begin{bmatrix} E^\top \lambda^{\boldsymbol{l}_k} + \kappa_\mathrm{p}^2 \mathrm{z}^{\boldsymbol{l}_k} \\ F^\top \lambda^{\boldsymbol{m}_k} + \kappa_\mathrm{p}^2 \mathrm{z}^{\boldsymbol{m}_k} \end{bmatrix} = \begin{bmatrix} C^\top G_k^\top \\ D^\top G_k^\top \end{bmatrix}, \\
& \begin{bmatrix} {\epsilon^x}^\top \lambda^{\boldsymbol{l}_k} - G_k C \mathrm{z}^{\boldsymbol{l}_k} + \kappa_\mathrm{p}^2 \left\|\mathrm{z}^{\boldsymbol{l}_k}\right\|_2^2 + \kappa_\mathrm{d}^2 \left\|\lambda^{\boldsymbol{l}_k}\right\|_2^2 \\ {\epsilon^w}^\top \lambda^{\boldsymbol{m}_k} - G_k D \mathrm{z}^{\boldsymbol{m}_k} + \kappa_\mathrm{p}^2 \left\|\mathrm{z}^{\boldsymbol{m}_k}\right\|_2^2 + \kappa_\mathrm{d}^2 \left\|\lambda^{\boldsymbol{m}_k}\right\|_2^2 \end{bmatrix} = \mathbf{0}, \\
& F \mathrm{z}^{\boldsymbol{q}_t} - \kappa_\mathrm{d}^2 \lambda^{\boldsymbol{q}_t} \leq \epsilon^w, \\
& F^\top \lambda^{\boldsymbol{q}_t} + \kappa_\mathrm{p}^2 \mathrm{z}^{\boldsymbol{q}_t} = F_t^\top, \\
& {\epsilon^w}^\top \lambda^{\boldsymbol{q}_t} - F_k \mathrm{z}^{\boldsymbol{q}_t} + \kappa_\mathrm{p}^2 \left\|\mathrm{z}^{\boldsymbol{q}_t}\right\|_2^2 + \kappa_\mathrm{d}^2 \left\|\lambda^{\boldsymbol{q}_t}\right\|_2^2 = 0, \\
& \forall i \in \mathbb{I}_1^{m_X}, \; \forall k \in \mathbb{I}_1^{m_Y}, \; \forall t \in \mathbb{I}_1^{m_W}, \\
& A^\mathrm{E} \mathrm{z} = b^\mathrm{E}, \quad A^\mathrm{I} \mathrm{z} \leq b^\mathrm{I},
\end{aligned}$$

where $z^* \in \mathbb{R}^{m_X(n_x+n_w)+m_Y(n_x+n_w)+m_W n_w}$ are the primal variables and

$\lambda^* \in \mathbb{R}^{m_X(m_X+m_W)+m_Y(m_X+m_W)+m_W^2}$ are the dual variables satisfying (3.78). We solve the NLP in (3.79a) using the IPOPT [151] interior-point solver. We supply the solver with exact evaluations of the gradients and the Hessian, select a global tolerance level of 10^{-7} for termination, and terminate the algorithm after 2500 iterations if the solution has not yet been found.

The results of initializing both Algorithm 2 and the NLP (3.79a) at the same point are shown in Table 2. In the table, we indicate the problem dimensions, and the values of $\|\epsilon\|_1$ at the termination of Algorithm 2 and IPOPT for solving (3.79a). The dimensions of the matrices $E \in \mathbb{R}^{m_X \times n_x}$, $F \in \mathbb{R}^{m_W \times n_w}$, and $H \in \mathbb{R}^{m_B \times n_y}$ selected for these examples are:

$$n_x, n_w = 2 \Rightarrow m_X, m_W = 20,$$

$$n_x, n_w = 3 \Rightarrow m_X, m_W = 84,$$

$$n_x, n_w = 4 \Rightarrow m_X, m_W = 160,$$

$$n_y = 2 \Rightarrow m_B = 6,\ n_y = 3 \Rightarrow m_B = 42.$$

We observe that the values of $\|\epsilon\|_1$ at termination are close for the two algorithms. In Figure 3 (top plot), we compare the number of iterations for the termination of the algorithms. For Algorithm 2, we plot both the upper and lower level iterations. The lower iterations count the total number of interior-point iterations performed in Steps 1 and 6 for evaluating $\boldsymbol{v}^\pi(\boldsymbol{\epsilon})$ using Algorithm 1 with $\delta_{\text{tol}} = 10^{-12}$. We note that these iterations are significantly cheaper than the PDIP iterations to solve Problem (3.79a), Moreover, they can be parallelized reducing computational time further. The upper iterations count the interior-point iterations as described in Algorithm 2. We observe that in the majority of cases, IPOPT terminates at the maximum iteration limit, i.e., the tolerance of 10^{-7} is not met even after 2500 iterations. While the obtained value at termination is feasible for Problem (3.79a), local optima are not reached. This is especially pronounced for higher dimensional systems, e.g., Cases 9-16. In the case of Algorithm 2, however, we observe much quicker convergence. In Figure 3 (bottom plot), using the obtained value of ϵ^w at termination, we recompute ϵ^x that satisfies the RPI constraint

$\boldsymbol{c}(\epsilon^x) + \boldsymbol{d}(\epsilon^w) = \epsilon^x$. We label this vector as $\epsilon^x_{\text{exact}}$, and evaluate the error with respect to ϵ^x_{term}, i.e., the value of ϵ^x at the termination of the algorithms. We observe that the regularization terms $\kappa_{\text{p}}^2, \kappa_{\text{d}}^2 = 10^{-8}$ at the termination of both the algorithms do not have a significant effect on the RPI constraint satisfaction.

Acknowledgment

The authors would like to thank Saša V. Raković for constructive comments clarifying Lemma 3.1.1.

Algorithm 2: Smoothing-based PDIP algorithm for Problem (3.46)

Result: Return z;

Input: System matrices (A, B, C, D); Output-constraint set $\mathcal{Y}$ in hyperplane and vertex notations; Matrices E, F, H parameterizing polytopes $\mathbb{X}(\epsilon^x), \mathbb{W}(\epsilon^w), \mathbb{B}(\epsilon)$ respectively;

Initialize: $\tau, \pi > 0$, $(\mathrm{z}, \gamma, \zeta)$, $(\mathbf{z}^\pi, \boldsymbol{\lambda}^\pi, \mathbf{s}^\pi)$;

while τ,π,$\|\nabla_{\mathrm{z}}\mathcal{L}^\pi, h_{\mathrm{E}}^\pi, \max(\mathbf{0}, h_{\mathrm{I}}^\pi)\|_\infty > \delta_{tol}$ **do**

1. Evaluate $\boldsymbol{v}^\pi(\boldsymbol{\epsilon}), \nabla_{\boldsymbol{\epsilon}}\boldsymbol{v}^\pi(\boldsymbol{\epsilon}), \nabla^2_{\boldsymbol{\epsilon\epsilon}}\boldsymbol{v}^\pi(\boldsymbol{\epsilon})$ for each $\boldsymbol{v}^\pi(\boldsymbol{\epsilon}) = \{\boldsymbol{c}_i^\pi(\epsilon^x), \boldsymbol{d}_i^\pi(\epsilon^w), \boldsymbol{l}_k^\pi(\epsilon^x), \boldsymbol{m}_k^\pi(\epsilon^w), \boldsymbol{q}_t^\pi(\epsilon^w)\}$ using Algorithm 1, and Equations (3.65) and (3.66);
2. Compute f^π, h_{E}^π, h_{I}^π, $\nabla_{\mathrm{z}} f^\pi$, $\nabla_{\mathrm{z}} h_{\mathrm{E}}^\pi$, $\nabla_{\mathrm{z}} h_{\mathrm{I}}^\pi$, $\nabla^2_{\mathrm{zz}}\mathcal{L}^\pi$;
3. Compute regularized Hessian $\mathcal{B}$ approximating $\nabla^2_{\mathrm{zz}}\mathcal{L}^\pi$ and ensuring descent for Problem (3.77);
4. Compute Newton direction by solving

$$\begin{bmatrix} \mathcal{B} & \nabla_{\mathrm{z}} h_{\mathrm{E}}^\pi & \nabla_{\mathrm{z}} h_{\mathrm{I}}^\pi & \mathbf{0} \\ \nabla_{\mathrm{z}} {h_{\mathrm{E}}^\pi}^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \nabla_{\mathrm{z}} {h_{\mathrm{I}}^\pi}^\top & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathcal{Z} & \mathcal{G}_{\mathrm{I}} \end{bmatrix} \begin{bmatrix} \Delta \mathrm{z} \\ \Delta\gamma_{\mathrm{E}} \\ \Delta\gamma_{\mathrm{I}} \\ \Delta\zeta \end{bmatrix} = - \begin{bmatrix} \nabla_{\mathrm{z}}\mathcal{L}^\pi \\ h_{\mathrm{E}}^\pi \\ h_{\mathrm{I}}^\pi + \zeta \\ \zeta \circ \gamma_{\mathrm{I}} - \tau\mathbf{1} \end{bmatrix},$$

   where $\mathcal{Z} := \mathrm{diag}(\zeta)$, $\mathcal{G}_{\mathrm{I}} := \mathrm{diag}(\gamma_{\mathrm{I}})$;
5. Compute largest $\bar{\alpha} \in (0, 1]$ such that $\zeta + \bar{\alpha}\Delta\zeta \geq \bar{\epsilon}\zeta$ and $\gamma_{\mathrm{I}} + \bar{\alpha}\Delta\gamma_{\mathrm{I}} \geq \bar{\epsilon}\gamma_{\mathrm{I}}$ for some $\bar{\epsilon} > 0$;
6. Globalization method yielding $\alpha \in (0, \bar{\alpha}]$;
7. Update $(\mathrm{z}, \gamma, \zeta) \leftarrow (\mathrm{z}, \gamma, \zeta) + \alpha(\Delta\mathrm{z}, \Delta\gamma_{\mathrm{E}}, \Delta\gamma_{\mathrm{I}}, \Delta\zeta)$;

if $\|\mathcal{K}^{\tau,\pi}(\mathrm{z}, \gamma, \zeta)\|_\infty < \tau$ **then**

Update $\tau \leftarrow \theta_\tau\tau$, $\pi \leftarrow \theta_\pi\pi$

end

end

#	(n_x, n_w, n_y)	Algorithm 2	IPOPT for (3.79a)
1	$(2, 2, 2)$	2.7422	2.7422
2	$(2, 2, 3)$	33.3250	33.2500
3	$(2, 3, 2)$	1.4002	1.4706
4	$(2, 3, 3)$	16.9777	16.8023
5	$(2, 4, 2)$	2.7422	3.0815
6	$(2, 4, 3)$	27.0213	26.8366
7	$(3, 2, 2)$	2.0244	2.9159
8	$(3, 2, 3)$	21.3772	21.0908
9	$(3, 3, 2)$	2.1299	3.0646
10	$(3, 3, 3)$	25.2940	26.6438
11	$(3, 4, 2)$	20.8378	28.6038
12	$(4, 2, 3)$	0.8741	1.0193
13	$(4, 2, 2)$	27.3955	27.6824
14	$(4, 3, 3)$	1.8457	2.1947
15	$(4, 3, 2)$	27.2488	27.2586
16	$(4, 4, 3)$	22.1434	25.0673

Table 2: Dimension of random systems, along with a comparison of the values of $\|\epsilon\|_1$ at the termination of Algorithm 2 and NLP (3.79a) when solved with IPOPT. For each example, the target output set is $\mathcal{Y} = \mathcal{B}_\infty^{n_y}$.

Chapter 4

Computation of Least-Conservative State-Constraint Sets for Decentralized MPC with Dynamic and Constraint Coupling

4.1 Introduction

Model Predictive Control (MPC) of interconnected systems has been an active area of research, driven by practical requirements posed by communication and computation limitations [8]. Several control schemes satisfying these requirements have been proposed, which are based on decomposition methods of either the coupled system or of the centralized optimization problem [82]. These schemes are broadly divided into two categories: *distributed* MPC (DMPC) and *decentralized* MPC (DeMPC), with the division usually being defined based on the communication between the controllers. With respect to interconnection patterns, the two broad categories are *dynamic coupling* and *constraint coupling*.

Dynamic couplings lead to interactions between the states of disparate constituent subsystems, thus requiring coordination between local controllers. Tube-based MPC [87] has been used as an effective framework to tackle this coordination problem. By modeling the state interactions as local disturbances, local controllers can be designed that explicitly take these disturbances into account to ensure robust constraint satisfaction. An example that uses this framework is the DMPC scheme proposed in [35], which accommodates both dynamic and constraint coupling. This scheme requires communication between the controllers of reference trajectories, and true states and inputs. On the DeMPC side, schemes that do not require communication between the controllers have been proposed. The lack of communication introduces unavoidable conservativeness, which should be tackled is a structured way. For example, in [120], local tube-based MPC controllers are synthesized using feedback gains, which are computed by solving an offline optimization problem that minimizes the conservativeness. However, the scheme only accommodates dynamic coupling and not constraint coupling. A common theme among these approaches is the adoption of the method presented in [112] to compute tight outer approximations of the minimal Robust Positive Invariant (mRPI) set, which is an essential ingredient of tube-based MPC. Thus, broadly the current problem belongs to the broader class of problems related to computing decentralized RPI sets. For example, in [117], RPI sets for dynamically coupled systems with *summarized information* regarding the connected subsystems are characterized and computed. In [28], such information is used to progressively shrink ellipsoidal terminal PI sets. Without making such summarized information assumptions, LMI-based methods are presented in [101] to compute fully-decentralized zonotopic RPI sets for systems with decoupled constraints, along with corresponding invariance inducing feedback gains. While this approach can be extended to accommodate systems with coupled constraints, a comparison is a subject of future study. In [43], zonotopic RCI sets are synthesized an a similar setting by not fixing the controller parameterization to a linear law, and adopting a compositional framework.

Recently, building on the work presented in [118], a one-step approach to compute outer approximations of the mRPI set has been presented in [144]. This approach, which allows for very quick online recomputation of a small RPI set, has been purposed in the development of a DMPC scheme in [145]. The recomputation leads to disturbance sets which reduce in size as the set-points are reached, therefore improving the performance of the overall distributed scheme.

In this chapter, we present a method to compute state-constraint sets for local tube-based MPC controllers [87] used within the DeMPC scheme of [120]. We consider linear time-invariant systems which can be coupled in both dynamics and constraints. The method is centered on the formulation of an offline optimization problem, which is developed using a set-based framework. The decoupled state-constraint sets are computed such that (a) the corresponding output set is the least conservative inner-approximation of the coupled constraint set, and (b) feasibility and stability of the local tube-based MPC controllers is ensured. The formulation of the optimization problem relies on the results on polytopically parameterized RPI sets developed in Chapter 3.

4.2 Decentralized Tube-Based MPC of Coupled Linear Systems

4.2.1 System Description

We consider a linear time-invariant system of the form

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{4.1}$$

with state $\mathbf{x} \in \mathbb{R}^{n_x}$, input $\mathbf{u} \in \mathbb{R}^{n_u}$. This system is subject to constraints

$$U := \{\mathbf{u} : G^{\mathrm{u}}\mathbf{u} \leq g^{\mathrm{u}}\}, \qquad g^{\mathrm{u}} \in \mathbb{R}^{m_U}, \tag{4.2a}$$

$$\mathcal{Y} := \{\mathbf{y} \in \mathbb{R}^{n_y} : \mathbf{y} = \mathbf{C}\mathbf{x}, G^{\mathrm{y}}\mathbf{y} \leq g^{\mathrm{y}}\}, \qquad g^{\mathrm{y}} \in \mathbb{R}^{m_Y}. \tag{4.2b}$$

Assumption 4.1. *The sets U and $\mathcal{Y}$ are full-dimensional polytopes containing the origin in their interior.* $\square$

We assume that the system in (4.1) can be partitioned into M subsystems, each with dynamics described by

$$x_{[i]}(t+1) = A_{[ii]}x_{[i]}(t) + B_{[i]}u_{[i]}(t) + \sum_{j \in L_i} A_{[ij]}x_{[j]}(t), \tag{4.3}$$

where i indicates the i^{th} subsystem with states $x_{[i]} \in \mathbb{R}^{n_x^i}$ and inputs $u_{[i]} \in \mathbb{R}^{n_u^i}$. The overall state and input vectors are then

$$\mathbf{x}(t) = [x_{[1]}(t)^\top, .., x_{[M]}(t)^\top]^\top,$$
$$\mathbf{u}(t) = [u_{[1]}(t)^\top, .., u_{[M]}(t)^\top]^\top.$$

respectively. The set L_i indicates the indices of the neighbors of i which are dynamically coupled to it, i.e.,

$$L_i := \{j \in M : i \neq j,\ A_{[ij]} \neq \mathbf{0}\}.$$

From (4.3), we have $\mathbf{B} = \mathrm{diag}(B_{[1]}, \cdots, B_{[M]})$. In addition, we assume that the input constraints are decoupled, i.e., $U = \prod_{i \in M} U_i$, with $u_{[i]} \in U_i$ being the input constraint on individual subsystem i. Note that unlike in related literature [110, 120, 62], we do not assume $\mathcal{Y}$ to be decoupled between the subsystems. Our aim is to solve the following fully decentralized MPC problem:

Problem 4.1. *Design M model predictive controllers $\mathcal{C}_i$, one per subsystem i, described by* (4.3), *such that*

(a) The state $\mathbf{x}$ is regulated to $\mathbf{0}$;

(b) System constraints (4.2) *are satisfied;*

(c) Each controller $\mathcal{C}_i$ has access only to local states $x_{[i]}$;

(d) There is no communication between the controllers.

In order to solve Problem 4.1, we adopt the DeMPC scheme of [120], which uses the tube-based MPC approach [87] to design each controller

$\mathcal{C}_i$. In the original approach, state-constraint sets $\mathcal{X}_i$ on individual subsystem i are known a priori, while we only know the coupled constraint $\mathcal{Y}$. Hence, in the next subsection, we recall the scheme in [120] for arbitrary state-constraint sets $\mathcal{X}_i$, and use the properties of the scheme to derive requirements on $\mathcal{X}_i$ in order to satisfy the coupled constraint $\mathcal{Y}$.

4.2.2 DeMPC Formulation

In order to formulate the controller as decentralized, we model all couplings as disturbances. Accordingly, we rewrite (4.3) as

$$x_{[i]}(t+1) = A_{[ii]}x_{[i]}(t) + B_{[i]}u_{[i]}(t) + w_{[i]}(t), \tag{4.4}$$

that is subject to the additive disturbance

$$w_{[i]}(t) := \sum_{j \in L_i} A_{[ij]}x_{[j]}(t).$$

As in standard tube-based MPC (recall from Section 2.2.3), we assume that each subsystem i is equipped with a *pre-designed* feedback controller $K_{[i]} \in \mathbb{R}^{n_u^i \times n_x^i}$ satisfying the following assumptions.

Assumption 4.2.

1. *Each matrix pair* $(A_{[ii]}, B_{[i]})$ *is controllable;*
2. $\forall\, i \in \mathbb{I}_1^M,\ \rho(A_{K_{[i]}}) < 1$, *where* $A_{K_{[i]}} := A_{[ii]} + B_{[i]}K_{[i]}$*;*
3. $\rho(\mathbf{A_K}) < 1$, *where* $\mathbf{A_K} := \mathbf{A} + \mathbf{BK}$ *and* $\mathbf{K} := \mathrm{diag}(K_{[1]}, \cdots, K_{[M]})$.

$\square$

Using $K_{[i]}$, we parameterize the control input as

$$u_{[i]}(t) = \hat{u}_{[i]}(t) + K_{[i]}\Delta x_{[i]}(t), \tag{4.5}$$

where $\Delta x_{[i]}(t) := x_{[i]}(t) - \hat{x}_{[i]}(t)$ is the state error with respect to the *nominal* system

$$\hat{x}_{[i]}(t+1) = A_{[ii]}\hat{x}_{[i]}(t) + B_{[i]}\hat{u}_{[i]}(t). \tag{4.6}$$

We also define the input error $\Delta u_{[i]}(t) := u_{[i]}(t) - \hat{u}_{[i]}(t)$. Using (4.4), (4.5) and (4.6), the dynamics of the *error* system for subsystem i can be derived as

$$\Delta x_{[i]}(t+1) = A_{K_{[i]}} \Delta x_{[i]}(t) + w_{[i]}(t). \tag{4.7}$$

Since we assume that the state of each subsystem i is constrained to the set $\mathcal{X}_i$, we have

$$w_{[i]}(t) \in \mathcal{W}_i := \bigoplus_{j \in L_i} A_{[ij]} \mathcal{X}_j. \tag{4.8}$$

Given the disturbance set $\mathcal{W}_i$, the error state $\Delta x_{[i]}$ always belongs to the corresponding mRPI set $\Delta\mathcal{X}_i(\mathcal{W}_i)$:

$$\Delta x_{[i]}(t) \in \Delta\mathcal{X}_i(\mathcal{W}_i) := \bigoplus_{t=0}^{\infty} \left(A_{K_{[i]}}\right)^t \mathcal{W}_i. \tag{4.9}$$

In the following, we first formulate tube-based robust MPC controllers $\mathcal{C}_i$ by relying on sets $\mathcal{X}_i$ and $\Delta\mathcal{X}_i(\mathcal{W}_i)$ and afterwards discuss the properties that these sets must satisfy in order to guarantee that $\mathbf{y} \in \mathcal{Y}$. Each $\mathcal{C}_i$ is based on solving

$$\begin{aligned}
\min_{\mathbf{z}_{[i]}} \; & \sum_{s=t}^{t+N-1} \left\|\hat{x}_{[i]}(s)\right\|^2_{Q_{[i]}} + \left\|\hat{u}_{[i]}(s)\right\|^2_{R_{[i]}} + \left\|\hat{x}_{[i]}(t+N)\right\|^2_{P_{[i]}} && \text{(4.10a)} \\
\text{s.t. } \; & x_{[i]}(t) \in \hat{x}_{[i]}(t) \oplus \Delta\mathcal{X}_i(\mathcal{W}_i), && \text{(4.10b)} \\
& \hat{x}_{[i]}(s+1) = A_{[ii]}\hat{x}_{[i]}(s) + B_{[i]}\hat{u}_{[i]}(s), \qquad s \in \mathbb{I}_t^{t+N_{[i]}-1}, && \text{(4.10c)} \\
& \hat{x}_{[i]}(s) \in \mathcal{X}_i \ominus \Delta\mathcal{X}_i(\mathcal{W}_i), \qquad s \in \mathbb{I}_{t+1}^{t+N_{[i]}-1}, && \text{(4.10d)} \\
& \hat{u}_{[i]}(s) \in U_i \ominus K_{[i]}\Delta\mathcal{X}_i(\mathcal{W}_i), \qquad s \in \mathbb{I}_t^{t+N_{[i]}-1}, && \text{(4.10e)} \\
& \hat{x}_{[i]}(t+N) \in \mathcal{X}_i^{\mathrm{t}}, && \text{(4.10f)}
\end{aligned}$$

with the optimization vector

$$\mathbf{z}_{[i]} := \{\hat{x}_{[i]}(t), \cdots, \hat{x}_{[i]}(t+N), \hat{u}_{[i]}(t), \cdots, \hat{u}_{[i]}(t+N-1)\}.$$

The nominal model (4.6) is used to perform predictions of state evolutions, as indicated in (4.10c). The initial state is left as a free variable to be optimized through constraint (4.10b), and the predicted state and input constraints are tightened through constraints (4.10d) and (4.10e), such that the actual subsystem state $x_{[i]}(t) \in \mathcal{X}_i$ and $u_{[i]}(t) \in U_i$ for all t. The feedback gain $K_{[i]}$ is chosen to be the terminal control law, and terminal set $\mathcal{X}_i^{\mathrm{t}}$ is chosen to be a feasible Positive Invariant (PI) set for the autonomous system

$$\hat{x}_{[i]}(t+1) = A_{[i]}^{K} \hat{x}_{[i]}(t),$$

i.e., it satisfies the PI inclusion

$$A_{[i]}^{\mathrm{K}} \mathcal{X}_i^{\mathrm{t}} \subseteq \mathcal{X}_i^{\mathrm{t}},$$

while belonging to the tightened constraint set as

$$\mathcal{X}_i^{\mathrm{t}} \subseteq \{\hat{x}_{[i]} : \hat{x}_{[i]} \in \mathcal{X}_i \ominus \Delta\mathcal{X}_i(\mathcal{W}_i),\ K_{[i]}\hat{x}_{[i]} \in U_i \ominus K_{[i]}\Delta\mathcal{X}_i(\mathcal{W}_i)\}.$$

The matrices $Q_{[i]} > 0$ and $R_{[i]} > 0$ are chosen such that $K_{[i]}$ is the associated LQ control gain for nominal system i, and $P_{[i]}$ is the solution of the corresponding Discrete Algebraic Riccati Equation. Upon solving (6.8), control input $u_{[i]}(t) = \hat{u}_{[i]}(t) + K_{[i]}(x_{[i]}(t) - \hat{x}_{[i]}(t))$ is applied to the plant.

We recall the properties of the DeMPC scheme from [120], and derive requirements on $\mathcal{X}_i$ in the following result.

Proposition 4.1. *Suppose Assumptions 4.1 and 4.2 hold, and for each $i \in \mathbb{I}_1^M$, sets $\mathcal{X}_i$ satisfy*

$$\Delta\mathcal{X}_i(\mathcal{W}_i) \subseteq \mathrm{int}(\mathcal{X}_i), \tag{4.11a}$$

$$K_{[i]}\Delta\mathcal{X}_i(\mathcal{W}_i) \subseteq \mathrm{int}(U_i). \tag{4.11b}$$

(a) For each controller $\mathcal{C}_i$, we denote the feasible set

$$\mathcal{X}_i^N := \{x_{[i]} :\ \text{(4.10b)-(4.10f) feasible for } x_{[i]}(t) = x_{[i]}\}.$$

Then if $x_{[i]}(0) \in \mathcal{X}_i^N$, the controlled system in (4.3) satisfies $x_{[i]}(t) \in \mathcal{X}_i$ and $u_{[i]}(t) \in U_i$ for all $t \geq 0$, and the state $\mathbf{x}$ of System (4.1) is asymptotically stabilized to the origin with region of attraction $\Pi_{i\in M}\mathcal{X}_i^N$.

(b) Defining $\mathbf{C}_{[i]} \in \mathbb{R}^{n_y \times n_x^i}$ as the matrix composed of columns of matrix $\mathbf{C}$ multiplying states $x_{[i]}$ of subsystem i, if the inclusion

$$\bigoplus_{i=0}^{M} \mathbf{C}_{[i]}\mathcal{X}_i \subseteq \mathcal{Y}. \tag{4.12}$$

is satisfied, then the controllers $\mathcal{C}_i$ solve Problem 4.1.

$\square$

Proof.

(a) The conditions in (4.11) ensure that constraint sets in (4.10d) and (4.10e) are non-empty. This leads to non-empty feasible sets $\mathcal{X}_i^N$. For a given subsystem i, if $w_{[i]}(t) \in \mathcal{W}_i$ for all $t \geq 0$, then the proof of recursive feasibility and asymptotic stability of state $x_{[i]}$ of subsystem i to the mRPI set $\Delta\mathcal{X}_i(\mathcal{W}_i)$ follows from [87]. Since for every $i \in \mathbb{I}_1^M$, $x_{[i]}(0) \in \mathcal{X}_i^N$ implies $x_{[i]}(t) \in \mathcal{X}_i$ for all $t \geq 0$ because of closed-loop action by the controller $\mathcal{C}_i$ on subsystem i, it then indeed holds that $w_{[i]}(t) \in \mathcal{W}_i$ for all $t \geq 0$ by definition of $w_{[i]}$. Hence, each substate $x_{[i]}$ is asymptotically stabilized to $\Delta\mathcal{X}_i(\mathcal{W}_i)$, such that the state $\mathbf{x}$ of the overall system (4.1) is asymptotically stabilized to $\Pi_{i\in M}\Delta\mathcal{X}_i(\mathcal{W}_i)$ from any $\mathbf{x} \in \Pi_{i\in M}\mathcal{X}_i^N$. Then, the proof of stability to the origin follows from the observation that the set $\Pi_{i\in M}\Delta\mathcal{X}_i(\mathcal{W}_i)$ is Positive Invariant for the closed-loop system $\mathbf{x}(t+1) = \mathbf{A_K}\mathbf{x}(t)$. For further details the reader is refered to [120, Theorem 1].

(b) The condition in (4.12) translates to

$$\mathcal{X}_i = \left\{ x_{[i]} : \forall\, x_{[j]} \in \mathcal{X}_j,\ \mathbf{C}_{[i]}x_{[i]} + \sum_{j\in L_i} \mathbf{C}_{[j]}x_{[j]} \in \mathcal{Y} \right\},\ \forall\, i \in \mathbb{I}_1^M.$$

This implies that if (4.12) holds, then $x_{[i]} \in \mathcal{X}_i$ for all i ensures $y \in \mathcal{Y}$. The fact that the former is guaranteed by Part (a) concludes the proof.

$\square$

From Assumptions 4.1 and 4.2, we see that requirements (4.11) and (4.12) can be satisfied by compact sets $\mathcal{X}_i$, which we compute in the next section.

Remark 4.1. *Note that requirement* (4.12) *results in conservative sets $\mathcal{X}_i$, since it enforces that the control applied to the subsystem must satisfy system constraints $\mathcal{Y}$, for every possible control applied by the neighbors. This is unavoidable, unless communication is introduced. In case full state information of all neighbors were available to $\mathcal{C}_i$, one could formulate the local constraint set as*

$$\left\{ x_{[i]} :\ \exists x_{[j]} \in \mathcal{X}_j,\ \mathbf{C}_{[i]}x_{[i]} + \sum_{j\in L_i} \mathbf{C}_{[j]}x_{[j]} \in \mathcal{Y} \right\} \supseteq \mathcal{X}_i.$$

4.3 Computation of State-Constraint Sets $\mathcal{X}_i$

In this section, we present a formulation and a solution procedure to compute the sets $\mathcal{X}_i$ that satisfy requirements (4.11) and (4.12). To this end, we introduce the system

$$\Delta\mathbf{x}(t+1) = \tilde{\mathbf{A}}\Delta\mathbf{x}(t) + \tilde{\mathbf{B}}\mathbf{x}(t), \tag{4.13}$$

where the matrices $\tilde{\mathbf{A}} := \mathrm{diag}(A^K_{[1]}, \cdots, A^K_{[M]})$ and

$$\tilde{\mathbf{B}} := \begin{bmatrix} \mathbf{0} & A_{[12]} & \cdots & A_{[1M]} \\ A_{[21]} & \mathbf{0} & \cdots & A_{[2M]} \\ \cdots & \cdots & \mathbf{0} & \cdots \\ A_{[M1]} & A_{[M2]} & \cdots & \mathbf{0} \end{bmatrix}$$

capture the dynamic coupling between the subsystems. For this system, we introduce the sets

$$\mathcal{X} := \prod_{i \in M} \mathcal{X}_i, \qquad \Delta\mathcal{X}(\mathcal{X}) := \prod_{i \in M} \Delta\mathcal{X}_i(\mathcal{W}_i).$$

The set $\Delta\mathcal{X}(\mathcal{X})$ is the mRPI set of states corresponding to the system (4.13) when driven by *disturbances* $\mathbf{x}(t) \in \mathcal{X}$, given by

$$\Delta\mathcal{X}(\mathcal{X}) := \bigoplus_{t=0}^{\infty} \tilde{\mathbf{A}}^t \tilde{\mathbf{B}}\mathcal{X}. \tag{4.14}$$

We make the following assumption on the sets $\mathcal{X}_i$.

Assumption 4.3. *Each $\mathcal{X}_i$ is a compact convex set containing the origin.* □

Assumption 4.3 implies that the mRPI set $\Delta\mathcal{X}(\mathcal{X})$ is a compact convex set containing the origin [63]. In order to encode inclusions (4.11), we introduce scalars $\phi_{\mathrm{x}}, \phi_{\mathrm{u}} \in [0, 1)$, and write the inclusions as

$$\Delta\mathcal{X}(\mathcal{X}) \subseteq \phi_{\mathrm{x}}\mathcal{X}, \qquad \mathbf{K}\Delta\mathcal{X}(\mathcal{X}) \subseteq \phi_{\mathrm{u}}U. \tag{4.15}$$

The values of ϕ_{x} and ϕ_{u} are tuning parameters which are related to the strength of dynamic coupling. Larger values correspond to increased permissible dynamic coupling, and hence increased size of sets $\mathcal{X}_j$. However, this also corresponds to increased dynamical disturbance from the neighboring subsystems, resulting in increased constraint tightening, and a larger stabilization region. Thus, the scalars ϕ_{x} and ϕ_{u} are tuning factors that must be selected depending on the application at hand.

Considering requirement (4.12), we first note that

$$\mathbf{C}\mathcal{X} = \bigoplus_{i=0}^{M} \mathbf{C}_{[i]}\mathcal{X}_i \subseteq \mathcal{Y}, \tag{4.16}$$

by definition of $\mathcal{X}$. Ideally, one would like to satisfy the inclusion

$$\mathbf{C}\mathcal{X} \subseteq \mathcal{Y}$$

with equality. This would imply that the sets $\mathcal{X}_i$ are chosen such that all the points in set $\mathcal{Y}$ are reachable by $\mathbf{C}\mathbf{x}$. This, however, might not be feasible given requirements (4.15). Hence, we choose to instead minimize the approximation error of the output-constraint set $\mathcal{Y}$ by the output-reachable set $\mathbf{C}\mathcal{X}$, similarly to Chapter 3. We recall that this approximation error is defined as

$$\begin{aligned} \mathrm{d}_{\mathcal{Y}}(\mathbf{C}\mathcal{X}) := \min_{\epsilon} \quad & \|\epsilon\|_1 \\ \text{s.t.} \quad & \mathcal{Y} \subseteq \mathbf{C}\mathcal{X} \oplus \mathbb{B}(\epsilon), \end{aligned}$$

where the normal vectors $\{H_i^\top, i \in \mathbb{I}_1^{m_B}\}$ of the set $\mathbb{B}(\epsilon) := \{y : Hy \leq \epsilon\}$ are user-specified. From these requirements, we obtain the following optimization problem for fixed scalars ϕ_{x} and ϕ_{u}:

$$\min_{\epsilon, \mathcal{X} = \Pi_{i \in M}\mathcal{X}_i} \quad \|\epsilon\|_1 \tag{4.17a}$$

$$\text{s.t.} \quad \Delta\mathcal{X}(\mathcal{X}) \subseteq \phi_{\mathrm{x}}\mathcal{X}, \tag{4.17b}$$

$$\mathbf{K}\Delta\mathcal{X}(\mathcal{X}) \subseteq \phi_{\mathrm{u}}U, \tag{4.17c}$$

$$\mathbf{C}\mathcal{X} \subseteq \mathcal{Y}, \tag{4.17d}$$

$$\mathcal{Y} \subseteq \mathbf{C}\mathcal{X} \oplus \mathbb{B}(\epsilon), \tag{4.17e}$$

$$\mathbf{0} \in \mathcal{X}_i, \quad i \in \mathbb{I}_1^M. \tag{4.17f}$$

Our approach explicitly tackles the issue of conservativeness discussed in Remark 4.1: The sets $\mathcal{X}_i$ are computed such that $\mathbf{C}\mathcal{X}$ is the largest feasible inner-approximation of $\mathcal{Y}$, i.e., $\mathrm{d}_{\mathcal{Y}}(\mathbf{C}\mathcal{X})$ is minimized while ensuring feasibility and stability of $\mathcal{C}_i$. This implies that the overall system output $\mathbf{y} = \mathbf{C}\mathbf{x}$ is restricted to the least conservative subset of $\mathcal{Y}$ when controllers $\mathcal{C}_i$ safely regulate the system state $\mathbf{x}$ to the origin.

Remark 4.2. *The structure of* (4.13) *follows from the assumption of a block-diagonal matrix* $\mathbf{B}$*, i.e., decoupled inputs. The approach can be extended to accommodate coupled inputs and input constraints through minor reformulations.* □

4.3.1 Finite-Dimensional Parameterization

As discussed in Chapter 3, Problem (4.17) is in general intractable since it is formulated with the mRPI set $\Delta\mathcal{X}(\mathcal{X})$ that in general evades an explicit representation. Hence, we apply the techniques proposed in Chapter 3 in order to solve it approximately. In particular, we parameterize the sets $\mathcal{X}_i$ and $\Delta\mathcal{X}_i(\mathcal{W}_i)$ as polytopes, such that the sets $\mathcal{X}$ and $\Delta\mathcal{X}(\mathcal{X})$ are cartesian products of these polytopes.

We parameterize each set $\mathcal{X}_i$ using a finite-dimensional vector $\epsilon^{x,i}$ as

$$\mathcal{X}_i = \mathbb{X}_i(\epsilon^{x,i}) := \{x_{[i]} : F^i x_{[i]} \leq \epsilon^{x,i}\},$$

where the normal vectors $\{(F^i_j)^\top \in \mathbb{R}^{n^i_x}, j \in \mathbb{I}_1^{m^i_X}\}$ are fixed a priori. Since the corresponding mRPI sets $\Delta\mathcal{X}_i(\mathcal{W}_i)$ are in general not finitely determined [16], we rely on an outer RPI approximation, parameterized using a finite-dimensional vector $\epsilon^{\Delta x,i}$ as

$$\Delta\mathcal{X}_i(\mathcal{W}_i) \subseteq \Delta\mathbb{X}_i(\epsilon^{\Delta x,i}) := \{\Delta x_{[i]} : E^i \Delta x_{[i]} \leq \epsilon^{\Delta x,i}\},$$

where the normal vectors $\{(E^i_j)^\top \in \mathbb{R}^{n^i_x}, j \in \mathbb{I}_1^{m^i_{\Delta X}}\}$ are fixed a priori. The overall state-constraint set is hence

$$\begin{aligned}\mathcal{X} = \mathbb{X}(\epsilon^x) : &= \left\{ \mathbf{x} \; : \; \begin{bmatrix} F^1 & & \\ & \ddots & \\ & & F^M \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} \epsilon^{x,1} \\ \vdots \\ \epsilon^{x,M} \end{bmatrix} \right\} \\ &= \{\mathbf{x} : F\mathbf{x} \leq \epsilon^x\},\end{aligned}$$

and the corresponding mRPI set is approximated as

$$\Delta\mathcal{X}(\mathbb{X}(\epsilon^x)) \subseteq \Delta\mathbb{X}(\epsilon^{\Delta x}) := \left\{ \Delta\mathbf{x} \ : \ \begin{bmatrix} E^1 & & \\ & \ddots & \\ & & E^M \end{bmatrix} \Delta\mathbf{x} \leq \begin{bmatrix} \epsilon^{\Delta x,1} \\ \vdots \\ \epsilon^{\Delta x,M} \end{bmatrix} \right\}$$
$$= \{\Delta\mathbf{x} : E\Delta\mathbf{x} \leq \epsilon^{\Delta x}\}.$$

For convenience of notation, we also define

$$m_X := \sum_{i=1}^{M} m_X^i, \qquad m_{\Delta X} := \sum_{i=1}^{M} m_{\Delta X}^i,$$

such that $\epsilon^x \in \mathbb{R}^{m_X}$ and $\epsilon^{\Delta x} \in \mathbb{R}^{m_{\Delta X}}$.

In order to obtain a close approximation of the equality constraint (4.14) for a given disturbance set $\mathbb{X}(\epsilon^x)$, we use the parameterized RPI set $\Delta\mathbb{X}(\epsilon^{\Delta x})$ that minimizes $d_{\mathrm{H}}(\Delta\mathcal{X}(\mathbb{X}(\epsilon^x)), \Delta\mathbb{X}(\epsilon^{\Delta x}))$.

Finally, since $\Delta\mathcal{X}(\mathbb{X}(\epsilon^x)) \subseteq \Delta\mathbb{X}(\epsilon^{\Delta x})$, we replace $\Delta\mathcal{X}(\mathbb{X}(\epsilon^x))$ by $\Delta\mathbb{X}(\epsilon^{\Delta x})$ in inclusions (4.15). Note that process noise can be accommodated in this framework through matrix $\tilde{\mathbf{B}}$ and parameterized sets $\mathcal{X}$.

In terms of the above parameterized sets, Problem (4.17) is approximated as the following bilevel optimization problem:

Problem 4.2.

$$\min_{\epsilon^x \geq \mathbf{0}, \epsilon} \quad \|\epsilon\|_1 \tag{4.18a}$$
$$\text{s.t.} \quad \Delta\mathbb{X}(\epsilon^{\Delta x}) \subseteq \phi_{\mathrm{x}}\mathbb{X}(\epsilon^x), \tag{4.18b}$$
$$\mathbf{K}\Delta\mathbb{X}(\epsilon^{\Delta x}) \subseteq \phi_{\mathrm{u}}U, \tag{4.18c}$$
$$\mathbf{C}\mathbb{X}(\epsilon^x) \subseteq \mathcal{Y}, \tag{4.18d}$$
$$\mathcal{Y} \subseteq \mathbf{C}\mathbb{X}(\epsilon^x) \oplus \mathbb{B}(\epsilon), \tag{4.18e}$$
$$\begin{aligned} \epsilon^{\Delta x} = \arg\min_{\underline{\epsilon}^{\Delta x}} \ & d_{\mathrm{H}}(\Delta\mathcal{X}(\mathbb{X}(\epsilon^x)), \Delta\mathbb{X}(\underline{\epsilon}^{\Delta x})), \\ \text{s.t.} \ & \tilde{\mathbf{A}}\Delta\mathbb{X}(\underline{\epsilon}^{\Delta x}) \oplus \tilde{\mathbf{B}}\mathbb{X}(\epsilon^x) \subseteq \Delta\mathbb{X}(\underline{\epsilon}^{\Delta x}). \end{aligned} \tag{4.18f}$$

4.3.2 Inclusion encoding

In order to encode the RPI inclusion

$$\tilde{\mathbf{A}}\Delta\mathbb{X}(\epsilon^{\Delta x}) \oplus \tilde{\mathbf{B}}\mathbb{X}(\epsilon^x) \subseteq \Delta\mathbb{X}(\epsilon^{\Delta x}),$$

we introduce the support-functions

$$\forall\, i \in \mathbb{I}_1^{m_{\Delta X}} \begin{cases} \boldsymbol{c}_i\left(\epsilon^{\Delta x}\right) := h_{\tilde{\mathbf{A}}\Delta\mathbb{X}(\epsilon^{\Delta x})}\left(E_i^\top\right), \\ \boldsymbol{d}_i\left(\epsilon^{x}\right) := h_{\tilde{\mathbf{B}}\mathbb{X}(\epsilon^{x})}\left(E_i^\top\right), \\ \boldsymbol{b}_i\left(\epsilon^{\Delta x}\right) := h_{\Delta\mathbb{X}(\epsilon^{\Delta x})}\left(E_i^\top\right), \end{cases}$$

such that the RPI inclusion can equivalently be written as

$$\boldsymbol{c}(\underline{\epsilon}^{\Delta x}) + \boldsymbol{d}(\epsilon^{x}) \leq \boldsymbol{b}(\underline{\epsilon}^{\Delta x}). \tag{4.19}$$

Then, assuming that for the system in (4.13) the matrices E and F satisfy Assumption 3.2, we know from Theorem 3.1 that

$$\text{(4.18f)} \iff \boldsymbol{c}(\epsilon^{\Delta x}) + \boldsymbol{d}(\epsilon^{x}) = \epsilon^{\Delta x}, \tag{4.20}$$

such that the we replace the lower-level problem by the equality

$$\boldsymbol{c}(\epsilon^{\Delta x}) + \boldsymbol{d}(\epsilon^{x}) = \epsilon^{\Delta x}.$$

In order to encode inclusions (4.18b), (4.18c) and (4.18d), we introduce the support-functions

$$\begin{aligned} &\forall\, i \in \mathbb{I}_1^{m_X} \begin{cases} \boldsymbol{g}_i^{\mathrm{x}}\left(\epsilon^{\Delta x}\right) := h_{\Delta\mathbb{X}(\epsilon^{\Delta x})}\left(E_i^\top\right), \\ \boldsymbol{f}_i^{\mathrm{x}}\left(\epsilon^{x}\right) := h_{\mathbb{X}(\epsilon^{x})}\left(E_i^\top\right), \end{cases} \\ &\forall\, i \in \mathbb{I}_1^{m_U} \left\{ \boldsymbol{g}_i^{\mathrm{u}}\left(\epsilon^{\Delta x}\right) := h_{\mathbf{K}\Delta\mathbb{X}(\epsilon^{\Delta x})}\left({G_i^{\mathrm{u}}}^\top\right), \right. \\ &\forall\, i \in \mathbb{I}_1^{m_Y} \left\{ \boldsymbol{g}_i^{\mathrm{y}}\left(\epsilon^{x}\right) := h_{\mathbf{C}\mathbb{X}(\epsilon^{x})}\left({G_i^{\mathrm{y}}}^\top\right), \right. \end{aligned}$$

and define the vector-valued functions

$$\boldsymbol{g}(\epsilon^{x}, \epsilon^{\Delta x}) := \begin{bmatrix} \boldsymbol{g}^{\mathrm{x}}(\epsilon^{\Delta x}) \\ \boldsymbol{g}^{\mathrm{u}}(\epsilon^{\Delta x}) \\ \boldsymbol{g}^{\mathrm{y}}(\epsilon^{x}) \end{bmatrix}, \qquad \boldsymbol{f}(\epsilon^{x}) := \begin{bmatrix} \phi_{\mathrm{x}}\boldsymbol{f}^{\mathrm{x}}(\epsilon^{x}) \\ \phi_{\mathrm{u}} g^{\mathrm{u}} \\ g^{\mathrm{y}} \end{bmatrix}.$$

Then, we replace (4.18b), (4.18c) and (4.18d) by the functional inequality

$$\boldsymbol{g}(\epsilon^{x}, \epsilon^{\Delta x}) \leq \boldsymbol{f}(\epsilon^{x}). \tag{4.21}$$

Finally, in order to encode inclusion (4.18e), we assume to know the vertices of the set $\mathcal{Y}$, i.e.,

$$\{\mathrm{y}_{[i]}, i \in \mathbb{I}_1^{v\mathcal{Y}}\} = \mathrm{vert}(\mathcal{Y}),$$

using which we encode the inclusion as $(\epsilon^x, \epsilon) \in \Xi$, where

$$\Xi := \left\{(\epsilon^x, \epsilon) : \forall\, i \in \mathbb{I}_1^{v\mathcal{Y}}, \exists\{\mathbf{x}_{[i]} \in \mathbb{X}(\epsilon^x), \mathrm{b}_{[i]} \in \mathbb{B}(\epsilon)\} : \mathrm{y}_{[i]} = \mathbf{C}\mathbf{x}_{[i]} + \mathrm{b}_{[i]}\right\}.$$

Thus, using the aforementioned inclusion encodings, we write Problem (4.2) equivalently as

$$\begin{aligned}
\min_{\epsilon^x, \epsilon^{\Delta x}, \epsilon, z} \quad & \|\epsilon\|_1 && \text{(4.22a)}\\
\text{s.t.} \quad & \boldsymbol{c}(\epsilon^{\Delta x}) + \boldsymbol{d}(\epsilon^x) = \epsilon^{\Delta x}, && \text{(4.22b)}\\
& \boldsymbol{g}(\epsilon^x, \epsilon^{\Delta x}) \leq \boldsymbol{f}(\epsilon^x), && \text{(4.22c)}\\
& (\epsilon^x, \epsilon) \in \Xi, && \text{(4.22d)}\\
& \epsilon^x \geq \mathbf{0}, && \text{(4.22e)}
\end{aligned}$$

where $z := \{\mathbf{x}_{[i]}, \mathrm{b}_{[i]}, i \in \mathbb{I}_1^{v\mathcal{Y}}\}$ are the auxiliary variables required to encode the set Ξ. This problem is now in the form of Problem (3.30), such that we solve it using any of the techniques proposed in Chapter 3. In the numerical example, we solve the resulting optimization problem with the smoothening-based Interior Point solver presented in Section 3.4.

4.3.3 Integration with Controllers $\mathcal{C}_i$

Upon solving (4.22), we recover constraint sets $\mathcal{X}_i$ from the solution $\mathbb{X}(\epsilon^x)$. Then, we compute the sets $\mathcal{W}_i$ given by (4.8). For each $\mathcal{W}_i$, we compute RPI sets $\Delta\tilde{\mathcal{X}}_i(\mathcal{W}_i)$ by following the method presented in [111] to tightly approximate $\Delta\mathcal{X}_i(\mathcal{W}_i)$. By construction, we obtain

$$\Delta\mathcal{X}_i(\mathcal{W}_i) \subseteq \Delta\tilde{\mathcal{X}}_i(\mathcal{W}_i) \subseteq \Delta\mathbb{X}_i(\epsilon^{\Delta x, i})$$

for a tight enough $\Delta\tilde{\mathcal{X}}_i(\mathcal{W}_i)$. Using $\mathcal{X}_i$ and $\Delta\tilde{\mathcal{X}}_i(\mathcal{W}_i)$, we construct the optimization problems in (6.8) solved by $\mathcal{C}_i$. We use Proposition 4.1 to check the validity of a given initial state.

Remark 4.3. *One can directly use the RPI sets $\Delta\mathbb{X}_i(\epsilon^{\Delta x,i})$ in place of $\Delta\mathcal{X}_i(\mathcal{W}_i)$. However, this results in a smaller feasible region $\mathcal{X}_i^{N_{[i]}}$, and increases conservativeness of $\mathcal{C}_i$.* □

Remark 4.4. *The proposed formulation allows one to introduce specific conditions to be satisfied by the parameterization of the sets $\mathcal{X}$, e.g., symmetry constraints can be imposed; and the inclusion of a feasible region of the state-space in $\mathcal{X}$ can be imposed through the sufficiency conditions presented in [127].* □

Remark 4.5. *The computed sets $\mathcal{X}_i$ can be used to synthesize local controllers $\mathcal{C}_i$ using other methods, e.g. [26].* □

4.4 Numerical Example

We consider an LTI plant composed of three discrete-time double integrators given by

$$A_{[ii]} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad B_{[i]} = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, \qquad \forall\, i \in \mathbb{I}_1^3,$$

that are dynamically coupled through the matrix

$$\tilde{\mathbf{B}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0480 & -0.0060 & -0.0480 & -0.0600 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0360 & -0.0240 & 0 & 0 & 0.0360 & 0.0240 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0360 & 0.0360 & 0.0120 & 0.0600 & 0 & 0 \end{bmatrix},$$

i.e., $\mathbf{A} = \mathrm{diag}(A_{[11]}, A_{[22]}, A_{[33]}) + \tilde{\mathbf{B}}$, and $\mathbf{B} = \mathrm{diag}(B_{[1]}, B_{[2]}, B_{[3]})$.

Each integrator is subject to local constraints

$$-\mathbf{1} \leq x_{[i]} \leq \mathbf{1}, \qquad -1 \leq u_{[i]} \leq 1, \qquad \forall\, i \in \mathbb{I}_1^3.$$

and the overall system is subject to coupled constraints

$$x_{[1]1} - x_{[2]1} \leq 0.99, \qquad x_{[2]1} - x_{[3]1} \leq 0.99,$$

which represent collision avoidance specifications on the integrators.

In order to design local MPC controllers $\mathcal{C}_i$ for each subsystem i, we first equip each system with an LQR feedback gain, corresponding to

$$Q_{[i]} = \mathbf{I},\ \forall\, i \in \mathbb{I}_1^3, \qquad\qquad R_{[1]} = 1,\ R_{[2]} = 5,\ R_{[3]} = 10.$$

Then, we parameterize each state-constraint set $\mathcal{X}_i$ as the uniform polytope $\mathbb{X}_i(\epsilon^{x,i})$ with $m_X^1 = 8$, $m_X^2 = 12$ and $m_X^3 = 10$. We also parameterize each RPI set $\Delta\mathbb{X}_i(\epsilon^{\Delta x,i})$ in the same way as $\mathbb{X}_i(\epsilon^{x,i})$. Finally, we select $\phi_x = 0.3$ and $\phi_u = 0.5$. By parameterizing $\Delta\mathbb{X}_i(\epsilon^{\Delta x,i})$ with the same set of normal vectors as $\mathbb{X}_i(\epsilon^{x,i})$ for each $i \in \mathbb{I}_1^3$, we simplfy the condition

$$\boldsymbol{g}^{\mathrm{x}}\left(\epsilon^{\Delta x}\right) \le \phi_{\mathrm{x}} \boldsymbol{f}^{\mathrm{x}}(\epsilon^{x})$$

in (4.21) to the linear inequality $\epsilon^{\Delta x} \le \phi_{\mathrm{x}}\epsilon^{x}$. Based on these parameterizations, we formulate Problem (4.22) as Problem (3.45), that we then solve using the smoothening-based interior-point solver presented in Section 3.4.

The results obtained are plotted in Figure 4. Using the sets $\mathcal{X}_i$, we compute tighter RPI sets $\Delta\tilde{\mathcal{X}}_i(\mathcal{W}_i) \subseteq \Delta\mathbb{X}_i(\epsilon^{\Delta x,i})$ using the method in [111]. The resulting sets $\Delta\tilde{\mathcal{X}}_i(\mathcal{W}_i)$ approximate the mRPI sets $\Delta\mathcal{X}_i(\mathcal{W}_i)$ as

$$\Delta\mathcal{X}_i(\mathcal{W}_i) \subseteq \Delta\tilde{\mathcal{X}}_i(\mathcal{W}_i) \subseteq \Delta\mathcal{X}_i(\mathcal{W}_i) \oplus 10^{-4}\mathcal{B}_\infty^{n_x}.$$

Using this set, we perform constraint tightening for the MPC controllers as

$$\mathcal{X}_i \ominus \tilde{\mathcal{X}}_i(\mathcal{W}_i), \qquad\qquad U_i \ominus K_{[i]}\tilde{\mathcal{X}}_i(\mathcal{W}_i).$$

Finally, inside the tightened constraint set we use the Maximal Positive Invariant set, computed as in [44], as the terminal set $\mathcal{X}_i^{\mathrm{terminal}}$. The results of synthesizing the tube-MPC controllers $\mathcal{C}_i$ using these sets is plotted in Figure 4. For comparison, we synthesize a standard *centralized* MPC controller for System (4.1), and enforce the constraints (4.2) inside this MPC controller. This controller solver the following QP at each timestep t, that is parameterized by the current state measurement $\mathbf{x}(t)$:

$$
\begin{aligned}
\min_{\mathbf{v}} \quad & \sum_{s=t}^{t+N-1} \|\mathbf{x}(s)\|_{\mathbf{Q}}^2 + \|\mathbf{u}(s)\|_{\mathbf{R}}^2 + \|\mathbf{x}(N)\|_{\mathbf{P}}^2 && && \text{(4.23a)}\\
\text{s.t.} \quad & \mathbf{x}(s+1) = \mathbf{A}\mathbf{x}(s) + \mathbf{B}\mathbf{u}(s), && s \in \mathbb{I}_t^{t+N-1}, && \text{(4.23b)}\\
& \mathbf{C}\mathbf{x}(s) \in \mathcal{Y},\ \mathbf{u}(s) \in U, && s \in \mathbb{I}_t^{t+N-1}, && \text{(4.23c)}\\
& \mathbf{x}(t+N) \in \mathbf{X}^{\mathrm{t}}, && && \text{(4.23d)}
\end{aligned}
$$

with variables $\mathbf{v} := \{\mathbf{x}(t+1), \cdots, \mathbf{x}(t+N), \mathbf{u}(t), \cdots, \mathbf{u}(t+N-1)\}$, and the cost matrices $\mathbf{Q} = \mathrm{diag}(Q_{[1]}, \cdots, Q_{[M]})$ and $\mathbf{R} = \mathrm{diag}(R_{[1]}, \cdots, R_{[M]})$. We select $\mathbf{P}$ as the minimum trace matrix that satisfies the matrix inequality $\mathbf{A}^{\mathrm{K}^\top}\mathbf{P}\mathbf{A}_{\mathbf{K}} - \mathbf{P} \preceq -(\mathbf{Q} + \mathbf{K}^\top\mathbf{R}\mathbf{K})$, computed using well-known techniques [21]. For the current example, we obtain

$$
\mathbf{P} = \begin{bmatrix}
2.4306 & 1.1516 & -0.0615 & 0.1401 & 0.0869 & 0.3060\\
1.1516 & 2.6234 & -0.0273 & -0.0398 & 0.0153 & 0.2239\\
-0.0615 & -0.0273 & 2.9340 & 2.2977 & 0.0007 & -0.0000\\
0.1401 & -0.0398 & 2.2977 & 6.0911 & 0.1748 & 0.4379\\
0.0869 & 0.0153 & 0.0007 & 0.1748 & 3.3027 & 3.2482\\
0.3060 & 0.2239 & -0.0000 & 0.4379 & 3.2482 & 9.4144
\end{bmatrix}.
$$

Finally, we select $\mathbf{X}^{\mathrm{t}}$ as the Maximal Positive Invariant set for the closed-loop system $\mathbf{x}(t+1) = \mathbf{A}_{\mathbf{K}}\mathbf{x}(t)$ satisfying the system constraints as $\mathbf{X}^{\mathrm{t}} \subseteq \{\mathbf{x} : \mathbf{C}\mathbf{x} \in \mathcal{Y},\ \mathbf{K}\mathbf{x} \in U\}$. We observe that, while the closed-loop performance deteriorates as expected in the DeMPC case, we achieve our objectives for controller design, i.e., we stabilize the closed-loop system while satisfying constraints in a decentralized manner.

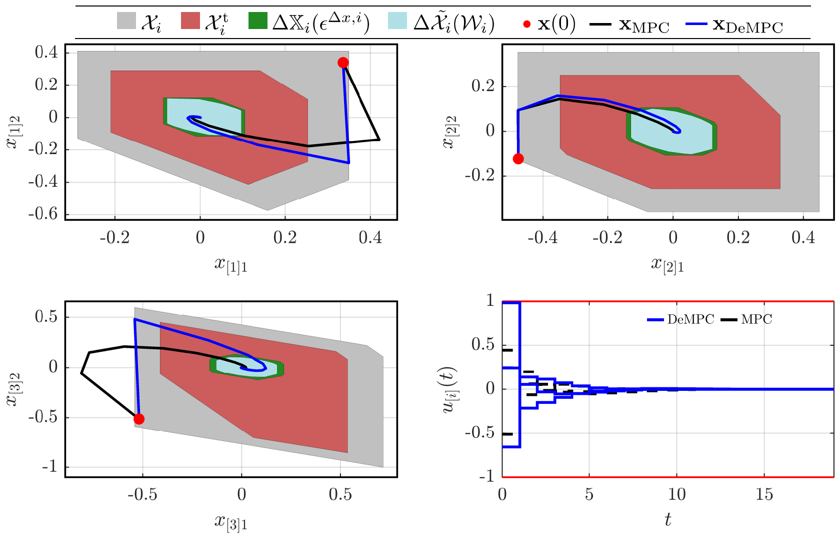


Figure 4: Computed sets and simulation results in state space with MPC controllers formulated with horizon $N = 5$. Red dots indicate initial states $x_{[i]}(0)$. DeMPC controllers $\mathcal{C}_i$ restrict the system state as $x_{[i]} \in \mathcal{X}_i$ to satisfy the constraints in (4.2), while centralized MPC does not require this restriction.

Chapter 5

Computation of Safe Disturbance Sets using Implicit RPI sets

In this chapter, we revisit Problem (3.18), i.e., the problem of computing a disturbance set such that the corresponding set of reachable outputs approximates an assigned set, while remaining constrained in it. Unlike the approach used in Chapter 3 however, we solve the problem using an implicit RPI set parameterization. For convenience, we recall that Problem (3.18) is written as

$$\min_{\mathcal{W}} \ \mathrm{d}_{\mathcal{Y}}(\mathcal{S}(l,\mathcal{W})) \tag{5.1a}$$

$$\text{s.t. } \mathcal{Y}_{\mathrm{RPI}}(\mathcal{W}) \subseteq \mathcal{Y}, \tag{5.1b}$$

$$\mathbf{0} \in \mathcal{W}. \tag{5.1c}$$

The output constraint set $\mathcal{Y}$ is given as the polytope

$$\mathcal{Y} = \{y : G_i y \leq g_i, \ \forall\, i \in \mathbb{I}_1^{m_{\mathcal{Y}}}\} = \{y : Gy \leq g\},$$

and the distance function defining the objective is given for some user-specified index $l > 0$ as

$$\mathrm{d}_{\mathcal{Y}}(\mathcal{S}(l,\mathcal{W})) := \min_{\epsilon}\{\|\epsilon\|_1 : \mathcal{Y} \subseteq \mathcal{S}(l,\mathcal{W}) \oplus \mathbb{B}(\epsilon)\}$$

using the set $\mathbb{B}(\epsilon) = \{y : Hy \leq \epsilon\}$, in which normal vectors $\{H_i^\top, i \in \mathbb{I}_1^{n_B}\}$ are specified a priori by the user, and the set $\mathcal{S}(l, \mathcal{W})$ is the l-step output reachable set.

In Implicit RPI (IRPI) approach that we present in this chapter tackles directly the following main drawbacks of the Explicit RPI (ERPI) approach presented in Chapter 3:

1. Given matrix A of system (7.1), it is not well understood how to select a matrix E satisfying Assumption 3.2, i.e., it is not clear how to parameterize an RPI set $\mathbb{X}(\epsilon^x)$.

2. The approximation error of an RPI set $\mathbb{X}(\epsilon^x)$ with respect to the mRPI set $\mathcal{X}_{\mathrm{m}}(\mathbb{W}(\epsilon^w))$ cannot be specified a priori.

3. The disturbance set parameterization $\mathbb{W}(\epsilon^w)$ is such that the resulting set of feasible disturbance sets satisfying the output inclusion condition $C\mathbb{X}(\epsilon^x(\epsilon^w)) \oplus D\mathbb{W}(\epsilon^w) \subseteq \mathcal{Y}$ is nonconvex and nonsmooth, thus requiring the implementation of a specialized solver.

We tackle Issues 1 and 2 by directly using IRPI sets. In particular, the RPI set parameterization is such that we need not fix a representation for the set a priori, but rather only specify an approximation error with respect to the mRPI set. In order to tackle Issue 3, we present a novel disturbance set parameterization that will result in the set of feasible disturbance sets being a polyhedron, from which any point can be selected by using a simple Linear Programing (LP) solver. However, we work under the following slightly restrictive assumption on the set $\mathcal{Y}$ that was not required in Chapter 3.

Assumption 5.1. *The output constraint set $\mathcal{Y}$ contains the origin in its nonempty interior, such that $g > \mathbf{0}$.* □

We refer to our approach as using *implicit* RPI sets, since unlike in Chapter 3 we do not explicitly compute a representation of the RPI set, but rather use an implicit representation parameterized by the disturbance set. In the development of our methods, we use some notation and basic results regarding set operations that we describe next for convenience.

Preliminaries and notation

Given compact set $\mathcal{S} \subset \mathbb{R}^n$, and matrices $\mathbf{T} \in \mathbb{R}^{l\times n}$, $\mathbf{M} \in \mathbb{R}^{q\times n}$, we denote by $h_{\mathbf{T}\mathcal{S}}(\mathbf{M})$ the q-dimensional vector with elements $h_{\mathbf{T}\mathcal{S}}(\mathbf{M}_i^\top)$, i.e.,

$$h_{\mathbf{T}\mathcal{S}}(\mathbf{M}) := [h_{\mathbf{T}\mathcal{S}}(\mathbf{M}_1^\top) \ \cdots \ h_{\mathbf{T}\mathcal{S}}(\mathbf{M}_q^\top)]^\top.$$

Given $\mathcal{S} = \bar{z} \oplus \{z : -\epsilon^z \leq z \leq \epsilon^z\}$ shaped as a box with $\bar{z}, \epsilon^z \in \mathbb{R}^n$ and any vector $\mathbf{p} \in \mathbb{R}^{l\times 1}$, the support function is given by [16, Chapter 6]:

$$h_{\mathbf{T}\mathcal{S}}(\mathbf{p}) = \mathbf{p}^\top \mathbf{T}\bar{z} + |\mathbf{p}^\top \mathbf{T}|\epsilon^z. \tag{5.2}$$

Proposition 5.1. *[131] Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subset \mathbb{R}^n$ be any compact and convex sets containing the origin, $M \in \mathbb{R}^{m\times n}$ be any matrix of adequate dimension, and $\alpha \geq \beta > 0$ be any scalars.*
(a) $\mathcal{X} \subseteq \mathcal{Y} \Rightarrow M\mathcal{X} \subseteq M\mathcal{Y}$; (b) $\mathcal{X} \subseteq \mathcal{Y} \Leftrightarrow \mathcal{X} \oplus \mathcal{Z} \subseteq \mathcal{Y} \oplus \mathcal{Z}$;
(c) $\alpha\mathcal{X} \oplus \beta\mathcal{X} = (\alpha+\beta)\mathcal{X}$; (d) $\mathcal{X} \subseteq \mathcal{X} \oplus \mathcal{Y}$;
(e) If $\mathcal{X} = \mathrm{ConvHull}(x_{[i]}, i \in \mathbb{I}_1^v)$, then $\mathcal{X} \subseteq \mathcal{Y} \oplus \mathcal{Z}$ holds if and only if for each $i \in \mathbb{I}_1^v$, there exist $y_{[i]} \in \mathcal{Y}$ and $z_{[i]} \in \mathcal{Z}$ such that $x_{[i]} = y_{[i]} + z_{[i]}$.

5.1 Implicit RPI set parametrization

In this section, we approximate Problem (3.7) using arbitrarily tight RPI approximations of the mRPI set $\mathcal{X}_{\mathrm{m}}(\mathcal{W})$. To that end, we will rely on the following definition.

Definition: Given a disturbance set $\mathcal{W}$, for a given $\mu > 0$, an RPI set $\mathcal{X}_\mu(\mathcal{W})$ is a μ-RPI set for System (3.1a) if it satisfies the inclusions

$$\mathcal{X}_{\mathrm{m}}(\mathcal{W}) \ \subseteq \ \mathcal{X}_\mu(\mathcal{W}) \ \subseteq \ \mathcal{X}_{\mathrm{m}}(\mathcal{W}) \oplus \mu\mathcal{B}_\infty^{n_x}. \qquad \square \tag{5.3}$$

Essentially, a μ-RPI set approximates the mRPI set with approximation error μ. Using a μ-RPI set $\mathcal{X}_\mu(\mathcal{W})$ as the RPI set $\mathcal{X}_{\mathrm{RPI}}(\mathcal{W})$, we define the corresponding set of outputs as

$$\mathcal{Y}_\mu(\mathcal{W}) := C\mathcal{X}_\mu(\mathcal{W}) \oplus D\mathcal{W},$$

which is used as the set $\mathcal{Y}_{\mathrm{RPI}}$ in Problem (3.18). Using this set, we write Problem (3.18) as

$$\min_{\mathcal{W}} \; \mathrm{d}_{\mathcal{Y}}(\mathcal{S}(l, \mathcal{W})) \tag{5.4a}$$

$$\text{s.t. } \mathcal{Y}_{\mu}(\mathcal{W}) \subseteq \mathcal{Y}, \tag{5.4b}$$

$$\mathbf{0} \in \mathcal{W}. \tag{5.4c}$$

The rationale for formulating Problem (5.4) is the following. From the inclusions in (5.3) and basic properties of the Minkowski sum, the inclusions

$$C\mathcal{X}_{\mathrm{m}}(\mathcal{W}) \oplus D\mathcal{W} \subseteq C\mathcal{X}_{\mu}(\mathcal{W}) \oplus D\mathcal{W} \subseteq C\mathcal{X}_{\mathrm{m}}(\mathcal{W}) \oplus D\mathcal{W} \oplus \mu C\mathcal{B}_{\infty}^{n_x}. \tag{5.5}$$

Then defining the sets of disturbance sets

$$\begin{aligned}
\mathcal{P}_1 &:= \{\mathcal{W} \;:\; \mathbf{0} \in \mathcal{W},\; C\mathcal{X}_{\mathrm{m}}(\mathcal{W}) \oplus D\mathcal{W} \oplus \mu C\mathcal{B}_{\infty}^{n_x} \subseteq \mathcal{Y}\}, \\
\mathcal{P}_2 &:= \{\mathcal{W} \;:\; \mathbf{0} \in \mathcal{W},\; C\mathcal{X}_{\mu}(\mathcal{W}) \oplus D\mathcal{W} \subseteq \mathcal{Y}\}, \\
\mathcal{P}_3 &:= \{\mathcal{W} \;:\; \mathbf{0} \in \mathcal{W},\; C\mathcal{X}_{\mathrm{m}}(\mathcal{W}) \oplus D\mathcal{W} \subseteq \mathcal{Y}\}.
\end{aligned}$$

Observe that $\mathcal{P}_2$ is the feasible set of Problem (5.4), and $\mathcal{P}_3$ is the feasible set of Problem (3.7). From the inclusions in (5.5), we observe that the feasible sets satisfy

$$\mathcal{P}_1 \subseteq \mathcal{P}_2 \subseteq \mathcal{P}_3. \tag{5.6}$$

Moreover, for any μ_1, μ_2 such that $0 < \mu_1 < \mu_2$, the inclusions

$$C\mathcal{X}_{\mathrm{m}}(\mathcal{W}) \oplus D\mathcal{W} \oplus \mu_1 C\mathcal{B}_{\infty}^{n_x} \subseteq C\mathcal{X}_{\mathrm{m}}(\mathcal{W}) \oplus D\mathcal{W} \oplus \mu_2 C\mathcal{B}_{\infty}^{n_x}$$

follow, which implies that as the value of the RPI set approximation error μ reduces, the set $\mathcal{P}_1$ goes closer to $\mathcal{P}_3$. For any $\mu > 0$, since the inclusions in (5.6) always hold, it implies that as μ reduces, the feasible set of Problem (5.4) approximates better (from inside) the feasible set of Problem (3.7), thus leading to reduced conservativeness in Problem (5.4) in comparison to Problem (3.7).

Thus, in order to make it possible to specify the desired approximation error a priori, we will exploit the approximation property (5.3) of μ-RPI sets to provide strong guarantees when approximating the mRPI set $\mathcal{X}_{\mathrm{m}}(\mathcal{W})$ in the formulation of Problem (3.7). To this end, we recall next the following result from [111] that can be used to compute a μ-RPI set, provided that the disturbance set $\mathcal{W}$ is known. Since in our setting that is not the case, we will have to develop some additional theoretical results in order to be able to exploit it.

Lemma 5.1. *[111, Section III-B] Suppose that Assumption 3.1 holds. For some index $s > 0$, scalars $\alpha \in [0, 1)$, $\lambda \in [0, 1]$, and disturbance set $\mathcal{W}$ such that $\mathbf{0} \in \mathcal{W}$, if*

$$A^s(B\mathcal{W} \oplus \lambda\mathcal{B}_\infty^{n_x}) \subseteq \alpha(B\mathcal{W} \oplus \lambda\mathcal{B}_\infty^{n_x}) \tag{5.7}$$

holds, then the parametrized set

$$\mathcal{R}(s, \alpha, \lambda, \mathcal{W}) := (1-\alpha)^{-1} \bigoplus_{t=0}^{s-1} A^t(B\mathcal{W} \oplus \lambda\mathcal{B}_\infty^{n_x}) \tag{5.8}$$

is RPI for System (3.1a) *with persistent disturbances $w \in \mathcal{W}$. Moreover, if for some scalar $\mu > 0$, the inclusion*

$$(1-\alpha)^{-1} \bigoplus_{t=0}^{s-1} A^t(\alpha B\mathcal{W} \oplus \lambda\mathcal{B}_\infty^{n_x}) \subseteq \mu\mathcal{B}_\infty^{n_x} \tag{5.9}$$

holds, then $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$ is a μ-RPI set. $\square$

We briefly recall the rationale behind Lemma 5.1. For any compact and convex disturbance set $\mathcal{W}$ containing the origin and for any index $s > 0$, the inclusion

$$\mathcal{X}(s, \mathcal{W}) := \bigoplus_{t=0}^{s-1} A^t B\mathcal{W} \subseteq \bigoplus_{t=0}^{\infty} A^t B\mathcal{W} \overset{(3.10)}{=} \mathcal{X}_\mathrm{m}(\mathcal{W})$$

holds. Then, we know from [112, Theorem 1] that if, for some index $s > 0$ and a scalar $\alpha \in [0, 1)$, the inclusion

$$A^s B\mathcal{W} \subseteq \alpha B\mathcal{W} \tag{5.10}$$

holds, the scaled set $(1-\alpha)^{-1}\mathcal{X}(s, \mathcal{W})$ is RPI for System (3.1a) with disturbance set $\mathcal{W}$. However, as noted in [111], there might not exist some parameters (s, α) satisfying inclusion (5.10) unless the origin is included in the interior of the set $B\mathcal{W}$. This can occur, for example, if the rank of matrix B is smaller than n_x, thus posing a structural restriction on the applicability of [112, Theorem 1]. To overcome this limitation, a modification proposed in [111] involves perturbing the disturbance set with some $\lambda \in [0, 1]$ as

$$\tilde{\mathcal{W}}(\lambda) := B\mathcal{W} \oplus \lambda \mathcal{B}_{\infty}^{n_x}.$$

Then, since the interior of the set $\tilde{\mathcal{W}}(\lambda)$ is nonempty for every $\lambda \in (0, 1]$, there always exists some (s, α) satisfying the inclusion in (5.7), i.e.,

$$A^s \tilde{\mathcal{W}}(\lambda) \subseteq \alpha \tilde{\mathcal{W}}(\lambda). \tag{5.11}$$

From [112, Theorem 1], it then follows that the set $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$ defined in (5.8) is RPI for the modified system

$$x(t+1) = Ax(t) + \tilde{w}(t) \tag{5.12}$$

with disturbance set $\tilde{\mathcal{W}}(\lambda)$. Finally, the result in Lemma 5.1 follows from [111, Lemma 2], which states that every RPI set of System (5.12) with disturbance set $\tilde{\mathcal{W}}(\lambda)$ is an RPI set for System (3.1a) with disturbance set $\mathcal{W}$. For the reasoning behind inclusion (5.9) that leads to $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$ being a μ-RPI set, the reader is referred to [111, Theorem 3].

Lemma 5.1 provides an important result that enables us to construct an RPI set $\mathcal{X}_{\mathrm{RPI}}(\mathcal{W})$ parameterized as the μ-RPI set $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$. Using this set, we define the corresponding output set as

$$\mathcal{O}(s, \alpha, \lambda, \mathcal{W}) := C\mathcal{R}(s, \alpha, \lambda, \mathcal{W}) \oplus D\mathcal{W} \;\supseteq\; \mathcal{Y}_{\mathrm{m}}(\mathcal{W}), \tag{5.13}$$

that represents the set $\mathcal{Y}_{\mathrm{RPI}}(\mathcal{W})$ in Problem (3.18). Hence, we write Problem (3.18) for the μ-RPI set based parameterization as

$$\min_{\mathcal{W}} \; \mathrm{d}_{\mathcal{Y}}(\mathcal{S}(l, \mathcal{W})) \tag{5.14a}$$

$$\text{s.t. } \mathcal{O}(s, \alpha, \lambda, \mathcal{W}) \subseteq \mathcal{Y}, \tag{5.14b}$$

$$\mathbf{0} \in \mathcal{W}, \tag{5.14c}$$

in which we recall that the objective is defined using the l-step reachable set of outputs

$$\mathcal{S}(l, \mathcal{W}) := \bigoplus_{t=0}^{l-1} CA^t B\mathcal{W} \oplus D\mathcal{W}. \tag{5.15}$$

Before we tackle Problem (5.14), however, we must ensure that the set $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$ formulating $\mathcal{O}(s, \alpha, \lambda, \mathcal{W})$ in (5.13) is μ-RPI. According to Lemma 5.1, this is guaranteed if the parameters (s, α, λ) are such that inclusions (5.7) and (5.9) hold at the optimal solution. In order to guarantee that these inclusions hold, the following three approaches can be considered.

(1) *Iterative verification:* Problem (5.14) is solved with some arbitrary parameters (s, α, λ). At the optimizer, inclusions (5.7) and (5.9) are checked with the chosen (s, α, λ). If they are verified, then $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$ is a μ-RPI set and the procedure terminates. Else, the parameters (s, α, λ) are recomputed at the optimizer to verify (5.7) and (5.9), and the procedure is repeated with the updated (s, α, λ).

(2) *Optimizing over* (s, α, λ)*:* The parameters (s, α, λ) are appended as optimization variables, and inclusions (5.7) and (5.9) are appended as constraints in Problem (5.14).

(3) *Preselecting* (s, α, λ)*:* Problem (5.14) is solved with (s, α, λ) chosen a priori such that inclusions (5.7) and (5.9) hold for all feasible disturbance sets $\mathcal{W}$.

While Approach (1) is viable, it requires a strategy to update the parameters (s, α, λ) after each iteration in order to guarantee convergence. The development of such strategies is a subject of future research. Approach (2), while presenting a more attractive alternative since the optimizer is guaranteed to verify inclusions (5.7) and (5.9) by construction, results in a computationally intractable optimization problem. Further study of this approach too will be the subject of future research. In this chapter, we adopt Approach (3), i.e., we select parameters (s, α, λ) such that inclusions (5.7) and (5.9) are verified by all disturbance sets $\mathcal{W}$ feasible for Problem (5.14). In the next subsection, we present this approach.

5.1.1 Preselecting RPI set parameters (s, α, λ)

By definition, the requirement on parameters (s, α, λ) to adopt Approach (3) is that inclusions (5.7) and (5.9) must be verified for all feasible disturbance sets $\mathcal{W}$ of Problem (5.14). In this chapter, we relax this requirement to the following.

- ***Requirement (a)***: *For some user-specified $\gamma > 0$, parameters (s, α, λ) are such that inclusions* (5.7) *and* (5.9) *hold for all disturbance sets*

$$\mathcal{W} \in \boldsymbol{\mathcal{L}}_\gamma := \{\mathcal{W} \, : \, \mathbf{0} \in \mathcal{W}, \; B\mathcal{W} \subseteq \gamma \mathcal{B}_\infty^{n_x}\}. \tag{5.16}$$

As we will show in the sequel, the restriction $B\mathcal{W} \subseteq \gamma \mathcal{B}_\infty^{n_x}$ over the disturbance sets permits us to characterize a set of parameters (s, α, λ) satisfying the requirement. The formulation of Requirement (a) motivates us to impose the additional constraint $B\mathcal{W} \subseteq \gamma \mathcal{B}_\infty^{n_x}$ in Problem (5.14) to obtain

$$\begin{aligned}
\min_{\mathcal{W}} \;\; & \mathrm{d}_{\mathcal{Y}}(\mathcal{S}(l, \mathcal{W})) && (5.17a)\\
\text{s.t.} \;\; & \mathcal{O}(s, \alpha, \lambda, \mathcal{W}) \subseteq \mathcal{Y}, && (5.17b)\\
& B\mathcal{W} \subseteq \gamma \mathcal{B}_\infty^{n_x}, && (5.17c)\\
& \mathbf{0} \in \mathcal{W}, && (5.17d)
\end{aligned}$$

such that if parameters (s, α, λ) satisfy Requirement (a), the set $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$ formulating $\mathcal{O}(s, \alpha, \lambda, \mathcal{W})$ is μ-RPI for every feasible disturbance set $\mathcal{W}$.

We impose the following additional requirement on the parameters (s, α, λ) to guarantee feasibility of Problem (5.17).

- ***Requirement (b)***: *Parameters (s, α, λ) are such that there exists a disturbance set $\mathcal{W} \in \boldsymbol{\mathcal{L}}_\gamma$ satisfying constraint* (5.17b).

In the following result, we translate Requirements (a) and (b) into sufficient conditions on the parameters (s, α, λ).

Lemma 5.2. *If the parameters $s > 0$, $\alpha \in [0, 1)$ and $\lambda \in [0, 1]$ verify the inclusion*

$$\lambda (1 - \alpha)^{-1} \bigoplus_{t=0}^{s-1} C A^t \mathcal{B}_\infty^{n_x} \subseteq \mathcal{Y}, \tag{5.18}$$

then there exists some disturbance set $\mathcal{W} \in \boldsymbol{\mathcal{L}}_\gamma$ for any $\gamma > 0$ such that the inclusion $\mathcal{O}(s, \alpha, \lambda, \mathcal{W}) \subseteq \mathcal{Y}$ holds. Moreover, if for some user-specified scalars

$\mu, \gamma > 0$, *the inclusions*

$$A^s(\gamma + \lambda)\mathcal{B}_\infty^{n_x} \subseteq \alpha\lambda\mathcal{B}_\infty^{n_x}, \tag{5.19}$$

$$(1-\alpha)^{-1}\bigoplus_{t=0}^{s-1} A^t(\alpha\gamma + \lambda)\mathcal{B}_\infty^{n_x} \subseteq \mu\mathcal{B}_\infty^{n_x}, \tag{5.20}$$

hold, then $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$ *is a* μ*-RPI set corresponding to any disturbance set* $\mathcal{W} \in \boldsymbol{\mathcal{L}}_\gamma$. □

Proof. Note that, by (5.13), we have

$$\mathcal{O}(s, \alpha, \lambda, \mathcal{W}) = (1-\alpha)^{-1}\bigoplus_{t=0}^{s-1} CA^t(B\mathcal{W} \oplus \lambda\mathcal{B}_\infty^{n_x}) \oplus D\mathcal{W}.$$

Consequently, if inclusion (5.18) holds, then the inclusion $\mathcal{O}(s, \alpha, \lambda, \mathcal{W}) \subseteq \mathcal{Y}$ is satisfied by $\mathcal{W} = \{\mathbf{0}\} \in \boldsymbol{\mathcal{L}}_\gamma$, concluding the proof of the first claim.

For the second claim, if inclusion (5.19) holds, then for any $\mathcal{W} \in \boldsymbol{\mathcal{L}}_\gamma$, the inclusions

$$\begin{aligned} A^s(B\mathcal{W} \oplus \lambda\mathcal{B}_\infty^{n_x}) &\subseteq A^s(\gamma + \lambda)\mathcal{B}_\infty^{n_x} && \rightarrow \text{Since } B\mathcal{W} \subseteq \gamma\mathcal{B}_\infty^{n_x} \\ &\subseteq \alpha\lambda\mathcal{B}_\infty^{n_x} && \rightarrow \text{Directly from (5.19)} \\ &\subseteq \alpha(B\mathcal{W} \oplus \lambda\mathcal{B}_\infty^{n_x}) && \rightarrow \text{Since } \mathbf{0} \in \alpha B\mathcal{W} \end{aligned}$$

follow from basic properties of set operations in Proposition 5.1, such that inclusion (5.7) holds. Similarly, if inclusion (5.20) holds, then for every disturbance set $\mathcal{W} \in \boldsymbol{\mathcal{L}}_\gamma$, the inclusions

$$\begin{aligned} &(1-\alpha)^{-1}\bigoplus_{t=0}^{s-1} A^t(\alpha B\mathcal{W} \oplus \lambda\mathcal{B}_\infty^{n_x}) \\ &\subseteq (1-\alpha)^{-1}\bigoplus_{t=0}^{s-1} A^t(\alpha\gamma + \lambda)\mathcal{B}_\infty^{n_x} && \rightarrow \text{Since } B\mathcal{W} \subseteq \gamma\mathcal{B}_\infty^{n_x} \\ &\subseteq \mu\mathcal{B}_\infty^{n_x} && \rightarrow \text{Directly from (5.20)} \end{aligned}$$

follow from basic properties of set operations in Proposition 5.1, such that inclusion (5.9) holds. Consequently, by Lemma 5.1, $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$ is a μ-RPI set. □

As per Lemma 5.2, parameters (s, α, λ) that verify inclusions (5.19)-(5.20) satisfy Requirement (a), and those that verify inclusion (5.18) satisfy of Requirement (b). Hence, if parameters (s, α, λ) verifying inclusions (5.18)-(5.20) are used to formulate Problem (5.17), then the problem is guaranteed to be feasible, and for every feasible disturbance set $\mathcal{W}$, the set $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$ formulating $\mathcal{O}(s, \alpha, \lambda, \mathcal{W})$ is μ-RPI. In the next section, we present an approach to compute such parameters (s, α, λ).

5.1.2 Computing RPI set parameters (s, α, λ)

In this subsection, we present an algorithm to compute parameters (s, α, λ) satisfying inclusions (5.18)-(5.20). To this end, we define the set of all parameters (s, α, λ) satisfying these inclusions for some user-specified scalars $\gamma, \mu > 0$ as

$$\Lambda_{\gamma,\mu} := \{(s, \alpha, \lambda) : s > 0, \alpha \in [0, 1), \lambda \in [0, 1], \text{ (5.18)} - \text{(5.20)}\}.$$

Then, suitable parameters (s, α, λ) can be selected by solving

$$\min_{s,\alpha,\lambda} \quad s \quad \text{s.t.} \quad (s, \alpha, \lambda) \in \Lambda_{\gamma,\mu}. \tag{5.21}$$

In Problem (5.21), we compute the smallest index s satisfying inclusions (5.18)-(5.20). This formulation is motivated by the fact that the index s characterizes the number of Minkowski sums defining the set $\mathcal{O}(s, \alpha, \lambda, \mathcal{W})$ in Constraint (5.17b), and a smaller number of Minkowski sums are desirable for reduced computational complexity. Unfortunately, solving Problem (5.21) directly is not viable since s is a discrete variable. Thus, we propose to instead follow an iterative procedure to solve Problem (5.21) that involves incrementing s, and searching for feasible parameters (α, λ).

For some $s > 0$, we define the set of parameters (α, λ) as

$$\boldsymbol{\mathcal{L}}_{\gamma,\mu}(s) := \{(\alpha, \lambda) \ : \ (s, \alpha, \lambda) \in \Lambda_{\gamma,\mu}\}, \tag{5.22}$$

based on which we define the optimization problem

$$\mathbb{H}(s) \begin{cases} \max\limits_{\alpha,\lambda} & \alpha + \lambda \\ \text{s.t.} & (\alpha, \lambda) \in \boldsymbol{\mathcal{L}}_{\gamma,\mu}(s). \end{cases} \tag{5.23}$$

Using this optimization problem, we define the iterative procedure to solve Problem (5.21) in Algorithm 3.

Algorithm 3: Solving Problem (5.21)

Result: Return (s, α, λ);
Input: Matrices A, C, G, vector g, scalars $\gamma, \mu > 0$;
Initialize: $s = 0,\ conv = 0$;
while $conv = 0$ **do**
  1. $s \leftarrow s + 1$;
  2. Solve $\mathbb{H}(s)$ for (α, λ);
  if $\mathbb{H}(s)$ *is feasible* **then**
    $conv \leftarrow 1$
  end
end

In the formulation of $\mathbb{H}(s)$, we maximize $\alpha + \lambda$ for numerical stability. However, since we aim for feasibility, any objective can be used. We now show that $\mathbb{H}(s)$ can be implemented as a Second-Order Cone Program (SOCP) [20, Sec 4.4.2]. Note that since $\alpha \in [0, 1)$ and $\lambda \in [0, 1]$, $\mathbb{H}(s)$ is bounded if feasible.

Implementation of $\mathbb{H}(s)$

In order to implement $\mathbb{H}(s)$, we need to encode inclusions (5.18)-(5.20) formulating the constraint set of the problem. We do so using support functions, as we now elaborate. Recalling the definition of the output constraint set $\mathcal{Y} = \{y : Gy \leq g\}$ with $G \in \mathbb{R}^{m_\mathcal{Y} \times n_y}$ from (3.2), and denoting $\tilde{\mathbf{I}}_{n_x} := [\mathbf{I}_{n_x} \quad -\mathbf{I}_{n_x}]^\top$, we note from properties of support functions that

$$(5.18) \iff \frac{\lambda}{1-\alpha} \sum_{t=0}^{s-1} h_{CA^t \mathcal{B}_\infty^{n_x}}(G) \leq g, \tag{5.24a}$$

$$(5.19) \iff (\gamma + \lambda) h_{A^s \mathcal{B}_\infty^{n_x}}(\tilde{\mathbf{I}}_{n_x}) \leq \alpha\lambda\mathbf{1}, \tag{5.24b}$$

$$(5.20) \iff \frac{\alpha\gamma + \lambda}{1-\alpha} \sum_{t=0}^{s-1} h_{A^t \mathcal{B}_\infty^{n_x}}(\tilde{\mathbf{I}}_{n_x}) \leq \mu\mathbf{1}. \tag{5.24c}$$

Exploiting Equation (5.2), we then define the constants

$$L^{[s]} := \sum_{t=0}^{s-1} h_{CA^t\mathcal{B}_\infty^{n_x}}(G) = \sum_{t=0}^{s-1} |GCA^t|\mathbf{1}, \tag{5.25a}$$

$$\theta^{[s]} := \min_{i\in\mathbb{I}_1^{m_\mathcal{Y}}} \left\{ \frac{g_i}{L_i^{[s]}} \right\}, \tag{5.25b}$$

$$M^{[s]} := \left\| \sum_{t=0}^{s-1} h_{A^t\mathcal{B}_\infty^{n_x}}(\tilde{\mathbf{I}}_{n_x}) \right\|_\infty = \left\| \sum_{t=0}^{s-1} |\tilde{\mathbf{I}}_{n_x} A^t|\mathbf{1} \right\|_\infty, \tag{5.25c}$$

and observe that the support function

$$h_{A^s\mathcal{B}_\infty^{n_x}}(\tilde{\mathbf{I}}_{n_x}) = |\tilde{\mathbf{I}}_{n_x} A^s|\mathbf{1} \leq \|A^s\|_\infty \mathbf{1},$$

following the usual definition of ∞-norm for matrices. Note that $L^{[s]} \in \mathbb{R}^{m_\mathcal{Y}}$ following from dimensions of matrix G. Then, the support function inequalities (5.24a)-(5.24c) can be written after simple algebraic manipulations as

$$\text{(5.24a)} \iff \lambda \leq (1-\alpha)\theta^{[s]}, \tag{5.26a}$$

$$\text{(5.24b)} \iff (\gamma+\lambda)\|A^s\|_\infty \leq \alpha\lambda, \tag{5.26b}$$

$$\text{(5.24c)} \iff (\alpha\gamma+\lambda)M^{[s]} \leq (1-\alpha)\mu. \tag{5.26c}$$

Hence, for a given $s > 0$, $\mathbb{H}(s)$ can be written as

$$\max_{\alpha,\lambda} \ \alpha + \lambda \tag{5.27a}$$

$$\text{s.t. } (5.26a),\ (5.26b),\ (5.26c), \tag{5.27b}$$

$$\alpha \in [0,1),\ \lambda \in [0,1]. \tag{5.27c}$$

While Constraints (5.24a) and (5.24c) in Problem (5.27) are linear in (α, λ), Constraint (5.26b) is nonlinear. However, it can be written as a second-order cone (SOC) by exploiting the fact that $\alpha, \lambda \geq 0$ in the feasible domain as follows. In these arguments, we denote $\zeta := \|A^s\|_\infty$ for notational convenience.

$$
\begin{aligned}
& & (\gamma+\lambda)\zeta &\le \alpha\lambda && \text{(5.28a)}\\
\Leftrightarrow & & \gamma\zeta &\le (\alpha-\zeta)\lambda && \text{(5.28b)}\\
\Leftrightarrow & & 4\gamma\zeta &\le 4(\alpha-\zeta)\lambda && \text{(5.28c)}\\
\Leftrightarrow & & 4\gamma\zeta+(\alpha-\zeta)^2+\lambda^2 &\le 4(\alpha-\zeta)\lambda+(\alpha-\zeta)^2+\lambda^2 && \text{(5.28d)}\\
\Leftrightarrow & & 4\gamma\zeta+(\alpha-\zeta-\lambda)^2 &\le (\alpha-\zeta+\lambda)^2 && \text{(5.28e)}\\
\Leftrightarrow & & \sqrt{4\gamma\zeta+(\alpha-\zeta-\lambda)^2} &\le \alpha-\zeta+\lambda && \text{(5.28f)}\\
\Leftrightarrow & & \left\|\begin{bmatrix} 2\sqrt{\gamma\zeta} \\ \alpha-\zeta-\lambda \end{bmatrix}\right\|_2 &\le \alpha-\zeta+\lambda. && \text{(5.28g)}
\end{aligned}
$$

Taking the square-root on both sides of the inequality in (5.28f) is valid since $\alpha-\zeta \ge 0$ from (5.28b) such that $\alpha-\zeta+\lambda \ge 0$, and the left-hand-side of (5.28e) is nonnegative. The inequality in (5.28g) is an SOC, such that Problem (5.27) is an SOCP in two-dimensions that can be solved very efficiently using off-the-shelf solvers.

Termination of Algorithm 3

Algorithm 3 terminates in finite time if and only if, for the user-specified $\gamma, \mu > 0$, there exists some index $s > 0$ such that $\mathbb{H}(s)$ is feasible. This is equivalent to the existence of some $s > 0$ such that the domain $\boldsymbol{\mathcal{L}}_{\gamma,\mu}(s)$ of $\mathbb{H}(s)$ defined in (5.22) is nonempty, that is in turn equivalent to nonemptiness of the set $\Lambda_{\gamma,\mu}$. In the following result, we show that indeed $\Lambda_{\gamma,\mu}$ is a nonempty set.

Theorem 5.1. *Suppose that Assumptions 5.1 and 3.1 hold. Then the set $\Lambda_{\gamma,\mu}$ of parameters (s, α, λ) verifying* (5.18)-(5.20) *is nonempty for any user-specified $\gamma > 0$ and $\mu > 0$.* □

Proof. For any given $\gamma, \mu > 0$ and some $s > 0$, we recall from (5.26c) that inclusion (5.20) is verified if and only if

$$\frac{\alpha\gamma+\lambda}{1-\alpha} M^{[s]} \le \mu. \tag{5.29}$$

By rearranging this inequality, we obtain

$$\alpha \le \frac{\mu - \lambda M^{[s]}}{\mu + \gamma M^{[s]}}. \tag{5.30}$$

Recalling the definition of $M^{[s]}$ from (5.25c), we define its limit

$$\hat{M} := \lim_{s \to \infty} M^{[s]},$$

and observe that since the set $\bigoplus_{t=0}^{\infty} A^t \mathcal{B}_{\infty}^{n_x}$ is compact under Assumption 3.1 [63] and the inclusion

$$\bigoplus_{t=0}^{s-1} A^t \mathcal{B}_{\infty}^{n_x} \subseteq \bigoplus_{t=0}^{\infty} A^t \mathcal{B}_{\infty}^{n_x} \tag{5.31}$$

holds for any $s > 0$, the inequalities $M^{[s]} \leq \hat{M} < \infty$ hold. This implies that

$$\frac{\mu - \lambda \hat{M}}{\mu + \gamma \hat{M}} \leq \frac{\mu - \lambda M^{[s]}}{\mu + \gamma M^{[s]}}$$

holds for any $s > 0$. Hence, for some user-specified $\gamma, \mu > 0$, if we select parameters $\alpha, \lambda \in (0, 1)$ verifying the inequality

$$\alpha \leq \frac{\mu - \lambda \hat{M}}{\mu + \gamma \hat{M}}, \tag{5.32}$$

then inclusion (5.20) will be verified for any $s > 0$.

Regarding inclusion (5.18), we recall from (5.26a) that it holds if and only if

$$\frac{\lambda}{1 - \alpha} \leq \theta^{[s]} \iff \alpha \leq 1 - \frac{\lambda}{\theta^{[s]}}. \tag{5.33}$$

Recalling the definition of $\theta^{[s]}$ from (5.25b), we define its limit

$$\hat{\theta} := \lim_{s \to \infty} \theta^{[s]}. \tag{5.34}$$

From the definition of $L^{[s]}$ in (5.25a), we observe that $L_i^{[s]}$ is monotonically nondecreasing in s for each component $i \in \mathbb{I}_1^{m_\mathcal{Y}}$ and g is a constant, $\theta^{[s]}$ is monotonically nonincreasing in s. In fact, it can be observed from that

$$\hat{L}_i := \lim_{s \to \infty} L_i^{[s]}, \qquad\qquad \hat{\theta} = \min_{i \in \mathbb{I}_1^{m_\mathcal{Y}}} \left\{ \frac{g_i}{\hat{L}_i} \right\}.$$

Since $\hat{L}_i < \infty$ under Assumption 3.1 and $g_i > 0$ under Assumption 5.1 for each component $i \in \mathbb{I}_1^{m_\mathcal{Y}}$, the inequalities $\theta^{[s]} \geq \hat{\theta} > 0$ hold. This implies that

$$1 - \frac{\lambda}{\hat{\theta}} \leq 1 - \frac{\lambda}{\theta^{[s]}},$$

such that that if we select some $\alpha, \lambda \in (0,1)$ satisfying

$$\alpha \leq 1 - \frac{\lambda}{\hat{\theta}}, \tag{5.35}$$

then inclusion (5.18) will be verified for any $s > 0$.

Thus, if there exist parameters $\alpha, \lambda \in (0,1)$ verifying inequalities (5.32) and (5.35), then these parameters verify inclusions (5.20) and (5.18) for all $s > 0$. We now demonstrate the existence of such parameters. To this end, we define

$$\hat{q} := \min \left\{ \frac{\min\{\mu, \hat{M}\}}{\hat{M}} \ , \quad \hat{\theta} \right\}, \tag{5.36}$$

and select $\lambda = \delta\hat{q}$ for some $\delta \in (0,1)$. For any user-specified $\mu > 0$, we have $\hat{q} \in (0,1)$ since $\hat{\theta}, \hat{M} > 0$. This implies that $\lambda = \delta\hat{q} \in (0,1)$ for any $\delta \in (0,1)$. We also define

$$\hat{r}(\delta) := \min \left\{ \frac{\mu - \delta\hat{q}\hat{M}}{\mu + \gamma\hat{M}} \ , \quad 1 - \frac{\delta\hat{q}}{\hat{\theta}} \right\}. \tag{5.37}$$

Then, substituting $\lambda = \delta\hat{q}$ in inequalities (5.32) and (5.35), we observe that if $\hat{r}(\delta) \in (0,1)$, then any $\alpha \in (0, \hat{r}(\delta)]$ verifies these inequalities. Hence, it remains to show that $\hat{r}(\delta) \in (0,1)$.

To that end, we note that since the inequalities

$$\mu - \delta\hat{q}\hat{M} < \mu + \gamma\hat{M} \qquad \text{and} \qquad \delta\hat{q}/\hat{\theta} > 0$$

hold for every $\delta \in (0,1)$, we always have $\hat{r}(\delta) < 1$. In order to show that $\hat{r}(\delta) > 0$, we consider the following two cases, which are based on Equation (5.36).

Case 1: if

$$\hat{q} = \frac{\mu}{\hat{M}} \leq \hat{\theta}, \tag{5.38}$$

then for any $\delta \in (0,1)$, we have either

$$\hat{r}(\delta) = (1-\delta)\left(\frac{\mu}{\mu + \gamma\hat{M}}\right) \qquad \text{or} \qquad \hat{r}(\delta) = 1 - \delta\left(\frac{\mu}{\hat{M}\hat{\theta}}\right).$$

Since $\mu, \gamma, \hat{M} > 0$, the first option satisfies $\hat{r}(\delta) > 0$. Regarding the second option, inequality (5.38) implies

$$\delta\mu < \mu \leq \hat{M}\hat{\theta} \qquad \Longrightarrow \qquad 1 - \delta\left(\frac{\mu}{\hat{M}\hat{\theta}}\right) = \hat{r}(\delta) > 0.$$

Case 2: if

$$\hat{q} = \hat{\theta} \leq \frac{\mu}{\hat{M}}, \tag{5.39}$$

then for any $\delta \in (0,1)$, we have either

$$\hat{r}(\delta) = \left(\frac{\mu - \delta\hat{\theta}\hat{M}}{\mu + \gamma\hat{M}} \right) \qquad \text{or} \qquad \hat{r}(\delta) = 1 - \delta$$

Since $\delta \in (0,1)$, the second option always satisfies $\hat{r}(\delta) > 0$. Regarding the first option, inequality (5.39) implies

$$\delta\hat{M}\hat{\theta} < \hat{M}\hat{\theta} \leq \mu \qquad \Longrightarrow \qquad \mu - \delta\hat{M}\hat{\theta} > 0,$$

such that $\hat{r}(\delta) > 0$ since $\mu, \gamma, \hat{M} > 0$. Thus, for any $\delta \in (0,1)$, we have $\hat{r}(\delta) \in (0,1)$, such that the parameters $\lambda = \delta\hat{q}$ and $\alpha \in (0, \hat{r}(\delta)]$ verify inclusions (5.20) and (5.18) for all $s > 0$.
Now, we show that there exists some $\alpha \in (0, \hat{r}(\delta)]$ and $s > 0$ also satisfying inclusion (5.19) with $\lambda = \delta\hat{q}$. To this end, we recall from (5.26b) that inclusion (5.19) is verified if and only if

$$(\gamma + \lambda) \left\| A^s \right\|_\infty \leq \alpha\lambda \tag{5.40}$$

holds. Then, defining

$$\alpha_{[s]}(\lambda) := \left(1 + \frac{\gamma}{\lambda}\right) \left\| A^s \right\|_\infty, \tag{5.41}$$

we observe that for a given $\lambda \in (0,1)$, $\gamma > 0$ and $s > 0$, $\alpha_{[s]}(\lambda)$ is the smallest value of α verifying inequality (5.40). This implies that if $\lambda = \delta\hat{q}$, then $\alpha_{[s]}(\delta\hat{q})$ verifies (5.40). Then, since Assumption 3.1 entails that [54]

$$\lim_{s\to\infty} \left\| A^s \right\|_\infty = 0,$$

there always exists some index $s = \hat{s}(\delta)$ such that $\alpha_{[\hat{s}(\delta)]}(\delta\hat{q}) \leq \hat{r}(\delta)$. Since such a triplet of parameters

$$(s = \hat{s}(\delta), \alpha \in (0, \hat{r}(\delta)], \lambda = \delta\hat{q}), \qquad \forall\, \delta \in (0,1), \tag{5.42}$$

also satisfies (5.20) and (5.18), the set $\Lambda_{\gamma,\mu}$ is nonempty. $\square$

Thus, using Algorithm 3, feasible parameters (s, α, λ) can be computed, using which Problem (5.17) can be formulated with the guarantee that the set $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$ is μ-RPI for every feasible disturbance set $\mathcal{W}$. We now recall again the formulation of this problem, in which we explicitly incorporate the objective $d_{\mathcal{Y}}(\cdot)$ defined in Equation (3.8):

$$\min_{\epsilon, \mathcal{W}} \quad \|\epsilon\|_1 \tag{5.43a}$$
$$\text{s.t.} \quad \mathcal{O}(s, \alpha, \lambda, \mathcal{W}) \subseteq \mathcal{Y}, \tag{5.43b}$$
$$B\mathcal{W} \subseteq \gamma \mathcal{B}_{\infty}^{n_x}, \tag{5.43c}$$
$$\mathbf{0} \in \mathcal{W}, \tag{5.43d}$$
$$\mathcal{Y} \subseteq \mathcal{S}(l, \mathcal{W}) \oplus \mathbb{B}(\epsilon). \tag{5.43e}$$

In the sequel, we present an optimization algorithm to solve Problem (5.43) after selecting an appropriate representation for the disturbance set $\mathcal{W}$.

Remark 5.1. *The conservativeness introduced due to constraint* (5.43c)*, i.e.,* $B\mathcal{W} \subseteq \gamma \mathcal{B}_{\infty}^{n_x}$*, in Problem* (5.43) *with respect to Problem* (5.14)*, can be eliminated by selecting a* $\gamma > 0$ *large enough such that every disturbance set* $\mathcal{W}$ *feasible for constraint* (5.43b) *satisfies constraint* (5.43c)*, thus rendering constraint* (5.43c) *inactive and making Problem* (5.43) *fully equivalent to Problem* (5.14)*. This, however, increases the complexity of Problem* (5.43)*. As* $\gamma > 0$ *increases for some fixed* $\lambda \in (0, 1)$ *and* $s > 0$*, the smallest value of* $\alpha = \alpha_{[s]}(\lambda)$ *verifying inequality* (5.40) *increases, as observed from the definition of* $\alpha_{[s]}(\lambda)$ *in Equation* (5.41)*. Then, the value* $s > 0$ *required to verify inequalities* (5.30) *and* (5.33) *with* $\alpha = \alpha_{[s]}(\lambda)$ *increases. This implies that the set* $\mathcal{O}(s, \alpha, \lambda, \mathcal{W})$ *defined in* (5.13) *and formulating constraint* (5.43b) *requires a larger number of Minkowski sums for its characterization, such that the complexity of Problem* (5.43) *increases.* □

5.2 Solving Problem (5.43)

In this section, we present a tractable encoding of the constraints of Problem (5.43), following which we present an optimization algorithm to solve the resulting problem.

5.2.1 Inclusions (5.43b) and (5.43c)

To encode inclusion (5.43b), i.e., $\mathcal{O}(s,\alpha,\lambda,\mathcal{W}) \subseteq \mathcal{Y}$, we recall that the constraint set is given by $\mathcal{Y} = \{y : Gy \leq g\}$, and the set $\mathcal{O}(s,\alpha,\lambda,\mathcal{W})$ is defined in Equation (5.13) as

$$\mathcal{O}(s,\alpha,\lambda,\mathcal{W}) = (1-\alpha)^{-1} \bigoplus_{t=0}^{s-1} CA^t(B\mathcal{W} \oplus \lambda\mathcal{B}_\infty^{n_x}) \oplus D\mathcal{W}.$$

Defining the matrices

$$\bar{G}_{[t]} := (1-\alpha)^{-1} GCA^t, \qquad \forall\, t \in \mathbb{I}_0^{s-1}, \tag{5.44}$$

we observe that the inclusion is verified if and only if the support function inequality

$$\sum_{t=0}^{s-1} h_{B\mathcal{W}}(\bar{G}_{[t]}) + h_{D\mathcal{W}}(G) \leq g - \lambda \sum_{t=0}^{s-1} h_{\mathcal{B}_\infty^{n_x}}(\bar{G}_{[t]}) \tag{5.45}$$

holds . Similarly, inclusion (5.43c), i.e., $B\mathcal{W} \subseteq \gamma\mathcal{B}_\infty^{n_x}$, is verified if and only if the support function inequality

$$h_{B\mathcal{W}}(\tilde{\mathbf{I}}_{n_x}) \leq \gamma \mathbf{1}, \tag{5.46}$$

holds according to basic properties of support functions. In order to encode the support function inequalities in (5.45) and (5.46) efficiently, we propose to use the disturbance set parameterization

$$\mathcal{W} = \mathrm{ConvHull}\left(\mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]}),\ j \in \mathbb{I}_1^N\right), \tag{5.47a}$$

$$\mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]}) := \bar{w}_{[j]} \oplus \{w : -\epsilon^w_{[j]} \leq w \leq \epsilon^w_{[j]}\}, \tag{5.47b}$$

i.e., as a convex hull of boxes $\{\mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]}), j \in \mathbb{I}_1^N\}$ where $N \geq 1$ is a user-specified amount of boxes, such that $\mathcal{W}$ is characterized by parameters $\{\bar{w}_{[j]}, \epsilon^w_{[j]} \in \mathbb{R}^{n_w}, j \in \mathbb{I}_1^N\}$. This parameterization presents a representational advantage over popular polytopic parameterizations such as zonotopes [127] that are constrained to be symmetric and hence can be conservative, and a computational advantage over parameterizations such as halfspace-representations [97], zonotopic intersections [**Althoff2011**], constrained zonotopes [**Scott2016**], etc. that are not immediately amenable to a simple encodings of the support function inequalities (5.45)-(5.46). Finally, it requires fewer parameters than simple vertex notations ($N \times 2n_w$ as opposed to $N \times 2^{n_w}$).

In order to encode the support function inclusions in (5.45) and (5.46) for the disturbance set parameterization in (5.47), we rely on the following general result regarding support functions over convex hulls of polytopes.

Proposition 5.2. *Given any polytopes $\{\mathcal{Q}_j \subset \mathbb{R}^n,\ j \in \mathbb{I}_1^q\}$, and denoting the convex hull*

$$\hat{\mathcal{Q}} := \mathrm{ConvHull}(\mathcal{Q}_j,\ j \in \mathbb{I}_1^q),$$

then for any matrix $\mathbf{T} \in \mathbb{R}^{l \times n}$ and vector $\mathbf{p} \in \mathbb{R}^l$, the support function for the set $\mathbf{T}\hat{\mathcal{Q}}$ at $\mathbf{p}$ is given by

$$h_{\mathbf{T}\hat{\mathcal{Q}}}(\mathbf{p}) = \max_{j \in \mathbb{I}_1^q} h_{\mathbf{T}\mathcal{Q}_j}(\mathbf{p}). \qquad \square \tag{5.48}$$

Proof. Since the set $\hat{\mathcal{Q}}$ is a polytope that is the convex hull of polytopes $\{\mathcal{Q}_j,\ j \in \mathbb{I}_1^q\}$, we know that

$$\mathrm{vert}(\hat{\mathcal{Q}}) \subseteq \bigcup_{j=1}^{q} \mathrm{vert}(\mathcal{Q}_j), \tag{5.49a}$$

$$\nexists\ \tilde{z}\ :\ \tilde{z} \in \bigcup_{j=1}^{q} \mathcal{Q}_j,\ \tilde{z} \notin \hat{\mathcal{Q}}. \tag{5.49b}$$

Defining $\mathbf{r} := \mathbf{T}^\top \mathbf{p}$, we know from the support function definition that (5.48) holds if and only if

$$\max_{z \in \hat{\mathcal{Q}}} \mathbf{r}^\top z = \max_{j \in \mathbb{I}_1^q} \left\{ \max_{z_{[j]} \in \mathcal{Q}_j} \mathbf{r}^\top z_{[j]} \right\}. \tag{5.50}$$

We now prove (5.50) by contradiction. Suppose that

$$\max_{z \in \hat{\mathcal{Q}}} \mathbf{r}^\top z\ >\ \max_{j \in \mathbb{I}_1^q} \left\{ \max_{z_{[j]} \in \mathcal{Q}_j} \mathbf{r}^\top z_{[j]} \right\}, \tag{5.51}$$

which holds if and only if there exists a vertex $z^* \in \mathrm{vert}(\hat{\mathcal{Q}})$ such that

$$\mathbf{r}^\top z^* = \max_{z \in \hat{\mathcal{Q}}} \mathbf{r}^\top z \text{ and } z^* \notin \bigcup_{j=1}^{q} \mathrm{vert}(\mathcal{Q}_j).$$

Since this contradicts (5.49a), (5.51) cannot hold. Now, suppose

$$\max_{z \in \hat{\mathcal{Q}}} \mathbf{r}^\top z \; < \; \max_{j \in \mathbb{I}_1^q} \left\{ \max_{z_{[j]} \in \mathcal{Q}_j} \mathbf{r}^\top z_{[j]} \right\}, \tag{5.52}$$

which holds if and only if there exists some $\tilde{z}^*$ such that

$$\tilde{z}^* \in \bigcup_{j=1}^{q} \mathcal{Q}_j \text{ and } \tilde{z}^* \notin \hat{\mathcal{Q}}.$$

Since this contradicts (5.49b), (5.52) cannot hold and the proof is complete. □

Using Proposition 5.2 with $\mathcal{Q}_j = \mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]})$, $q = N$ and $\hat{\mathcal{Q}} = \mathcal{W}$ as per the definition of the disturbance set in (5.47), and recalling the expression for support functions of over boxes from (5.2), the support function over $\mathcal{W}$ for any given matrix $\mathbf{T} \in \mathbb{R}^{l \times n_w}$ and vector $\mathbf{p} \in \mathbb{R}^l$ is obtained as

$$h_{\mathbf{T}\mathcal{W}}(\mathbf{p}) = \max_{j \in \mathbb{I}_1^N} \{\mathbf{p}^\top \mathbf{T} \bar{w}_{[j]} + |\mathbf{p}^\top \mathbf{T}| \epsilon^w_{[j]}\}. \tag{5.53}$$

We now exploit (5.53) to encode (5.45)-(5.46) as linear inequalities. Firstly, inequality (5.45) holds by Equation (5.53) if and only if

$$\sum_{t=0}^{s-1} \left(\max_{j \in \mathbb{I}_1^N} \left\{ \bar{G}_{[t]} B \bar{w}_{[j]} + |\bar{G}_{[t]} B| \epsilon^w_{[j]} \right\} \right) + \left(\max_{j \in \mathbb{I}_1^N} \left\{ G D \bar{w}_{[j]} + |GD| \epsilon^w_{[j]} \right\} \right) \leq g - \lambda \sum_{t=0}^{s-1} |\bar{G}_{[t]}| \mathbf{1}. \tag{5.54}$$

To encode (5.54), we introduce $\mathbf{Q} := \{\mathbf{Q}_{[t]} \in \mathbb{R}^{m_{\mathcal{Y}}},\ t \in \mathbb{I}_0^{s-1}\}$ and $\mathbf{r} \in \mathbb{R}^{m_{\mathcal{Y}}}$, along with the inequalities

$$\forall\, j \in \mathbb{I}_1^N,\ \forall\, t \in \mathbb{I}_0^{s-1} \left\{ \begin{array}{l} \bar{G}_{[t]} B \bar{w}_{[j]} + |\bar{G}_{[t]} B| \epsilon^w_{[j]} \leq \mathbf{Q}_{[t]}, \\ G D \bar{w}_{[j]} + |GD| \epsilon^w_{[j]} \leq \mathbf{r}. \end{array} \right. \tag{5.55}$$

Then, the inequality in (5.54) holds if and only if there exists some $(\mathbf{Q}, \mathbf{r})$ satisfying the inequalities in (5.55) along with

$$\sum_{t=0}^{s-1} \mathbf{Q}_{[t]i} + \mathbf{r}_i \leq g_i - \lambda \sum_{t=0}^{s-1} |\bar{G}_{[t]i}|\mathbf{1}, \qquad \forall\, i \in \mathbb{I}_1^{m_{\mathcal{Y}}}. \tag{5.56}$$

Thus, we encode the support function inequality in (5.45) for the disturbance set parameterization in (5.47) as the linear inequalities (5.55) and (5.56). Similarly, support function inequality (5.46) holds according to Equation (5.53) if and only if

$$\tilde{\mathbf{I}}_{n_x} B\bar{w}_{[j]} + |\tilde{\mathbf{I}}_{n_x} B|\epsilon^w_{[j]} \leq \gamma\mathbf{1}, \qquad \forall\, j \in \mathbb{I}_1^N. \tag{5.57}$$

5.2.2 Inclusions (5.43d) and (5.43e)

We encode inclusion (5.43d), i.e., $\mathbf{0} \in \mathcal{W}$, by enforcing the inclusion $\mathbf{0} \in \mathbb{W}(\bar{w}_{[1]}, \epsilon^w_{[1]})$ for simplicity through

$$\begin{bmatrix} \mathbf{I}_{n_w} \\ -\mathbf{I}_{n_w} \end{bmatrix} \mathbf{0} \leq \begin{bmatrix} \epsilon^w_{[1]} \\ \epsilon^w_{[1]} \end{bmatrix} + \begin{bmatrix} \mathbf{I}_{n_w} \\ -\mathbf{I}_{n_w} \end{bmatrix} \bar{w}_{[1]} \tag{5.58}$$

since $\mathbf{0} \in \mathbb{W}(\bar{w}_{[1]}, \epsilon^w_{[1]})$ implies $\mathbf{0} \in \mathcal{W}$ from (5.47).

In order to encode inclusion (5.43e), i.e., $\mathcal{Y} \subseteq \mathcal{S}(l, \mathcal{W}) \oplus \mathbb{B}(\epsilon)$, we recall from (5.15) that the l-step reachable set $\mathcal{S}(l, \mathcal{W})$ is

$$\mathcal{S}(l, \mathcal{W}) = \bigoplus_{t=0}^{l-1} CA^t B\mathcal{W} \oplus D\mathcal{W}.$$

Then, we note that that the inclusion holds if and only if the support function inequality

$$h_{\mathcal{Y}}(\mathbf{p}) \leq \sum_{t=0}^{l-1} h_{CA^tB\mathcal{W}}(\mathbf{p}) + h_{D\mathcal{W}}(\mathbf{p}) + h_{\mathbb{B}(\epsilon)}(\mathbf{p}) \tag{5.59}$$

is verified for all $\mathbf{p} \in \mathbb{R}^{n_y}$. If the hyperplane notation of $\mathcal{W}$ is known, then the result in [127, Theorem 1] can be used to to derive sufficient linear conditions for the inequality in (5.59). Unfortunately, since the hyperplane notation is unknown a priori because of the parameterization of $\mathcal{W}$

in (5.47), we instead rely on Proposition 5.1(e) to encode inclusion (5.43e). Denoting

$$\{\mathrm{y}_{[i]},\ i \in \mathbb{I}_1^{v\mathcal{Y}}\} := \mathrm{vert}(\mathcal{Y}),$$

and recalling the definition of $\mathcal{S}(l, \mathcal{W})$ from (5.15), we know from Proposition 5.1(e) that inclusion (5.43e) holds if and only if

$$\forall\, i \in \mathbb{I}_1^{v\mathcal{Y}},\ \exists \left\{ \begin{array}{l} \left\{ \{\mathrm{w}^1_{[it]},\ t \in \mathbb{I}_0^{l-1}\},\ \mathrm{w}^2_{[i]} \right\} \in \mathcal{W}, \\ \mathrm{b}_{[i]} \in \mathbb{B}(\epsilon), \end{array} \right. \tag{5.60a}$$

$$\text{such that } \mathrm{y}_{[i]} = \sum_{t=0}^{l-1} CA^{l-1-t} B\mathrm{w}^1_{[it]} + D\mathrm{w}^2_{[i]} + \mathrm{b}_{[i]}, \tag{5.60b}$$

where, for each $i \in \mathbb{I}_1^{v\mathcal{Y}}$, feasible disturbance sequences in (5.60a) drive the output of System (7.1) in l-steps to some

$$\mathbf{y}_{[i]}(l) := \sum_{t=0}^{l-1} CA^{l-1-t} B\mathrm{w}^1_{[it]} + D\mathrm{w}^2_{[i]},$$

that belongs in the vicinity of vertex $\mathrm{y}_{[i]}$ as $\mathrm{y}_{[i]} - \mathbf{y}_{[i]}(l) \in \mathbb{B}(\epsilon)$. We will show in the sequel that the conditions in (5.60) verifying inclusion (5.43e) can be tractably encoded using necessary and sufficient conditions for the disturbance set parameterization in (5.47). To that end, we assume to know the vertices of the output constraint set $\mathcal{Y}$.

Assumption 5.2. *The vertices* $\{\mathrm{y}_{[i]}, i \in \mathbb{I}_1^{v\mathcal{Y}}\}$ *are known.* □

In order to encode the conditions in (5.60), we need to enforce the point-wise constraints $\mathrm{w}^1_{[it]} \in \mathcal{W}$ and $\mathrm{w}^2_{[i]} \in \mathcal{W}$ in (5.60a). According to the parameterization of $\mathcal{W}$ in (5.47), these constraints can be enforced by guaranteeing that points $\mathrm{w}^1_{[it]}$ and $\mathrm{w}^2_{[i]}$ belong to the convex hull of boxes $\mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]})$ as

$$\mathrm{w}^1_{[it]}, \mathrm{w}^2_{[i]} \in \mathrm{ConvHull}(\mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]}),\ j \in \mathbb{I}_1^N).$$

In the following result, we show that this condition is equivalent to enforcing $\mathrm{w}^1_{[it]}$ and $\mathrm{w}^2_{[i]}$ to belong to convex hulls of points, with each point belonging to a box $\mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]})$.

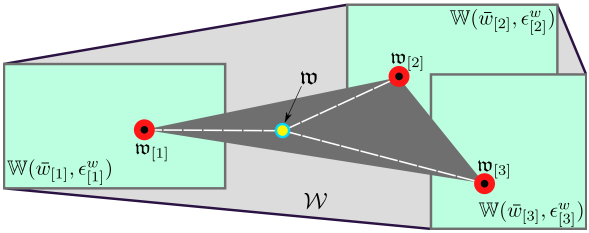


Figure 5: Illustration of Proposition 5.3 for $\mathcal{W}$ parameterized as a convex hull of $N = 3$ boxes.

Proposition 5.3. *Given the disturbance set parametrization in Equation* (5.47)*, there exists some* $\mathfrak{w} \in \mathcal{W}$ *if and only if there exist some* $\mathfrak{w}_{[j]} \in \mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]})$ *for each* $j \in \mathbb{I}_1^N$ *such that* $\mathfrak{w} \in \mathrm{ConvHull}(\mathfrak{w}_{[j]}, j \in \mathbb{I}_1^N)$. □

Proof. Sufficiency follows by observing that

$$\mathfrak{w} \in \mathrm{ConvHull}(\mathfrak{w}_{[j]} \in \mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]}), j \in \mathbb{I}_1^N) \implies \mathfrak{w} \in \mathcal{W}$$

from Equation (5.47). For the necessary condition, we define

$$\{\mathfrak{v}_{[ji]}, i \in \mathbb{I}_1^{2^{n_w}}\} := \mathrm{vert}(\mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]})),$$

and note that if $\mathfrak{w} \in \mathcal{W}$, then there exist $\mathfrak{p}_{[ji]} \geq 0$ satisfying

$$\mathfrak{w} = \sum_{j=1}^{N} \left(\sum_{i=1}^{2^{n_w}} \mathfrak{p}_{[ji]} \mathfrak{v}_{[ji]} \right), \qquad \sum_{j=1}^{N} \sum_{i=1}^{2^{n_w}} \mathfrak{p}_{[ji]} = 1, \tag{5.61}$$

by convexity of $\mathcal{W}$. Then, we define $\hat{\mathfrak{p}}_{[j]} := \sum_{i=1}^{2^{n_w}} \mathfrak{p}_{[ji]}$, and consider the two following cases:

$$\begin{aligned} &1) \text{ If } \hat{\mathfrak{p}}_{[j]} > 0: \ \text{Set } \hat{\mathfrak{v}}_{[j]} := \left(\sum_{i=1}^{2^{n_w}} \mathfrak{p}_{[ji]} \mathfrak{v}_{[ji]} \right) / \hat{\mathfrak{p}}_{[j]}; \\ &2) \text{ If } \hat{\mathfrak{p}}_{[j]} = 0: \ \text{Select any } \hat{\mathfrak{v}}_{[j]} \in \mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]}). \end{aligned}$$

We note that $\hat{\mathfrak{v}}_{[j]} \in \mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]})$ in Case 1, and Case 2 occurs if and only if $\mathfrak{p}_{[ji]} = 0$ for all $i \in \mathbb{I}_1^{2^{n_w}}$. Finally, we observe that the equations in (5.61) can be rearranged as

$$\mathfrak{w} = \sum_{j=1}^{N} \hat{\mathfrak{p}}_{[j]} \hat{\mathfrak{v}}_{[j]}, \qquad \sum_{j=1}^{N} \hat{\mathfrak{p}}_{[j]} = 1.$$

Variable	Dimension	Variable	Dimension
$\mathbf{x}$	$2Nn_w + (s+1)m\mathcal{Y}$	$\boldsymbol{\beta}$	$v_\mathcal{Y} \times N(l+1)$
$\mathbf{w}$	$v_\mathcal{Y} \times (l+1)n_w$	$\mathbf{z}$	$n_B + v_\mathcal{Y} \times n_y$
$\bar{\mathbf{w}}$	$v_\mathcal{Y} \times N(l+1)n_w$		

Table 3: Dimensions of variables defined in (5.64).

Setting $\mathfrak{w}_{[j]} = \hat{\mathfrak{v}}_{[j]} \in \mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]})$ concludes the proof. $\square$

We illustrate the idea of Proposition 5.3 in Figure 5. Using this result, we replace inclusions $\mathrm{w}^1_{[it]}, \mathrm{w}^2_{[i]} \in \mathcal{W}$ in (5.60a) with the equivalent inclusions

$$\begin{aligned} \mathrm{w}^1_{[it]} \in \mathcal{W}^1_{[it]} &:= \mathrm{ConvHull}(\bar{\mathrm{w}}^1_{[itj]} \in \mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]}), j \in \mathbb{I}_1^N), \\ \mathrm{w}^2_{[i]} \in \mathcal{W}^2_{[i]} &:= \mathrm{ConvHull}(\bar{\mathrm{w}}^2_{[ij]} \in \mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]}), j \in \mathbb{I}_1^N), \end{aligned} \tag{5.62}$$

by introducing variables $\bar{\mathrm{w}}^1_{[itj]}, \bar{\mathrm{w}}^2_{[ij]} \in \mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]})$. Then, we write the conditions in (5.60) equivalently as

$$\forall\, j \in \mathbb{I}_1^N,\ \forall\, t \in \mathbb{I}_0^{l-1},\ \forall\, i \in \mathbb{I}_1^{v_\mathcal{Y}}, \tag{5.63a}$$

$$\mathrm{y}_{[i]} = \sum_{t=0}^{l-1} CA^{l-1-t}B\mathrm{w}^1_{[it]} + D\mathrm{w}^2_{[i]} + \mathrm{b}_{[i]}, \tag{5.63b}$$

$$\mathrm{w}^1_{[it]} = \sum_{j=1}^{N} \beta^1_{[itj]}\bar{\mathrm{w}}^1_{[itj]},\ \mathrm{w}^2_{[i]} = \sum_{j=1}^{N} \beta^2_{[ij]}\bar{\mathrm{w}}^2_{[ij]}, \tag{5.63c}$$

$$\bar{\mathrm{w}}^1_{[itj]} \in \mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]}),\ \bar{\mathrm{w}}^2_{[ij]} \in \mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]}), \tag{5.63d}$$

$$\sum_{j=1}^{N} \beta^1_{[itj]} = 1, \beta^1_{[itj]} \geq 0,\ \sum_{j=1}^{N} \beta^2_{[ij]} = 1, \beta^2_{[ij]} \geq 0, \tag{5.63e}$$

$$\mathrm{b}_{[i]} \in \mathbb{B}(\epsilon), \tag{5.63f}$$

in which the variables $\beta^1_{[itj]}$ and $\beta^2_{[ij]}$ are introduced to encode the convex-hull inclusions in (5.62) through (5.63c) and (5.63e).

Thus, we encode constraints (5.43b)-(5.43e) as (5.55)-(5.57), (5.58) and (5.63) respectively. For simplicity of notation in the sequel, we denote the

Constraint	#
(5.55)-(5.57), (5.58)	$(s+1)m_{\mathcal{Y}} + 2(n_x + n_w)$ Lin. ineq.
(5.63b)	$v_{\mathcal{Y}} \times n_y$ Lin. eq.
(5.63c)	$v_{\mathcal{Y}} \times (l+1)n_w$ Bilin. eq.
(5.63d)	$v_{\mathcal{Y}} \times 2N(l+1)n_w$ Lin. ineq.
(5.63e)	$v_{\mathcal{Y}} \times N(l+1)$ Lin. ineq., $v_{\mathcal{Y}} \times (l+1)$ Lin. eq
(5.63f)	$v_{\mathcal{Y}} \times n_B$ Linear ineq.

Table 4: Number of constraints.

collections of variables

$$
\begin{aligned}
\mathbf{x} &:= \{\{\bar{w}_{[j]}, \epsilon^{w}_{[j]}, j \in \mathbb{I}_1^N\}, \mathbf{Q}, \mathbf{r}\}, && \text{(5.64a)}\\
\mathbf{w} &:= \{\mathrm{w}^1_{[it]}, \mathrm{w}^2_{[i]},\ i \in \mathbb{I}_1^{v_{\mathcal{Y}}}, t \in \mathbb{I}_0^{l-1}\}, && \text{(5.64b)}\\
\bar{\mathbf{w}} &:= \{\bar{\mathrm{w}}^1_{[itj]}, \bar{\mathrm{w}}^2_{[ij]},\ i \in \mathbb{I}_1^{v_{\mathcal{Y}}}, t \in \mathbb{I}_0^{l-1}, j \in \mathbb{I}_1^N\}, && \text{(5.64c)}\\
\boldsymbol{\beta} &:= \{\beta^1_{[itj]}, \beta^2_{[ij]},\ i \in \mathbb{I}_1^{v_{\mathcal{Y}}}, t \in \mathbb{I}_0^{l-1}, j \in \mathbb{I}_1^N\}, && \text{(5.64d)}\\
\mathbf{z} &:= [\epsilon^\top\ \mathrm{b}_{[1]}^\top\ \cdots\ \mathrm{b}_{v_{\mathcal{Y}}}^\top]^\top, && \text{(5.64e)}
\end{aligned}
$$

and denote $\mathbf{v} := \{\mathbf{x}, \mathbf{w}, \bar{\mathbf{w}}, \boldsymbol{\beta}, \mathbf{z}\}$. Over these variables, we denote the constraints in (5.55)-(5.57), (5.58) and (5.63) as

$$
\begin{aligned}
&\mathbf{x} \text{ satisfies (5.55)-(5.57), (5.58)} && \Leftrightarrow && \mathbf{A}\mathbf{x} \le \mathbf{b}, && \text{(5.65a)}\\
&\mathbf{w}, \mathbf{z} \text{ satisfy (5.63b)} && \Leftrightarrow && \mathbf{C}_{\mathbf{w}}\mathbf{w} + \mathbf{C}_{\mathbf{z}}\mathbf{z} = \mathbf{h}, && \text{(5.65b)}\\
&\mathbf{w}, \bar{\mathbf{w}}, \boldsymbol{\beta} \text{ satisfy (5.63c)} && \Leftrightarrow && \mathbf{g}(\bar{\mathbf{w}}, \boldsymbol{\beta}) = \mathbf{w}, && \text{(5.65c)}\\
&\mathbf{x}, \bar{\mathbf{w}} \text{ satisfy (5.63d)} && \Leftrightarrow && \mathbf{D}_{\mathbf{x}}\mathbf{x} + \mathbf{D}_{\bar{\mathbf{w}}}\bar{\mathbf{w}} \le \mathbf{0}, && \text{(5.65d)}\\
&\boldsymbol{\beta} \text{ satisfies (5.63e)} && \Leftrightarrow && \boldsymbol{\beta} \ge \mathbf{0},\ \ \mathbf{T}_{\boldsymbol{\beta}}\boldsymbol{\beta} = \mathbf{1}, && \text{(5.65e)}\\
&\mathbf{z} \text{ satisfies (5.63f)} && \Leftrightarrow && \mathbf{E}_{\mathbf{z}}\mathbf{z} \le \mathbf{0}. && \text{(5.65f)}
\end{aligned}
$$

The definitions of the matrices formulating (5.65) are given in Appendix B. Finally, we define the cost vector $\mathbf{c} := [\mathbf{1}_{n_B}^\top \quad \mathbf{0}_{n_y v_{\mathcal{Y}}}^\top]^\top$, such that $\mathbf{c}^\top\mathbf{z} =$

$\|\epsilon\|_1$. Then, we write Problem (5.43) as

$$
\begin{aligned}
\min_{\mathbf{v}=\{\mathbf{x},\mathbf{w},\bar{\mathbf{w}},\boldsymbol{\beta},\mathbf{z}\}} \quad & \mathbf{c}^\top \mathbf{z} && \text{(5.66a)}\\
\text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b}, && \text{(5.66b)}\\
& \mathbf{D}_{\mathbf{x}}\mathbf{x} + \mathbf{D}_{\bar{\mathbf{w}}}\bar{\mathbf{w}} \le \mathbf{0}, && \text{(5.66c)}\\
& \mathbf{C}_{\mathbf{w}}\mathbf{w} + \mathbf{C}_{\mathbf{z}}\mathbf{z} = \mathbf{h}, && \text{(5.66d)}\\
& \mathbf{E}_{\mathbf{z}}\mathbf{z} \le \mathbf{0}, && \text{(5.66e)}\\
& \boldsymbol{\beta} \ge \mathbf{0}, \;\; \mathbf{T}_{\boldsymbol{\beta}}\boldsymbol{\beta} = \mathbf{1}, && \text{(5.66f)}\\
& \mathbf{g}(\bar{\mathbf{w}}, \boldsymbol{\beta}) = \mathbf{w}. && \text{(5.66g)}
\end{aligned}
$$

The number of variables and constraints defining Problem (5.66) are shown in Tables 3 and 4 respectively, in which we observe that the number of variables and constraints scale linearly with the number of vertices $v_{\mathcal{Y}}$ of the output constraint set $\mathcal{Y}$. This problem is composed of a linear objective and polyhedral constraints, along with bilinear equality constraints in (5.66g) resulting from (5.63c). Since this problem is smooth, it can be solved to local optimality using any off-the-shelf NonLinear Programing (NLP) solver [102].

5.3 Approximate solutions of Problem (5.43)

While an NLP approach can be used to solve Problem (5.43) as discussed in the Section 5.2, the implementation of NLP solvers can be cumbersome in practice. Moreover, the quality of solutions computed by an NLP solver depends on the initial point. Hence, in this section, we present two simpler approaches to solve Problem (5.43) that are based on linear programing. In the first approach, we approximately solve Problem (5.66) using an alternating-minimization algorithm. The output of this algorithm can be used to initialize an NLP solver to solve Problem (5.66). In the second approach, we use a convex hull of zonotopes parameterization of the disturbance set $\mathcal{W}$, and exploit Proposition 5.2 to formulate an LP approximation of Problem (5.43).

5.3.1 Alternating-minimization approach

In this approach, we use the convex hull of boxes parameterization in (5.47) of the disturbance set $\mathcal{W}$. For this parameterization, we recall from Section 5.2 that Problem (5.43) can be formulated as Problem (5.66). While Problem (5.66) is composed of bilinear equality constraints (5.66g), we observe that the problem reduces to an LP for a fixed value of $\boldsymbol{\beta}$. Hence, we propose to approximate Problem (5.66) as the LP

$$\mathbb{P}(\boldsymbol{\beta}_{\mathrm{f}}) \begin{cases} \{\mathbf{x}_*, \mathbf{w}(\boldsymbol{\beta}_{\mathrm{f}}), \bar{\mathbf{w}}_*, \mathbf{z}(\boldsymbol{\beta}_{\mathrm{f}})\} := \underset{\mathbf{x},\mathbf{w},\bar{\mathbf{w}},\mathbf{z}}{\arg\min} \quad \mathbf{c}^\top \mathbf{z} \\ \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{s.t.} \quad (5.66\text{b}) - (5.66\text{e}), \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathbf{g}(\bar{\mathbf{w}}, \boldsymbol{\beta}_{\mathrm{f}}) = \mathbf{w}, \end{cases}$$

where $\boldsymbol{\beta}_{\mathrm{f}}$ is some value of $\boldsymbol{\beta}$ that satisfies constraint (5.66f)

A special case of arises when the number of boxes parameterizing the disturbance set is equal to the number of vertices of the output constraint set $\mathcal{Y}$, i.e., $N = v_{\mathcal{Y}}$. Then, problem $\mathbb{P}(\boldsymbol{\beta}_{\mathrm{f}})$ can be solved with the components of $\boldsymbol{\beta}_{\mathrm{f}}$ selected as

$$\forall\, i \in \mathbb{I}_1^{v_{\mathcal{Y}}}, \forall\, t \in \mathbb{I}_0^{s-1},\ \beta^1_{[itj]},\ \beta^2_{[ij]} = \begin{cases} 1, \text{ if } i = j, \\ 0, \text{ otherwise.} \end{cases} \tag{5.67}$$

This is equivalent to enforcing the disturbance sequences $\{\{\mathrm{w}^1_{[it]},\ t \in \mathbb{I}_0^{l-1}\},\ \mathrm{w}^2_{[i]}\}$ corresponding to vertex $\mathrm{y}_{[i]}$ of the disturbance set $\mathcal{Y}$ inside the box $\mathbb{W}(\bar{w}_{[i]}, \epsilon^w_{[i]})$, instead of in the set $\mathcal{W}$ as done in (5.60a).

While $\mathbb{P}(\boldsymbol{\beta}_{\mathrm{f}})$ provides an efficient way to approximate Problem (5.66), conservativeness can be reduced further by also optimizing over $\boldsymbol{\beta}$. To this end, we observe that Problem (5.66) reduces to an LP also for a fixed value of $\bar{\mathbf{w}}$. Based on this observation, we propose to solve the LP

$$\mathbb{Q}(\bar{\mathbf{w}}_*) \begin{cases} \{\mathbf{w}_*, \mathbf{z}_*, \boldsymbol{\beta}_*\} := \quad \underset{\mathbf{w},\mathbf{z},\boldsymbol{\beta}}{\arg\min} \quad \mathbf{c}^\top \mathbf{z} \\ \qquad\qquad\qquad\qquad\quad \text{s.t.} \quad (5.66\text{d}) - (5.66\text{f}), \\ \qquad\qquad\qquad\qquad\qquad\quad \mathbf{g}(\bar{\mathbf{w}}_*, \boldsymbol{\beta}) = \mathbf{w}, \end{cases}$$

where $\bar{\mathbf{w}}_*$ is an optimizer of problem $\mathbb{P}(\boldsymbol{\beta}_{\mathrm{f}})$. Using LPs $\mathbb{P}(\boldsymbol{\beta}_{\mathrm{f}})$ and $\mathbb{Q}(\bar{\mathbf{w}}_*)$, we define an alternating-minimization procedure in Algorithm 4 to approximately solve Problem (5.66), in which we select $\boldsymbol{\beta}_{\mathrm{f}} = \boldsymbol{\beta}_*$ and repeat

Algorithm 4: Alternating-minimization for Problem (5.66)

Result: Return $\mathbf{v}_*^{[\iota]}$;
Input: Initial $\boldsymbol{\beta} = \boldsymbol{\beta}_*^{[0]}$ satisfying (5.66f),$\zeta > 0$;
Initialize: $conv = 0$, $\iota = 1$;
while $conv = 0$ **do**
  1. Solve $\mathbb{P}(\boldsymbol{\beta}_*^{[\iota-1]})$ for $\left\{\mathbf{x}_*^{[\iota]}, \mathbf{w}(\boldsymbol{\beta}_*^{[\iota-1]}), \bar{\mathbf{w}}_*^{[\iota]}, \mathbf{z}(\boldsymbol{\beta}_*^{[\iota-1]})\right\}$;
  2. Solve $\mathbb{Q}(\bar{\mathbf{w}}_*^{[\iota]})$ for $\left\{\mathbf{w}_*^{[\iota]}, \mathbf{z}_*^{[\iota]}, \boldsymbol{\beta}_*^{[\iota]}\right\}$;
  if $\iota > 1$ *and* $\mathbf{c}^\top \mathbf{z}_*^{[\iota]} \geq \mathbf{c}^\top \mathbf{z}_*^{[\iota-1]} - \zeta$ **then**
    $conv \leftarrow 1$
    Else $\mathbf{v}_*^{[\iota]} \leftarrow \left\{\mathbf{x}_*^{[\iota]}, \mathbf{w}_*^{[\iota]}, \bar{\mathbf{w}}_*^{[\iota]}, \boldsymbol{\beta}_*^{[\iota]}, \mathbf{z}_*^{[\iota]}\right\}$, $\iota \leftarrow \iota + 1$;
  end
end

the steps. We use the superscript $^{[\iota]}$ to denote the iteration index. This procedure can be interpreted as follows. Using $\mathbb{P}(\boldsymbol{\beta}_\mathrm{f})$, a disturbance set $\mathcal{W}$ characterized by variables $\mathbf{x}$ is computed as a function of $\boldsymbol{\beta}_\mathrm{f}$. This computation involves selecting the disturbance sequences $\mathbf{w}(\boldsymbol{\beta}_\mathrm{f})$. Then using $\mathbb{Q}(\bar{\mathbf{w}}_*)$, these disturbance sequences are updated to $\mathbf{w}_*$ by optimizing over $\boldsymbol{\beta}$, while the sets $\mathcal{W}^1_{[it]}$ and $\mathcal{W}^2_{[i]}$ defined in (5.62) and characterized by $\bar{\mathbf{w}}_*$ are kept fixed.

Proposition 5.4. *Algorithm 4 terminates in finite time for any $\zeta > 0$ with feasible iterates.* □

Proof. For any $\iota > 0$, since $(\mathbf{w}(\boldsymbol{\beta}_*^{[\iota-1]}), \mathbf{z}(\boldsymbol{\beta}_*^{[\iota-1]}))$ computed by $\mathbb{P}(\boldsymbol{\beta}_*^{[\iota-1]})$ are feasible for $\mathbb{Q}(\bar{\mathbf{w}}_*^{[\iota]})$, whose optimizers $(\mathbf{w}_*^{[\iota]}, \mathbf{z}_*^{[\iota]})$ are in turn feasible for $\mathbb{P}(\boldsymbol{\beta}_*^{[\iota]})$, the inequalities

$$\mathbf{c}^\top \mathbf{z}(\boldsymbol{\beta}_*^{[\iota-1]}) \geq \mathbf{c}^\top \mathbf{z}_*^{[\iota]} \geq \mathbf{c}^\top \mathbf{z}(\boldsymbol{\beta}_*^{[\iota]}) \geq \mathbf{c}^\top \mathbf{z}_*^{[\iota+1]} \tag{5.68}$$

hold, such that $\mathbf{c}^\top \mathbf{z}(\boldsymbol{\beta}_*^{[\iota]})$ and $\mathbf{c}^\top \mathbf{z}_*^{[\iota]}$ are nonincreasing in $[\iota]$. By construction of Problem (5.66), we know that $\mathbf{c}^\top \mathbf{z} \geq 0$ for all feasible $\mathbf{z}$. Thus, $\mathbf{c}^\top \mathbf{z}(\boldsymbol{\beta}_*^{[\iota]})$ and $\mathbf{c}^\top \mathbf{z}_*^{[\iota]}$ are bounded below, such that for every $\zeta > 0$, there exists some $\iota < \infty$ such that $\mathbf{c}^\top \mathbf{z}_*^{[\iota]} \geq \mathbf{c}^\top \mathbf{z}_*^{[\iota-1]} - \zeta$ holds, concluding the proof of finite termination. Feasibility of the iterates $\mathbf{v}_*^{[\iota]}$ follows by noting that $\mathbb{P}(\boldsymbol{\beta}_*^{[\iota-1]})$ and $\mathbb{Q}(\bar{\boldsymbol{w}}_*^{[\iota]})$ enforce the same constraints over $\mathbf{v}_*^{[\iota]}$ as Problem (5.66). $\square$

5.3.2 Zonotopes-based LP approximation

In this approach, we parameterize the disturbance set $\mathcal{W}$ as a convex hull of zonotopes as

$$\mathcal{W} = \text{ConvHull}\left(\mathbb{W}(\bar{w}_{[j]}, \bar{c}_{[j]}),\ j \in \mathbb{I}_1^N\right), \tag{5.69a}$$

$$\mathbb{W}(\bar{w}_{[j]}, M_{[j]}) = \bar{w}_{[j]} \oplus M_{[j]}\text{diag}(\bar{c}_{[j]})\mathbb{B}_\infty^{k_{[j]}}, \tag{5.69b}$$

with parameters $\{\bar{w}_{[j]} \in \mathbb{R}^{n_w}, \bar{c}_{[j]} \in \mathbb{R}^{k_{[j]}},\ \forall j \in \mathbb{I}_1^N\}$, where $M_{[j]} \in \mathbb{R}^{n_w \times k_{[j]}}$ is some user-specified matrix defined with unit column vectors. This parameterization is adopted from [47], in which it was shown that $\mathfrak{w} \in \mathbb{W}(\bar{w}_{[j]}, \bar{c}_{[j]})$ if

$$\exists\ \mathfrak{q} \in [-\bar{c}_{[j]}, \bar{c}_{[j]}] \qquad : \qquad \mathfrak{w} = \bar{w}_{[j]} + M_{[j]}\mathfrak{q}. \tag{5.70}$$

Moreover, from basic properties of support functions and (5.2) the support function over $\mathbb{W}(\bar{w}_{[j]}, \bar{c}_{[j]})$ for any $\mathbf{T} \in \mathbb{R}^{m \times n_w}$ and $\mathbf{p} \in \mathbb{R}^m$ is given by

$$h_{\mathbf{T}\mathbb{W}(\bar{w}_{[j]}, \bar{c}_{[j]})}(\mathbf{p}) = \mathbf{p}^\top \mathbf{T}\bar{w}_{[j]} + |\mathbf{p}^\top \mathbf{T} M_{[j]}\text{diag}(\bar{c}_{[j]})|\mathbf{1}, \tag{5.71}$$

such that from Proposition 5.2, the support function over $\mathcal{W}$ is

$$h_{\mathbf{T}\mathcal{W}}(\mathbf{p}) = \max_{j \in \mathbb{I}_1^N} \{\mathbf{p}^\top \mathbf{T}\bar{w}_{[j]} + |\mathbf{p}^\top \mathbf{T} M_{[j]}\text{diag}(\bar{c}_{[j]})|\mathbf{1}\}. \tag{5.72}$$

We now formulate Problem (5.43) for the disturbance set parameterization in (5.69) by exploiting (5.70) and (5.72). We first encode Constraint (5.43b) , i.e.,

$$\mathcal{O}(s, \alpha, \lambda, \mathcal{W}) \subseteq \mathcal{Y},$$

in the same vein as (5.55)- (5.56) by exploiting Equation (5.72). In particular, introducing $\{\mathbf{Q}_{[t]} \in \mathbb{R}^{m_\mathcal{Y}}, t \in \mathbb{I}_0^{s-1}\}$ and $\mathbf{r} \in \mathbb{R}^{m_\mathcal{Y}}$, we encode the inclusion using the linear inequalities

$$\begin{aligned}
&\forall\, j \in \mathbb{I}_1^N, \;\; \left\{ \begin{array}{l} \bar{G}_{[t]} B \bar{w}_{[j]} + |\bar{G}_{[t]} B M_{[j]} \mathrm{diag}(\bar{c}_{[j]})| \mathbf{1} \leq \mathbf{Q}_{[t]}, \\ G D \bar{w}_{[j]} + |G D M_{[j]} \mathrm{diag}(\bar{c}_{[j]})| \epsilon^w_{[j]} \leq \mathbf{r}, \end{array} \right. \\
&\forall\, t \in \mathbb{I}_0^{s-1}
\end{aligned} \tag{5.73a}$$

$$\sum_{t=0}^{s-1} \mathbf{Q}_{[t]i} + \mathbf{r}_i \leq g_i - \lambda \sum_{t=0}^{s-1} |\bar{G}_{[t]i}| \mathbf{1}, \qquad \forall\, i \in \mathbb{I}_1^{m_\mathcal{Y}}. \tag{5.73b}$$

We recall that matrices $G_{[t]}$ are defined in Equation (5.44). Then, we encode Constraint (5.43c), i.e., $B\mathcal{W} \subseteq \gamma \mathcal{B}_\infty^{n_x}$, by exploiting Equation (5.72), similarly to (5.57) as

$$\tilde{\mathbf{I}}_{n_x} B \bar{w}_{[j]} + |\tilde{\mathbf{I}}_{n_x} B M_{[j]} \mathrm{diag}(\bar{c}_{[j]})| \mathbf{1} \leq \gamma \mathbf{1}, \qquad \forall\, j \in \mathbb{I}_1^N. \tag{5.74}$$

To encode Constraint (5.43d), i.e., $\mathbf{0} \in \mathcal{W}$, we enforce that $\mathbf{0} \in \mathbb{W}(\bar{w}_{[1]}, \bar{c}_{[1]})$ for simplicity, that holds as per (5.70) if

$$\exists\;\; \mathfrak{q}_0 \in [-\bar{c}_{[1]}, \bar{c}_{[1]}] \qquad : \qquad \mathbf{0} = \bar{w}_{[1]} + M_{[1]} \mathfrak{q}_0. \tag{5.75}$$

Finally, to encode Constraint (5.43e), i.e., $\mathcal{Y} \subseteq \mathcal{S}(l, \mathcal{W}) \oplus \mathbb{B}(\epsilon)$, we adopt Proposition 5.1(e), similarly to (5.60). However, in order to derive an LP approximation of Problem (5.43), we fix $N = v_\mathcal{Y}$, i.e., the number of zonotopes parameterizing the disturbance set is equal to the number of vertices of $\mathcal{Y}$, and encode Constraint (5.43e) using the sufficient conditions

$$\forall\, i \in \mathbb{I}_1^{v_\mathcal{Y}}, \; \exists \left\{ \begin{array}{l} \left\{ \{ \mathrm{w}^1_{[it]},\; t \in \mathbb{I}_0^{l-1} \},\; \mathrm{w}^2_{[i]} \right\} \in \mathbb{W}(\bar{w}_{[i]}, \bar{c}_{[i]}), \\ \mathrm{b}_{[i]} \in \mathbb{B}(\epsilon), \end{array} \right. \tag{5.76a}$$

$$\text{such that } \mathrm{y}_{[i]} = \sum_{t=0}^{l-1} C A^{l-1-t} B \mathrm{w}^1_{[it]} + D \mathrm{w}^2_{[i]} + \mathrm{b}_{[i]}. \tag{5.76b}$$

This encoding approximates the conditions in (5.60), by replacing the inclusions $\{\{\mathrm{w}^1_{[it]},\; t \in \mathbb{I}_0^{l-1}\},\; \mathrm{w}^2_{[i]}\} \in \mathcal{W}$ with $\{\{\mathrm{w}^1_{[it]},\; t \in \mathbb{I}_0^{l-1}\},\; \mathrm{w}^2_{[i]}\} \in \mathbb{W}(\bar{w}_{[i]}, \bar{c}_{[i]})$. Exploiting (5.70), these inclusions hold if

$$\begin{aligned}
&\forall\, i \in \mathbb{I}_1^{v_\mathcal{Y}}, \;\; \forall\, t \in \mathbb{I}_0^{l-1}, \;\; \exists\; \mathfrak{q}^1_{[it]},\; \mathfrak{q}^2_{[i]} \in [-\bar{c}_{[i]}, \bar{c}_{[i]}] \\
&\quad :\;\; \mathrm{w}^1_{[it]} = \bar{w}_{[i]} + M_{[i]} \mathfrak{q}^1_{[it]}, \qquad \mathrm{w}^2_{[i]} = \bar{w}_{[i]} + M_{[i]} \mathfrak{q}^2_{[i]}.
\end{aligned} \tag{5.77}$$

Thus, for the disturbance set parameterization in (5.69) with $N = v_{\mathcal{Y}}$, Problem (5.43) can be approximated as the LP

$$\min_{\hat{\mathbf{z}}} \|\epsilon\|_1 \tag{5.78a}$$

$$\text{s.t. } (5.73),\ (5.74),\ (5.75),\ (5.76b),\ (5.77), \tag{5.78b}$$

$$\mathrm{b}_{[i]} \in \mathbb{B}(\epsilon),\ \forall\, i \in \mathbb{I}_1^{v_{\mathcal{Y}}}, \tag{5.78c}$$

with the optimization variables

$$\hat{\mathbf{z}} := \left\{ \begin{array}{c} \{\bar{w}_{[j]}, \bar{c}_{[j]}, j \in \mathbb{I}_1^{v_{\mathcal{Y}}}\},\ \{\mathbf{Q}_{[t]}, \mathbf{r}, t \in \mathbb{I}_0^{s-1}\},\ \mathfrak{q}_0, \epsilon, \\ \{\mathrm{w}^1_{[it]}, \mathfrak{q}^1_{[it]}, \mathrm{w}^2_{[i]}, \mathfrak{q}^2_{[i]}, \mathrm{b}_{[i]}, i \in \mathbb{I}_1^{v_{\mathcal{Y}}}, i \in \mathbb{I}_0^{l-1}\} \end{array} \right\}.$$

We present two numerical examples in this section, with the SOCP computations in Algorithm 3 performed using MOSEK [93], and LP computations in Algorithm 4 performed using Gurobi [51]. We use the output of Algorithm 4 to initialize the Primal-Dual Interior Point solver IPOPT [151] to solve Problem (5.66). Finally, the MPT-toolbox [53] was used to plot the sets. The computations were performed on a laptop with an Intel i7-7500U processor and 16GB of RAM running MATLAB R2017b on Ubuntu 16.04.

5.3.3 Simple Illustrative Example

We consider the randomly generated system

$$A = \begin{bmatrix} -0.5844 & -0.2378 & -0.2015 \\ -0.2378 & 0.0368 & 0.6915 \\ -0.2015 & 0.6915 & -0.0162 \end{bmatrix},\ B = \begin{bmatrix} 0 & 0.8974 \\ 0 & -1.8597 \\ 0.8903 & 0.9479 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 2.0091 & -0.1402 \\ -0.9894 & 0 & 1.1447 \end{bmatrix},\quad D = \begin{bmatrix} -0.8078 & 0 \\ 0.9676 & 0.6751 \end{bmatrix},$$

with output constraint set $\mathcal{Y} = \{y : Gy \leq \mathbf{1}\}$ defined with

$$G = \begin{bmatrix} -0.4489 & -1.9691 & 1.0364 & 1.4018 & -0.9868 \\ 2.1848 & 1.2596 & 0.8726 & -0.3397 & -2.0995 \end{bmatrix}^\top.$$

We select $\mu = 10^{-3}$ and $\gamma = 0.2$ in Algorithm 3. Using the MOSEK [93] SOCP solver, we converge in 0.19s with $s = 59$, $\alpha = 6.789 \times 10^{-4}$, $\lambda =$

6.784×10^{-5}. Then, we select $N = 4$ boxes in $\mathbb{R}^2$ to parametrize $\mathcal{W}$ in (5.47), and use $l = s = 59$ to formulate (5.63). Finally, we choose a uniform 6-sided polytope in $\mathbb{R}^2$ to define $\mathbb{B}(\epsilon)$, using which we formulate Problem (5.66).

To solve Problem (5.66), we first implement Algorithm 4 with $\boldsymbol{\beta}$ initialized with $\beta^1_{[itj]} = 1/N$ and $\beta^2_{[ij]} = 1/N$. The algorithm terminates in 10 iterations with $\|\epsilon\|_1 = 1.0962$, and requires an average of 0.3792s per iteration. In Figure 6, we plot with thick blue lines the disturbance set and the corresponding output reachable set obtained at the termination of Algorithm 4. We also plot the values of $\|\epsilon\|_1$ computed over the iterations of Algorithm 4.

We use the output of the algorithm to initialize IPOPT to solve Problem (5.66). At termination, we obtain $\|\epsilon\|_1 = 1.055$. We plot the disturbance set along with the corresponding output reachable set solving Problem (5.66) in also Figure 6 as well. In order to solve Problem (5.66) efficiently with IPOPT, we observe that the only contribution to the Lagrangian Hessian stems from the bilinear constraints, which are nonzero only on off-diagonal blocks. Because IPOPT computes a positive-definite Hessian approximation by simply adding a positive diagonal matrix, we observed that it is beneficial to neglect the contribution to the Lagrangian Hessian of these bilinear terms. We therefore pass a zero Hessian approximation to IPOPT. In our experience, this modification permits IPOPT to quickly converge to a local minimum. For the current example, IPOPT converges in 281 iterations requiring 14.7236s. If instead the exact Lagrangian Hessian is passed, IPOPT converges with $\|\epsilon\|_1 = 1.0826$ in 994 iterations requiring 69.1807s. We compare our results with those obtained by tackling Problem (3.7) using the Explicit RPI (ERPI) set approach in Chapter 3. In this approach, the mRPI set is approximated with an RPI set parameterized with fixed normal vectors, and the approach computes an explicit representation of the RPI set along with a disturbance set also parameterized with fixed normal vectors. We select 163 hyperplanes to parameterize the RPI set, and 16 hyperplanes to parameterize the disturbance set. At the termination of the procedure in Chapter 3, we obtain $\|\epsilon\|_1 = 1.2039$, demonstrating that the implicit RPI

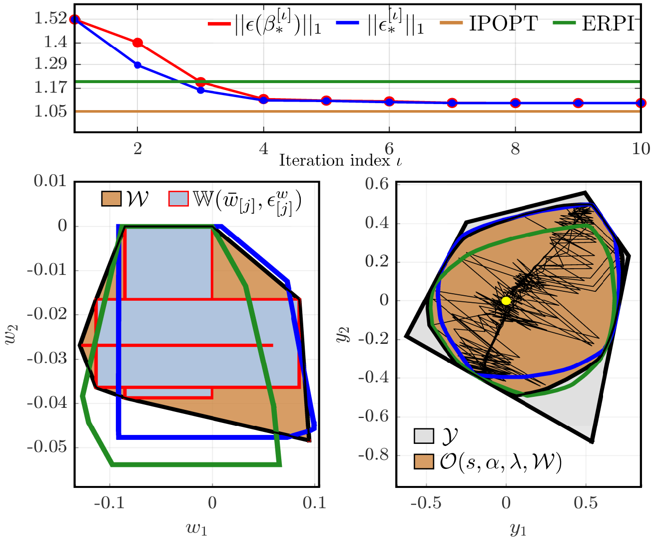


Figure 6: (Top) Convergence of Algorithm 4 : The inequalities in (5.68) hold, and the algorithm converges at $\iota = 10$ with $||\epsilon||_1 = 1.0962$; (Bottom-Left) The disturbance set $\mathcal{W}$, along with boxes $\mathbb{W}(\bar{w}_{[j]}, \epsilon^w_{[j]})$ composing $\mathcal{W}$ in (5.47); (Bottom-Right) Output constraint set $\mathcal{Y}$ and the output-reachable set $\mathcal{O}(s, \alpha, \lambda, \mathcal{W})$. Black lines are the output trajectories $y(t)$ of the system when initialized with $x(0) = \mathbf{0}$ (Yellow dot) and subject to random inputs $w \in \mathcal{W}$. As expected, we observe that $y \in \mathcal{O}(s, \alpha, \lambda, \mathcal{W}) \subset \mathcal{Y}$ always holds. The thick blue lines indicate the solution at the termination of Algorithm 4, and think green lines indicate the sets computed using the ERPI approach in Chapter 3.

set approach proposed in the current chapter can compute disturbance sets with reduced conservativeness. We plot these results in Figure 6. While the solution of the ERPI approach can potentially be refined further by increasing the number of hyperplanes parameterizing the RPI set, it would result in greatly increased computational expense. This is because the number of variables and constraints in the ERPI approach increase quadratically with the number of hyperplanes parameterizing the RPI set. We now study the effect of number of boxes N parameterizing the disturbance set $\mathcal{W}$ in (5.47) on the procedure to solve Problem (5.66). To this end, we parameterize $\mathcal{W}$ with number of boxes rang-

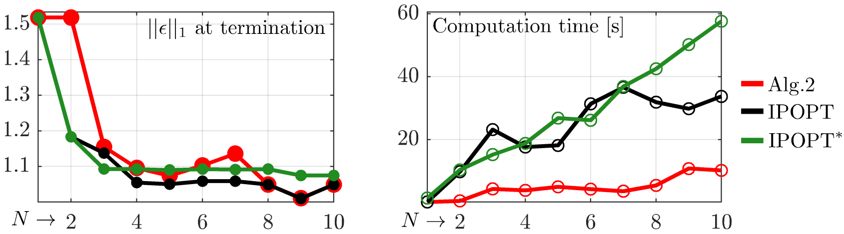


Figure 7: Effect of number of boxes N parameterizing the disturbance set $\mathcal{W}$ on Problem (5.66). The nonmonotonicity of $\|\epsilon\|_1$ is because of the nonlinear nature of the problem. However, as expected in general smaller values of $\|\epsilon\|_1$ are obtained as N increases. Red lines indicate solution of IPOPT when initialized with output of Algorithm 4, and green lines indicate solution with initial guess all-zero. Observe that smaller values of $\|\epsilon\|_1$ are computed if IPOPT is initialized with the solution of Algorithm 4.

ing from $N = 1$ till $N = 10$. For each N, we first simulate Algorithm 4 with $\beta^1_{[itj]} = 1/N$ and $\beta^2_{[ij]} = 1/N$. Then, we use the output of Algorithm 4 to initialize IPOPT to solve Problem (5.66). In Figure 7, we plot the values of $\|\epsilon\|_1$ obtained at the termination of Algorithm 4 and IPOPT. We also plot the total CPU time required for each procedure. We observe that as expected, the optimal value of $\|\epsilon\|_1$ reduces in general as the number of boxes N used to parameterize $\mathcal{W}$ increases. However, this trend is not strictly monotonic, owing to the nonlinear nature of the optimization procedure, which converges to a local minimum. We also observe that Algorithm 4 terminates in the majority of cases in under 10s, while IPOPT requires around 30s for larger values of N in addition to the time required for initialization using Algorithm 4. From this observation, we infer that if only a reasonably good solution of Problem (5.66) instead of minimizer is desired, then the output of Algorithm 4 can be considered as a viable solution. In Figure 7, we also plot the value of $\|\epsilon\|_1$ and the correspond computation time for IPOPT if initialized with an all-zero vector (shown in green). We observe that the values of $\|\epsilon\|_1$ are larger than when initialized with the output of Algorithm 4, motivating the use of a good initial guess to initialize IPOPT.

Finally, we apply the heuristics proposed in Section 5.3 to approximate Problem (5.43) as an LP. The resulting solutions are plotted in Figure 8. The solution indicated as Approximation 1 is obtained by solving LP $\mathbb{P}(\boldsymbol{\beta})$ with $N = v_{\mathcal{Y}} = 5$ boxes to parameterize the disturbance set $\mathcal{W}$ in (5.47), and the components of $\boldsymbol{\beta}$ are fixed as in Equation (5.67). The solution corresponds to $\|\epsilon\|_1 = 1.2058$, and the LP is solved in 0.4751s. The solution indicated as Approximation 2 is obtained by solving the LP in (5.78) with $N = v_{\mathcal{Y}} = 5$ zonotopes parameterizing the disturbance set $\mathcal{W}$ in (5.69). For each $j \in \mathbb{I}_1^N$, we fix matrices $M_{[j]} \in \mathbb{R}^{2\times 40}$ with columns uniformly sampled from $\mathcal{B}_2^2$. The solution corresponds to $\|\epsilon\|_1 = 1.0521$, and the LP is solved in 145.2394s. The large increase in computational time in the case of Approximation 2 is attributed to the fact that the disturbance set is parameterized by $v_{\mathcal{Y}} n_w + \sum_{j=1}^{v_{\mathcal{Y}}} k_{[j]} = 210$ parameters, while $\mathcal{W}$ is parameterized with $2v_{\mathcal{Y}} n_w = 20$ in the case of Approximation 1. For comparison, we fix $N = 9$ boxes to parameterize the disturbance set, and solve Problem (5.66) using IPOPT (initialized with the solution of Algorithm 4). The results obtained at termination are plotted with thick black lines in Figure 8, and they correspond to $\|\epsilon\|_1 = 1.0103$. This solution is computed in 40.5979s. From this observation, we infer that while LP (5.78) can be used to compute disturbance sets with reduced conservativeness that LP $\mathbb{P}(\boldsymbol{\beta})$, the increased computational time motivates the use of Algorithm 4 along with an NLP solver such as IPOPT to solve Problem (5.66) in practice.

5.3.4 Comparison against the ERPI approach of Chapter 3

We compare the performance of the method proposed in this chapter using Implicit RPI sets (IRPI) with the Explicit RPI (ERPI) approach proposed in Chapter 3 to tackle Problem (3.7) over 24 randomly generated systems with dimensions shown in Table 5. The values of $\rho(A)$ of the systems are plotted in Figure 9-(Top). For each example, we use the output constraint set $\mathcal{Y} = \mathcal{B}_\infty^{n_y}$ and matrix $H = [\mathbf{I}_{n_y} \ -\mathbf{I}_{n_y}]^\top$ to define $\mathbb{B}(\epsilon)$.

For our IRPI approach, we first use Algorithm 3 to compute the parameters (s, α, λ) that parameterize the RPI set $\mathcal{R}(s, \alpha, \lambda, \mathcal{W})$ in Equa-

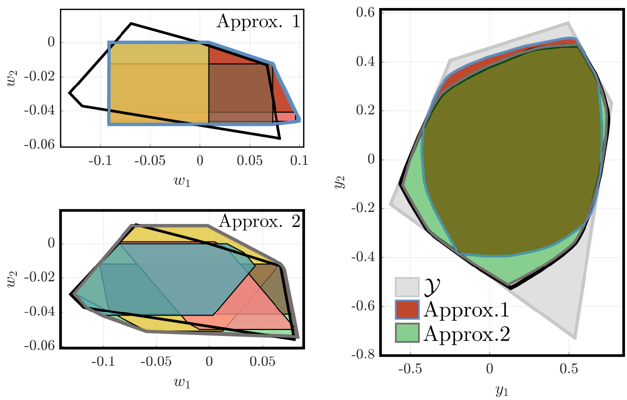


Figure 8: LP approximations of Problem (5.43). Approximation 1 corresponds to the solution of $\mathbb{P}(\boldsymbol{\beta})$ with $\boldsymbol{\beta}$ chosen as (5.67). Approximation 2 is obtained by solving LP (5.78) with $k_{[j]} = 40$. The black sets are obtained by solving Problem (5.66) with $N = 9$ boxes.

tion (5.8) with $\mu = 10^{-2}$ and $\gamma = 1$. The values of these parameters s are shown in Table 5, and (α, λ) are plotted in Figure 9-(Top). The average runtime of Algorithm 3 over all 24 examples in 0.3411s. Then, we set $N = 5$ number of boxes to parameterize the disturbance set in (5.47), and select $l = s$ to parameterize the set $\mathcal{S}(l, \mathcal{W})$ formulating Constraint (5.43e). To solve the resulting optimization problem (5.66), we first run Algorithm 4. We then use the output of the algorithm to initialize IPOPT to solve problem (5.66). We label the total runtime of Algorithm 4 and IPOPT to solve problem (5.66) as $\mathrm{t}_{\mathrm{IRPI}}$, and the objective value as $\|\epsilon_{\mathrm{IRPI}}\|_1$.

Regarding the ERPI approach, we recall that it uses an RPI set parameterized as the polytope

$$\mathcal{X}_{\mathrm{RPI}}(\mathcal{W}) := \{x : E_i x \leq \epsilon_i^x, \ i \in \mathbb{I}_1^{m_X}\}$$

to approximate the mRPI set $\mathcal{X}_{\mathrm{m}}(\mathcal{W})$ in Problem (3.7), with the normal vectors $\{E_i^\top, i \in \mathbb{I}_1^{m_X}\}$ fixed a priori. It also considers a disturbance set parameterized as the polytope

$$\mathcal{W} := \{w : F_i x \leq \epsilon_i^w, \ i \in \mathbb{I}_1^{m_W}\}$$

#	$(n_x, n_w, n_y, s, m_X, m_W)$	#	$(n_x, n_w, n_y, s, m_X, m_W)$
1	$(2, 2, 2, 8, 12, 6)$	2	$(2, 2, 3, 15, 12, 6)$
3	$(2, 3, 2, 12, 18, 42)$	4	$(2, 3, 3, 2, 18, 42)$
5	$(2, 4, 2, 9, 20, 80)$	6	$(2, 4, 3, 17, 8, 80)$
7	$(3, 2, 2, 13, 34, 6)$	8	$(3, 2, 3, 14, 34, 6)$
9	$(3, 3, 2, 16, 36, 42)$	10	$(3, 3, 3, 16, 35, 42)$
11	$(3, 4, 2, 7, 34, 80)$	12	$(3, 4, 3, 11, 34, 80)$
13	$(4, 2, 2, 18, 68, 6)$	14	$(4, 2, 3, 4, 8, 6)$
15	$(4, 3, 2, 18, 74, 42)$	16	$(4, 3, 3, 10, 75, 42)$
17	$(4, 4, 2, 7, 80, 80)$	18	$(4, 4, 3, 13, 53, 80)$
19	$(5, 2, 2, 7, 67, 6)$	20	$(5, 2, 3, 6, 20, 6)$
21	$(5, 3, 2, 5, 72, 42)$	22	$(5, 3, 3, 6, 102, 42)$
23	$(5, 4, 2, 5, 118, 80)$	24	$(5, 4, 3, 7, 108, 80)$

Table 5: Randomly generated systems

#	$(n_x, n_w, n_y, \rho(A), s, \|\epsilon_{\mathrm{IRPI}}\|_1, \mathrm{t}_{\mathrm{IRPI}}[\mathrm{s}], \mu_{\mathrm{IRPI}})$
1	$(10, 5, 2, 0.6975, 40, 0.5166, 45.4723, 0.0073)$
2	$(20, 10, 3, 0.69, 38, 1.0529, 258.9705, 0.0062)$
3	$(50, 20, 4, 0.699, 42, 1.3626, 1.1432 \times 10^4, 0.0064)$
4	$(100, 20, 2, 0.8, 76, 0.3093, 197.892, 0.0061)$

Table 6: Higher-dimensional systems

with normal vectors $\{F_i^\top, i \in \mathbb{I}_1^{m_W}\}$ also fixed a priori. The dimensions (m_X, m_W) are shown in Table 5. We solve the optimization problem resulting from the ERPI approach using the specialized smoothening-based interior-point algorithm presented in Section 3.4. We label the run-time of this algorithm as $\mathrm{t}_{\mathrm{ERPI}}$, and the objective value as $\|\epsilon_{\mathrm{ERPI}}\|_1$.

In Figure 9-(Bottom) , we plot the ratios of objective values $\|\epsilon\|_1$, and the solution times. We observe that our IRPI approach computes much smaller values of $\|\epsilon\|_1$ than the ERPI approach, in a computation time that is significantly smaller. The smaller values of $\|\epsilon\|_1$ yielded by the IRPI approach stem from the fact that we do not enforce a specific representation of the RPI set a priori, while the reduced computational time stems from the fact that the convex hull of boxes adopted for the distur-

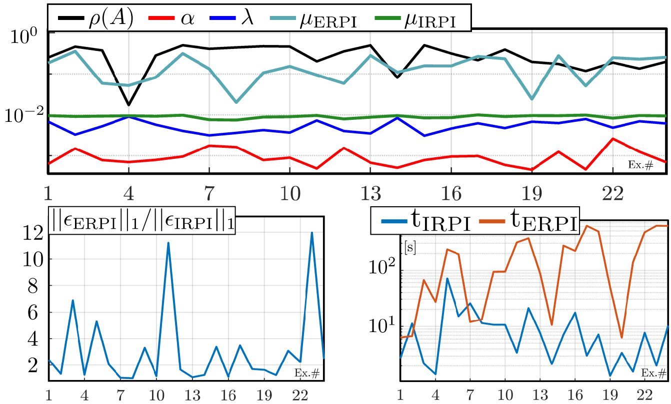


Figure 9: Comparison against the ERPI approach. The Implicit RPI approach tackles Problem (3.7) with reduced conservativeness and improved computational efficiency.

bance set $\mathcal{W}$ in (5.47) allows very cheap evaluations of the support functions. We remark that the smallest value of $\|\epsilon_{\mathrm{ERPI}}\|_1 \,/\, \|\epsilon_{\mathrm{IRPI}}\|_1$ is 1.0042 obtained for Example 8.

At the output of the ERPI approach, we construct the sets $\mathcal{X}_{\mathrm{RPI}}(\mathcal{W})$, and compute their approximation error with respect to the mRPI set. In other words, we compute the smallest values of μ_{ERPI} that verify $\mathcal{X}_{\mathrm{RPI}}(\mathcal{W}) \subseteq \mathcal{X}_{\mathrm{m}}(\mathcal{W}) \oplus \mu_{\mathrm{ERPI}}\mathcal{B}_{\infty}^{n_x}$. We plot these values in Figure 9-(Top), in which we observe that the ERPI set approximation error is larger than $\mu = 10^{-2}$ used for our IRPI approach. We also recompute the IRPI set approximation error μ_{IRPI}, i.e., the smallest value of μ_{IRPI} verifying $\mathcal{R}(s,\alpha,\lambda,\mathcal{W}) \subseteq \mathcal{X}_{\mathrm{m}}(\mathcal{W}) \oplus \mu_{\mathrm{IRPI}}\mathcal{B}_{\infty}^{n_x}$. As expected, these values satisfy $\mu_{\mathrm{IRPI}} \leq \mu = 10^{-2}$ over all the examples. While the RPI approximation error μ_{ERPI} can potentially be reduced and brought close to μ_{IRPI} by choosing a larger m_X thus reducing conservativeness of the ERPI approach, it will result in an increased computational complexity. On the other hand, the approach proposed in this chapter computes less conservative solutions with much lower computational complexity, thus demonstrating the efficacy of using IRPI sets to tackle Problem (3.7).

Finally, we use our IRPI approach to tackle Problem (3.7) for higher dimensional systems. Because of the large values of n_x, the large number of hyperplanes m_X required to represent an RPI set in the ERPI approach results in the system running out of memory, thus effectively failing to tackle Problem (3.7). On the other hand, the IRPI approach succeeds in tackling Problem (3.7) The system details and the results of the IRPI approach are shown in Table 6, in which we select the parameters $(\mu, \gamma, \mathcal{Y}, H, N, l)$ in the same way as for the examples in Table 5. We observe that despite being successful, the computation time of the IRPI approach scales poorly with dimension of the output constraint set $\mathcal{Y}$. This is because the number of vertices $v_{\mathcal{Y}} = 2^{n_y}$ leads in an exponential increase in the number of variables and constraints defining Problem (5.66) (can be observed from Tables 3-4) with n_y. This issue can be tackled if we can encode the inclusion $\mathcal{Y} \subseteq \mathcal{S}(l, \mathcal{W}) \oplus \mathbb{B}(\epsilon)$ directly using the hyperplane representation of $\mathcal{Y}$. The development of such techniques is the subject of future research.

5.3.5 Dynamics decoupling for reduced-order MPC

We consider an LTI system given by

$$\begin{bmatrix} x_{[1]}(t+1) \\ x_{[2]}(t+1) \end{bmatrix} = \begin{bmatrix} A_{[11]} & A_{[12]} \\ A_{[21]} & A_{[22]} \end{bmatrix} \begin{bmatrix} x_{[1]}(t) \\ x_{[2]}(t) \end{bmatrix} + \begin{bmatrix} B_{[1]} \\ \mathbf{0} \end{bmatrix} u(t), \tag{5.79a}$$

$$y(t) = C_{[1]} x_{[1]}(t) + C_{[2]} x_{[2]}(t), \tag{5.79b}$$

that is subject to input and output constraints $u \in \mathcal{U}$ and $y \in \mathcal{Y}$ respectively. For this system, we aim to design an MPC controller that satisfies the following requirements:

1. Regulate the substate $x_{[1]}$ with inputs $u \in \mathcal{U}$ with no access to substate $x_{[2]}$ and output y, and no knowledge of matrices
$$(A_{[21]}, A_{[22]}, C_{[1]}, C_{[2]}).$$
2. Satisfy given output constraints $y \in \mathcal{Y}$.

We make the following assumptions on System (5.79).

Assumption 5.3. *(a) $\rho(A_{[22]}) < 1$; (b) The output constraint set $\mathcal{Y}$ is a polytope containing the origin in its interior.*

Remark 5.2. *These design requirements are different from the ones tackled in other Reduced-Order MPC schemes, e.g., [137], in which the constraints are not coupled, and $(A_{[21]}, A_{[22]})$ are used to perform set-membership estimation of $x_{[2]}$.* $\square$

Since we are only interested in regulating the substate $x_{[1]}$ of System (5.79a), we write its dynamics separately as

$$x_{[1]}(t+1) = A_{[11]}x_{[1]}(t) + B_{[1]}u(t) + A_{[12]}x_{[2]}(t). \tag{5.80}$$

In the sequel, we treat System (5.80) as an uncertain LTI plant with additive disturbance $x_{[2]}$, for which we design a robust MPC (RMPC) controller that satisfies the aforementioned requirements. To this end, we make the following observations.

We first note that substate $x_{[2]}$ depends on substate $x_{[1]}$ as

$$x_{[2]}(t+1) = A_{[22]}x_{[2]}(t) + A_{[21]}x_{[1]}(t). \tag{5.81}$$

Then, given some compact convex set $\mathbb{X}_{[1]}$ containing the origin, if the substate $x_{[1]}$ satisfies

$$x_{[1]}(t) \in \mathbb{X}_{[1]}, \qquad\qquad \forall\, t \geq 0, \tag{5.82}$$

then the substate $x_{[2]}$ always belongs to the mRPI set $\mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$ of System (5.81) with disturbance set $\mathbb{X}_{[1]}$, i.e.,

$$x_{[2]}(t) \in \mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]}) := \bigoplus_{t=0}^{\infty} A_{[22]}^{t} A_{[21]}\mathbb{X}_{[1]}, \qquad\qquad \forall\, t \geq 0, \tag{5.83}$$

if $x_{[2]}(0) \in \mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$. Note that under Assumption 5.3(a) the mRPI set $\mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$ is compact, convex and unique [63]. Inclusion (5.83) implies that if System (5.80) is constrained as $x_{[1]} \in \mathbb{X}_{[1]}$, then the additive disturbance acting on the system always satisfies $x_{[2]} \in \mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$. Based on this observation, we propose to design a tube-based RMPC controller [87] for System (5.80) to robustly satisfy the state-constraints $x_{[1]} \in \mathbb{X}_{[1]}$ in the presence of additive disturbances $x_{[2]} \in \mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$. Since this controller does not require measurements of $x_{[2]}$ for its implementation, we satisfy Requirement (1).

Now, we focus on designing the state-constraint set $\mathbb{X}_{[1]}$. To this end, we formulate conditions on $\mathbb{X}_{[1]}$ to satisfy Requirements (1)-(2). We note that System (5.79) must satisfy

$$y(t) = C_{[1]}x_{[1]}(t) + C_{[2]}x_{[2]}(t) \in \mathcal{Y}, \qquad \forall\, t \geq 0 \tag{5.84}$$

as per Equation (5.79b). Then, since $x_{[1]} \in \mathbb{X}_{[1]}$ implies that $x_{[2]} \in \mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$, it is sufficient to enforce the inclusion

$$\mathcal{Y}_{\mathrm{m}}(\mathbb{X}_{[1]}) := C_{[1]}\mathbb{X}_{[1]} \oplus C_{[2]}\mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]}) \subseteq \mathcal{Y} \tag{5.85}$$

in order to satisfy (5.84), such that Requirement (2) is satisfied. The necessity of inclusion (5.85) follows from Requirement (1), according to which the inclusion in (5.84) must hold for all $x_{[2]} \in \mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$ since the controller cannot access measurements of $(x_{[2]}(t), y(t))$. Thus, we propose to design a state constraint set $\mathbb{X}_{[1]}$ for System (5.80) satisfying inclusion (5.85).

Based on these observations, we propose the following 2-step procedure to synthesize an MPC controller satisfying the aforementioned requirements.

1. Design a set $\mathbb{X}_{[1]}$ satisfying inclusion (5.85).

2. Design an RMPC controller for System (5.80) with state constraint set $\mathbb{X}_{[1]}$, input constraint set $\mathcal{U}$ (given a priori) and disturbance set $x_{[2]} \in \mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$.

For Step (1) of this design procedure, we propose to solve the optimization problem

$$\min_{\mathbb{X}_{[1]}} \ \mathrm{d}_{\mathcal{Y}}(\mathcal{Y}_{\mathrm{m}}(\mathbb{X}_{[1]})) \tag{5.86a}$$

$$\text{s.t. } \mathcal{Y}_{\mathrm{m}}(\mathbb{X}_{[1]}) \subseteq \mathcal{Y}, \tag{5.86b}$$

$$\mathbf{0} \in \mathbb{X}_{[1]}, \tag{5.86c}$$

that computes a set $\mathbb{X}_{[1]}$ while enforcing inclusion (5.85) through constraint (5.86b), and minimizes the approximation error of the output constraint set $\mathcal{Y}$ by the output-reachable set $\mathcal{Y}_{\mathrm{m}}(\mathbb{X}_{[1]})$ through objective (5.86a). Observing that Problem (5.86) is analogous to Problem (3.7) with

$$\text{matrices } (A, B, C, D) \leftarrow (A_{[22]}, A_{[21]}, C_{[2]}, C_{[1]}), \tag{5.87a}$$
$$\text{disturbance set } \mathcal{W} \leftarrow \mathbb{X}_{[1]}, \tag{5.87b}$$

we tackle Problem (5.86) using the methods described in this chapter to compute an optimal state-constraint set $\mathbb{X}_{[1]}$.

For Step (2), we design a tube-based Robust MPC (RMPC) controller [87] to control System (5.80) with state constraint set $x_{[1]} \in \mathbb{X}_{[1]}$ obtained by solving Problem (5.86) and disturbance set $x_{[2]} \in \mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$ as defined in Equation (5.83). Note that in principle any robust MPC scheme can be used, e.g., [26, 71]. Since the disturbance set (i.e., the mRPI set $\mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$) cannot generally be explicity computed, we propose to use the μ-RPI set $\mathcal{R}(s, \alpha, \lambda, \mathbb{X}_{[1]})$ defined in Equation (5.8) for system matrices (5.87a) as the disturbance set in place of the mRPI set $\mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$. Since the μ-RPI set satisfies the inclusion

$$\mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]}) \subseteq \mathcal{D}(\mathbb{X}_{[1]}) := \mathcal{R}(s, \alpha, \lambda, \mathbb{X}_{[1]}),$$

an RMPC controller that is designed to be robust against the *larger* disturbance set $\mathcal{D}(\mathbb{X}_{[1]})$ is robust against the *smaller* disturbance set $\mathcal{X}_{\mathrm{m}}(\mathbb{X}_{[1]})$.

We will now briefly recap the tube-based RMPC design procedure from [87] for System (5.80) subject to constraints $x_{[1]} \in \mathbb{X}_{[1]}$ and $u \in \mathcal{U}$, and disturbances $x_{[2]} \in \mathcal{D}(\mathbb{X}_{[1]})$. Assuming to know a feedback gain $K_{[1]}$ such that matrix $A^K_{[11]} := A_{[11]} + B_{[1]}K_{[1]}$ is strictly stable, the RMPC design procedure involves splitting the state $x_{[1]}$ into *nominal* and *perturbed* components as $x_{[1]} = \hat{x}_{[1]} + \Delta x_{[1]}$, and parameterizing the control input as $u = \hat{u} + K_{[1]}\Delta x_{[1]}$, where the nominal and perturbed components satisfy the dynamics

$$\hat{x}_{[1]}(t+1) = A_{[11]}\hat{x}(t) + B_{[1]}\hat{u}(t), \tag{5.88a}$$
$$\Delta x_{[1]}(t+1) = A^K_{[1]}\Delta x_{[1]}(t) + A_{[12]}x_{[2]}(t). \tag{5.88b}$$

Then, an RPI set $\Delta\mathbb{X}_{[1]}$ is computed for the perturbed System (5.88b) with disturbances $x_{[2]} \in \mathcal{D}(\mathbb{X}_{[1]})$. This set satisfies the usual RPI inclusion

$$A_{[1]}^K \Delta\mathbb{X}_{[1]} \oplus A_{[12]}\mathcal{D}(\mathbb{X}_{[1]}) \subseteq \Delta\mathbb{X}_{[1]}$$

and can be computed using, e.g., [112, 111, 144]. It is then used to compute a control input

$$u(t) = \hat{u}_*(t) + K_{[1]}(x_{[1]}(t) - \hat{x}_{[1]*}(t)),$$

where $(\hat{u}_*(t), \hat{x}_{[1]*}(t))$ solve the following Quadratic Program (QP) parameterized by the current measured state $x_{[1]}(t)$:

$$\begin{aligned} \min_{\mathbf{x}_{\mathrm{MPC}}} \; & \sum_{s=t}^{t+L-1} \left\| \begin{bmatrix} \hat{x}_{[1]}(s) - x_{\mathrm{tgt}} \\ \hat{u}(s) - u_{\mathrm{tgt}} \end{bmatrix} \right\|_{S_{[1]}}^2 + \left\| \hat{x}_{[1]}(t+L) - x_{\mathrm{tgt}} \right\|_{P_{[1]}}^2 \\ \text{s.t.} \; & \hat{x}_{[1]}(s+1) = A_{[11]}\hat{x}_{[1]}(s) + B_{[1]}\hat{u}(s), && s \in \mathbb{I}_t^{t+L-1}, \\ & \hat{x}_{[1]}(s) \in \mathbb{X}_{[1]} \ominus \Delta\mathbb{X}_{[1]}, && s \in \mathbb{I}_{t+1}^{t+L-1}, \\ & \hat{u}(s) \in \mathcal{U} \ominus K_{[1]}\Delta\mathbb{X}_{[1]}, && s \in \mathbb{I}_{t+1}^{t+L-1}, \\ & x_{[1]}(t) - \hat{x}_{[1]}(t) \in \Delta\mathbb{X}_{[1]}, \\ & \hat{x}_{[1]}(t+L) - x_{\mathrm{tgt}} \in \mathcal{X}_{\mathrm{term}}, \end{aligned} \tag{5.89}$$

where $\mathbf{x}_{\mathrm{MPC}} := \{\hat{x}_{[1]}(t), \ldots, \hat{x}_{[1]}(t+L), \hat{u}(t), \ldots, \hat{u}(t+L-1)\},$, and $(x_{\mathrm{tgt}}, u_{\mathrm{tgt}})$ are the reference state and input to track. For details regarding the conditions that the terminal set $\mathcal{X}_{\mathrm{term}}$ and cost matrices $(S_{[1]}, P_{[1]})$ must satisfy to guarantee recursive feasibility and exponential stability, we refer the reader to [87]. Here we focus on the tightened constraints

$$\mathbb{X}_{[1]} \ominus \Delta\mathbb{X}_{[1]}, \qquad \mathcal{U} \ominus K_{[1]}\Delta\mathcal{X}_{[1]}$$

used in the formulation of QP (7.3). We note that these tightened constraints are nonempty if the inclusions

$$\Delta\mathbb{X}_{[1]} \subseteq \mathrm{interior}(\mathbb{X}_{[1]}), \qquad K_{[1]}\Delta\mathbb{X}_{[1]} \subseteq \mathrm{interior}(\mathcal{U}). \tag{5.90}$$

are verified. Hence, we make the following assumption to guarantee feasibility of Problem (7.3).

Assumption 5.4. *The inclusions in* (5.90) *are verified.* $\square$

We note that, since $\mathbb{X}_{[1]}$ is the optimization variable in Problem (5.86), inclusions (5.90) can in principle be appended as constraints over $\mathbb{X}_{[1]}$. However, encoding such constraints requires further results that will be a subject of future work. We remark that if Problem (5.86) is instead tackled using the ERPI approach in Chapter 3, then these inclusions can be appended as constraints with minor modifications, such that Assumption 5.4 would be satisfied by construction. Note that such inclusions were appended as constraints while formulating the DeMPC problem in Chapter 4, in particular, in Constraint (4.18b) as $\Delta\mathbb{X}_{[1]} \subseteq \phi_{\mathrm{x}}\mathbb{X}_{[1]}$, where $\phi_{\mathrm{x}} \in (0,1)$ is some user-specified design parameter.

As an example, we consider System (5.79) with matrices

$$A_{[11]} = \begin{bmatrix} 1.0000 & 1.0000 \\ 0 & 1.0000 \end{bmatrix}, \quad A_{[12]} = \begin{bmatrix} -0.0524 & -0.3299 & 0.3061 & 0.2773 \\ -0.0048 & -0.1020 & 0.1244 & -0.1044 \end{bmatrix},$$
$$A_{[21]} = \begin{bmatrix} 0 & 0.0204 \\ 0 & 0.0344 \\ 0 & -0.0339 \\ 0 & 0.0134 \end{bmatrix}, \quad A_{[22]} = \begin{bmatrix} -0.0790 & 0.2854 & -0.0377 & 0.6949 \\ 0.2854 & -0.2284 & 0.2752 & 0.3536 \\ -0.0377 & 0.2752 & 0.6021 & -0.2824 \\ 0.6949 & 0.3536 & -0.2824 & -0.0129 \end{bmatrix},$$
$$C_{[1]} = \begin{bmatrix} 0.9407 & -0.3282 \\ -0.6624 & -0.7257 \end{bmatrix}, \quad C_{[2]} = \begin{bmatrix} 0.8716 & 0.3587 & 0.2407 & 0.5116 \\ -0.1863 & 0.1624 & 0.7122 & 1.7494 \end{bmatrix},$$

$B_{[1]} = [0.5 \quad 1]^\top$, and subject to input-constraints

$$\mathcal{U} = \{u : \|u\|_\infty \leq 0.7\}.$$

For Step (1) of the design procedure to compute a state-constraint set $\mathbb{X}_{[1]}$, we solve Problem (5.86) using the methods proposed in this chapter. To this end, we set the system matrices as per (5.87a). Then, recollecting that we aim to compute a disturbance set $\mathcal{W}$ that we will use as the state-constraint set $\mathbb{X}_{[1]}$, we formulate Problem (5.43) using parameters $s = 150$, $\lambda = 2.931\times10^{-5}$, $\alpha = 5.195\times10^{-4}$ that are computed by Algorithm 3 for $\mu = 10^{-3}$ and $\gamma = 0.1$ in 0.44s. We parametrize the set $\mathbb{B}(\epsilon)$ used to define $\mathrm{d}_{\mathcal{Y}}(\cdot)$ in Equation (3.8) with $n_B = 8$ vectors $\{H_i^\top, i \in \mathbb{I}_1^{n_B}\}$ sampled uniformly from the surface of $\mathcal{B}_\infty^2$.

Finally, we parametrize the disturbance set in (5.47) with $N = 4$ boxes in $\mathbb{R}^2$, and select $l = s = 150$ to formulate constraint (5.43e). In order to

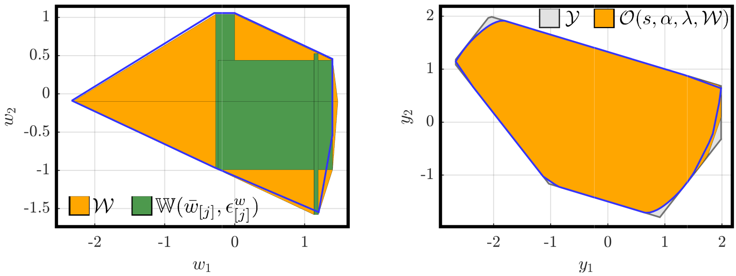


Figure 10: Results approximating Problem (5.86) as Problem (5.66) to compute $\mathbb{X}_{[1]} = \mathcal{W}$. The blue lines indicate the boundaries of corresponding sets obtained at termination of Algorithm 4.

solve the resulting Problem (5.66), we first implement Algorithm 4. The algorithm was initialized with each $\beta^1_{[itj]}, \beta^2_{[ij]} = 1/N$, and it terminates in $\iota = 16$ iterations with $\|\epsilon\|_1 = 0.9726$, consuming an average of 0.7655s per iteration. We then use the output of the algorithm to initialize IPOPT to solve Problem (5.66). At termination, we obtain $\|\epsilon\|_1 = 0.8828$. In Figure 10, we show the sets obtained at the termination of Algorithm 4, and those computed using IPOPT.

We then use the state-constraint set $\mathbb{X}_{[1]} = \mathcal{W}$ to synthesize the RMPC controller in (7.3) for system (5.80). To this end, we synthesize an LQR-feedback gain with $S_{[1]} = \operatorname{diag}(10, 1, 1)$, following which we select $P_{[1]}$ to be the solution of the DARE, and compute an RPI set $\Delta\mathbb{X}_{[1]}$ with 300 hyperplanes using the single-LP procedure in [144]: the computed set satisfies the inclusions in (5.90). Finally, we select the terminal set $\mathcal{X}_{\text{term}}$ as the maximal positive invariant set computed using the procedure in [44]. The results of synthesizing the RMPC controller in (7.3) for this parametrization, and applied to control the full-scale system (5.79) is shown in Figure 11. For comparison, we synthesize a full-order MPC controller that has a feedback of the full-state $[x^\top_{[1]}(t)\ x_{[2]}(t)^\top]$ for all $t \geq 0$, and uses the full-scale system (5.79) as the prediction model. In this controller, we directly constraint $y \in \mathcal{Y}$ along with $u \in \mathcal{U}$, and define the cost using matrices $(S_{[1]}, P_{[1]})$ as done in (7.3) along with a regularization term $10^{-6}||x_{[2]}||^2_2$ to guarantee positive definiteness of the cost. Finally, we use

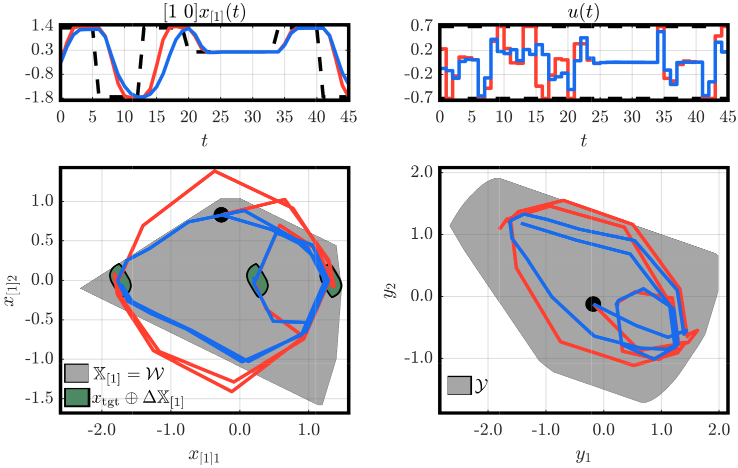


Figure 11: Comparison of closed-loop performance between reduced-order MPC controller (in blue) and full-order MPC controller (in red). While the full-order MPC controller satisfies $y \in \mathcal{Y}$ despite $x_{[1]} \notin \mathbb{X}_{[1]}$, the reduced-order MPC controller maintains $x_{[1]} \in \mathbb{X}_{[1]}$ to ensure that $y \in \mathcal{Y}$. The black dots indicate $x_{[1]}(0)$ and $y(0)$ in bottom-left and bottom-right plots respectively.

the same terminal constraint set (i.e., only over $x_{[1]}$) as done in (7.3). We observe in the figure that, as expected, the tracking performance with the reduced-order MPC controller is worse when compared to the full-order MPC controller, since the reduced-order controller requires $x_{[1]} \in \mathbb{X}_{[1]}$ to ensure that $y \in \mathcal{Y}$. However, we achieve the goals of the controller synthesis procedure, i.e., we perform reference tracking while ensuring that $y \in \mathcal{Y}$ without requiring measurements of neither $x_{[2]}$ nor y. Moreover, because of the reduced-order nature, the average iteration time of the proposed MPC controller is 0.015s, while the full-order MPC controller requires 0.033s when solved with the Gurobi QP solver.

5.4 Comparison with parametric sensitivity analysis-based approach

In this section, we compare our approach to computing a safe disturbance set using the 0-reachable set, with the parametric sensitivity approach presented in [160]. We specialize the approach of [160] to the case of an autonomous LTI system for the purpose of comparison. Moreover, we assume for simplicity that the system matrices are $C = \mathbf{I}$ and $D = \mathbf{0}$, such that the output-constraint set $\mathcal{Y}$ is the state-constraint set.

Given a *template* disturbance set $\tilde{\mathcal{W}}$ satisfying the usual assumptions of compactness and including the origin, the approach of [160] can be used to compute the largest scaling factor $\sigma \in [0, \bar{\sigma}]$ such that some given state $x \in \mathcal{Y}$ is included in the Maximal RPI (MRPI) set corresponding to the scaled disturbance set $\sigma\tilde{\mathcal{W}}$, where $\bar{\sigma} \geq 0$ is some user-specified scalar. In other words, for each $x \in \mathcal{Y}$, a function $V(x)$ is computed that satisfies

$$w \in \sigma\tilde{W},\ \forall\, \sigma \in [0, V(x) \leq \bar{\sigma}] \Rightarrow x \in \boldsymbol{\mathcal{O}}_\infty(\sigma\tilde{\mathcal{W}}). \tag{5.91}$$

We refer the reader to Section 2.2.2 for details regarding the MRPI set. In order to compute the function $V(x)$, a value iteration technique was proposed, the fixed-point of which was shown in [160, Theorem 3] to be recoverable from the MRPI set of the augmented uncertain system

$$\begin{aligned} &x(t+1) = Ax(t) + Bw(t), \qquad w(t) \in \sigma\tilde{\mathcal{W}} \\ &\sigma(t+1) = \sigma(t) \end{aligned}$$

subject to the constraints $(x, \sigma) \in \mathcal{Y} \times [0, \bar{\sigma}]$. This set can be computed as the limit of the set iterations

$$\tilde{\boldsymbol{\mathcal{O}}}_0 := \left\{ \begin{pmatrix} x \\ \sigma \end{pmatrix} \ :\ Gx \leq g,\ \sigma \in [0, \bar{\sigma}] \right\} \tag{5.92a}$$

$$\tilde{\boldsymbol{\mathcal{O}}}_{t+1} := \left\{ \begin{pmatrix} x \\ \sigma \end{pmatrix} \in \tilde{\boldsymbol{\mathcal{O}}}_t \ :\ \begin{pmatrix} Ax + Bw \\ \sigma \end{pmatrix} \in \tilde{\boldsymbol{\mathcal{O}}}_t,\ \forall\, w \in \sigma\tilde{\mathcal{W}} \right\} \tag{5.92b}$$

$$= \left\{ \begin{pmatrix} x \\ \sigma \end{pmatrix} \ :\ \begin{bmatrix} \boldsymbol{M}_t^x A & \boldsymbol{M}_t^\sigma + h_{\tilde{\mathcal{W}}}(\boldsymbol{M}_t^x B) \\ \mathbf{0} & -1 \\ \mathbf{0} & 1 \\ \boldsymbol{M}_t^x & \boldsymbol{M}_t^\sigma \end{bmatrix} \begin{bmatrix} x \\ \sigma \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{n}_t \\ 0 \\ \bar{\sigma} \\ \boldsymbol{n}_t \end{bmatrix} \right\}, \tag{5.92c}$$

$$\text{where } \tilde{\boldsymbol{\mathcal{O}}}_t = \left\{ \begin{pmatrix} x \\ \sigma \end{pmatrix} \ :\ \boldsymbol{M}_t^x x + \boldsymbol{M}_t^\sigma \sigma \leq \boldsymbol{n}_t \right\}. \tag{5.92d}$$

Note that in the computation of the set $\tilde{\boldsymbol{\mathcal{O}}}_{t+1}$, positive homogeneity of support functions is exploited. Then, for a given $x \in \mathcal{Y}$, the value function $V(x)$ satisfying (5.91) is given by

$$V(x) = \max\{\, \sigma \;:\; (x, \sigma) \in \tilde{\boldsymbol{\mathcal{O}}}_\infty \},$$

and the MRPI set for the system $x(t+1) = Ax(t) + Bw(t)$ with disturbances $w \in \sigma\tilde{\mathcal{W}}$ is given by

$$\boldsymbol{\mathcal{O}}_\infty(\sigma\tilde{\mathcal{W}}) = \{x : \boldsymbol{M}^x_\infty x \leq \boldsymbol{n}_\infty - \boldsymbol{M}^\sigma_\infty \sigma\}. \tag{5.93}$$

It follows immediately that the maximal scaled disturbance set that the system can tolerate is $\hat{\sigma}\tilde{\mathcal{W}}$, where

$$\hat{\sigma} = \max\left\{\sigma \;: \exists\, x \;:\; \begin{pmatrix} x \\ \sigma \end{pmatrix} \in \tilde{\boldsymbol{\mathcal{O}}}_\infty \right\}.$$

Such a maximal scaling factor $\hat{\sigma}$ has also been studied previously, e.g., [133], where-in it was referred to as the *critical scaling factor*. It implies that for any scaled disturbance set $\sigma\tilde{\mathcal{W}}$ with $\sigma > \hat{\sigma}$, there does not exist an RPI set for the system inside the constraint set. Moreover, the *largest* mRPI set corresponding to the maximal scaled disturbance set in included in the *smallest* MRPI set $\boldsymbol{\mathcal{O}}_\infty(\hat{\sigma}\tilde{\mathcal{W}})$, i.e.,

$$\mathcal{X}_{\mathrm{m}}(\hat{\sigma}\tilde{\mathcal{W}}) = \bigoplus_{t=0}^{\infty} A^t B(\hat{\sigma}\tilde{\mathcal{W}}) \subseteq \boldsymbol{\mathcal{O}}_\infty(\hat{\sigma}\tilde{\mathcal{W}}) \subseteq \mathcal{Y}. \tag{5.94}$$

Now considering the reachable set-based methodology proposed in this chapter, we observe that an arbitrarily tight lower bound to $\hat{\sigma}$ can be computed by solving a single LP. In particular, given a template disturbance set $\tilde{\mathcal{W}}$, safety of a scaled disturbance set $\sigma\tilde{\mathcal{W}}$ can be enforced as

$$\sigma \sum_{t=0}^{s-1} h_{B\tilde{\mathcal{W}}}(\bar{G}_{[t]}) \leq g - \lambda \sum_{t=0}^{s-1} h_{\mathcal{B}^{n_x}_\infty}(\bar{G}_{[t]}), \tag{5.95}$$

where the parameters (s, α, λ) are chosen appropriately for given (μ, γ) following Algorithm 3. Then, the largest scaling factor for given RPI set approximation error $\mu > 0$ can be computed by solving the LP

$$\sigma_\mu := \max_{\sigma}\; \sigma \text{ s.t. (5.95)},\;\; \sigma h_{B\mathcal{W}}(\tilde{\mathbf{I}}_{n_x}) \leq \gamma \mathbf{1}. \tag{5.96}$$

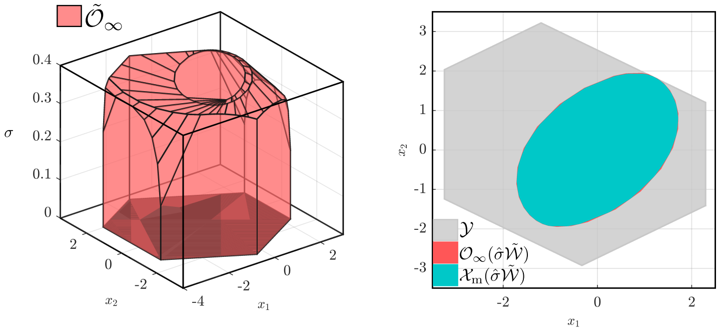


Figure 12: (Left) Set $\tilde{\boldsymbol{\mathcal{O}}}_\infty$ obtained at termination of iterations in (5.92); (Right) Corresponding smallest MRPI set and largest mRPI set defined in (5.93) and (5.94) respectively.

Clearly, σ_μ is a nondecreasing sequence with decreasing μ, and is upper-bounded by $\hat{\sigma}$. Moreover, for every $\sigma \in [0, \sigma_\mu]$ for every $\mu > 0$, the disturbance set $\sigma\tilde{\mathcal{W}}$ is a safe disturbance set, i.e., the inclusions

$$\mathcal{X}_{\mathrm{m}}(\sigma\tilde{\mathcal{W}}) \subseteq \mathcal{Y},\ \forall\, \sigma \in [0, \sigma_\mu],\ \forall\, \mu > 0$$

hold. On the other hand, defining

$$\sigma_t = \max\left\{\sigma\ :\exists\, x\ :\ \begin{pmatrix} x \\ \sigma \end{pmatrix} \in \tilde{\boldsymbol{\mathcal{O}}}_t\right\},$$

and observing that σ_t is a nonincreasing sequence in t with $\hat{\sigma} = \lim_{t\to\infty} \sigma_t$, we see that a disturbance set $\sigma_t\tilde{\mathcal{W}}$ is safe if and only if $\hat{\sigma} = \sigma_t$, thus requiring finite termination of the set iterations in (5.92).

Example

As an illustrative example, we consider the randomly generated system

$$x(t+1) = \begin{bmatrix} 0.8044 & -0.1734 \\ 0.1734 & 0.8044 \end{bmatrix} x(t) + \begin{bmatrix} 0.8264 & 0.5298 \\ 0.5460 & 0 \end{bmatrix} w(t),$$

and choose the template disturbance set $\tilde{\mathcal{W}}$ as a regular hexagon in $\mathbb{R}^2$. In Figure (12), we plot the sets obtained using the approach presented

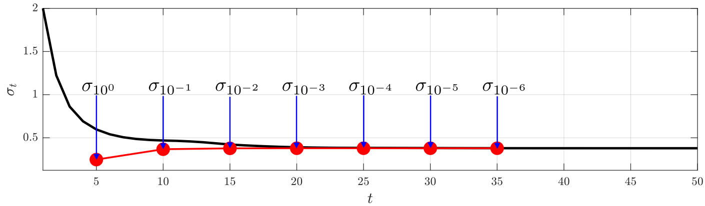


Figure 13: Comparison between the approach of [160] and the 0-reachable set based approach to compute the largest safe disturbance set.

in [160]. As expected, the smallest MRPI set $\mathcal{O}_\infty(\hat{\sigma}\tilde{\mathcal{W}})$ and the largest mRPI set $\mathcal{X}_{\mathrm{m}}(\hat{\sigma}\tilde{\mathcal{W}})$ are very close to each other while satisfying the inclusion in (5.94). In Figure 13, we plot the values of σ_μ for reducing values of μ obtained by solving LP (5.96). We observe that we monotonically approach $\hat{\sigma}$ from below as μ reduces.

Chapter 6

Input Constraint Sets for Robust Regulation of Linear Systems

6.1 Introduction

Constrained systems with unknown but bounded disturbances can be robustly stabilized using several control strategies, e.g., Robust Model Predictive Control (RMPC) schemes [87, 26, 42]. The main components that are required to synthesize controllers using these schemes are: a) a model of the system to control, including the descriptions of the state constraints and model uncertainty set; b) tuning parameters defining the cost function; c) a set of feasible inputs (the input constraint set). Then, the RMPC controller solves an online optimization problem to compute inputs that belong to the input constraint set, such that the system is stabilized from a given set of initial conditions. Component (a) can be obtained by using a system identification procedure, e.g., [79, 105, 95]; component (b) can be obtained by some tuning procedure, e.g., by preference-based calibration [164], or, if a desired linear feedback is available, through a controller matching procedure, e.g., [34, 162]. In this chapter, we tackle the computation of component (c), i.e., the input constraint set.

Typically, the input constraint set is directly characterized by the parameters that describe the technical specifications of the actuators. For example, pump parameters such as impeller size and motor capacity dictate the set of flow-rate inputs [56]. These parameters are usually selected during the system design phase by optimizing a criterion that captures various specifications such as costs, reliability, performance, etc. Hence, the procedures employed in the system design phase dictate the input constraints enforced in the control design phase. Given a set of input constraints, the set of initial-conditions from which the system can be robustly regulated is called the Maximal Robust Control Invariant (MRCI) set [58, 126, 134]. Then, given a desired set of initial-conditions of the system, the input constraint set could be *undersized*, i.e., the initial-condition set is not included in the MRCI set, or *oversized*, i.e., a potentially smaller input constraint set could be used to stabilize the system from those initial-conditions. In this chapter, we present a methodology to bridge the system design and control design phases by computing an *optimally-sized* input constraint set that explicitly accounts for the stabilizability requirements, i.e., it computes the actuator parameters that optimize the selection criterion used in the system design phase, while ensuring that the MRCI set corresponding to the resulting input constraint set contains a desired set of initial states. We also present a simple extension to the proposed methods to account for the modification in the system dynamics that can accompany actuator selection, thus enhancing its practicality as an engineering tool. In the rest of this chapter, we refer to the selection criterion as the input constraint set *size*, which is meant in an extended sense as a user-defined optimality metric. We consider linear time-invariant systems of the form

$$x(t+1) = Ax(t) + Bu(t) + B_w w(t), \tag{6.1}$$

with state $x \in \mathbb{R}^{n_x}$, control input $u \in \mathbb{R}^{n_u}$, additive bounded disturbance $w \in \mathcal{W} \subset \mathbb{R}^{n_w}$, and subject to state constraints $x \in \mathcal{X}$. In order to design the actuators for a given set $\Omega \subset \mathbb{R}^n$ of initial states, one can formulate the following problem:

Problem 6.1. *Find the **smallest set** U of input constraints required to **robustly** regulate $x(t)$, i.e., to guarantee constraint satisfaction $x \in \mathcal{X}$ with inputs $u \in U$ for all possible disturbances $w \in \mathcal{W}$, **from all** initial states $x(0) \in \Omega$.*

Note that the existence of a solution to Problem 6.2 entails the existence of a control law κ with a corresponding nonempty Robust Positive Invariant (RPI) set $\mathcal{X}_\kappa \supseteq \Omega$, as we will clarify in Section 6.2. Similar problems have been tackled previously in the context of actuator selection: in [147], the smallest number of actuators required to drive all $x(0) \in \Omega$ to some subset of the state-space is computed for a diagonal matrix B; in [146], the minimal actuator set problem is solved with an additional upper bound on the control effort required to reach the desired subset; in [136], an algorithm is presented to perform the actuator selection online, in a model predictive control fashion. However, none of these works consider systems with uncertainties and state constraints. The closest approach to the one we discuss was presented in [122], in which set-invariance properties were used to formulate an actuator-saturation design problem. Similar to the contribution in this chapter, it is assumed that a set of desired initial conditions is given a priori, and safe actuator saturation limits are computed. However, differently from our approach, it is assumed that the system is equipped with a static feedback law (requiring to work with positive invariant sets, rather than control invariant sets), and both uncertainty and state constraints can not be included in the formulation of the problem.

Unfortunately, as we will discuss in Section 6.2, solving Problem 6.1 might be difficult, so we propose to rely on RMPC to define κ and reformulate the problem in the following more tractable, though slightly conservative, way:

Problem 6.2. *Find the **smallest** input constraint set U required to **guarantee recursive feasibility** of the RMPC scheme presented in [87] **for all** $x(0) \in \Omega$.*

This second formulation is justified by the observation that, in practice, the technique of choice for enforcing robust invariance is often RMPC. To address Problem 6.2, we formulate an optimization problem by using tools of set invariance and provide an algorithm to solve it. We prove that the algorithm always terminates, and analyze its properties that are of practical significance.

This chapter is organized as follows. In Section 6.2 we formulate the problem introduced above of determining the input constraint set. Then, in Section 6.3 we develop an algorithm to compute the set U, and present its relevant properties. In Section 6.4, we discuss the implementation of the developed algorithm. Finally in Section 6.5 we present three numerical examples, with the first to illustrate some basic properties of the methods, the second showing an application of the methods to perform actuator selection with practical considerations, and the third to show the scalability of the proposed methodology.

6.2 Problem formulation

In this section, we formulate Problems 6.1 and 6.2 by recalling the concepts of control invariance from [58], and the tube-based RMPC scheme from [87].

Formulation of Problem 6.1:

Problem 6.1 can be formulated using the *Maximal Robust Control Invariant (MRCI)* set $\mathcal{X}_\infty$, which is such that [58, Definition 2.5] $\mathcal{X}_{\mathrm{RCI}} \subseteq \mathcal{X}_\infty \in \mathbb{X}$, for all $\mathcal{X}_{\mathrm{RCI}}$ satisfying

$$x \in \mathcal{X}_{\mathrm{RCI}} \Rightarrow \begin{cases} x \in \mathcal{X}, \\ \exists\, u \in U \ :, Ax + Bu + B_w w \in \mathcal{X}_{\mathrm{RCI}},\ \forall\, w \in \mathcal{W} \end{cases} \tag{6.2}$$

$$\mathbb{X} := \{\mathcal{X}_{\mathrm{RCI}} \ : \ (6.2)\text{ holds}\}. \tag{6.3}$$

This implies that for every initial state $x(0) \in \mathcal{X}_\infty$ of system (7.1) and every time instant $N \geq 1$, there exists an admissible control sequence $u(k) \in U, \forall\, k \in \mathbb{I}_0^{N-1}$ resulting in an admissible state sequence $x(k) \in \mathcal{X}_\infty, \forall\, k \in \mathbb{I}_0^{N}$ for all possible disturbances $w(k) \in \mathcal{W}$. Then, Problem 6.1 can be formulated as

$$\min_{U} \ f(U) \quad \text{s.t.} \quad \Omega \subseteq \mathcal{X}_\infty(\mathcal{U}), \tag{6.4}$$

where $f(U)$ is, e.g., a measure of the size of the input constraint set U, and we made the dependence of $\mathcal{X}_\infty$ on $\mathcal{U}$ explicit. Note that, if $\mathcal{X}_\infty(\mathcal{U})$ is known, one can define a control law κ as a function which, for each $x \in \mathcal{X}_\infty$, selects any input u which satisfies (6.2). Then, the associated maximum RPI (MRPI) set satisfies $\mathcal{X}_\kappa = \mathcal{X}_\infty$.

Problem 6.1 guarantees that state constraints can be robustly satisfied at all times and the system is regulated to $\mathcal{X}_\infty$. However, solving (6.4) is difficult, since one needs to solve an optimization problem with variable U and the MRCI set as a function of U. Additionally, one is often interested in regulating the state of system (7.1) to a smaller target neighborhood of the origin. A popular technique that allows one to achieve this objective is RMPC. In RMPC, robust invariance is enforced by requiring that the RMPC control law is able to steer all initial states $x(0)$ to a target RPI set [58, Definition 2.1] in N-steps. This implicitly defines a second larger RPI set (the feasible domain of RMPC) which approximates the MRCI set $\mathcal{X}_\infty$, but is by definition no larger, and hence a certain degree of conservativeness is introduced. Note that, rather than constructing the MRPI set first and defining a control law κ next, this second approach amounts to the opposite, which defines the mechanism used to formulate Problem 6.2.

Formulation of Problem 6.2:

In this chapter, we present the formulation for the tube-based RMPC scheme from [87], which is constructed using the disturbance-free nominal system

$$\hat{x}(t+1) = A\hat{x}(t) + B\hat{u}(t), \tag{6.5}$$

and a parametrized system input

$$u(t) = \hat{u}(t) + K(x(t) - \hat{x}(t)), \tag{6.6}$$

where K is a static feedback gain. Defining $A_K := A + BK$, the following standing assumptions are made.

Assumption 6.1.

(a) the static gain K is such that $\rho\left(A_K\right) < 1$;

(b) the sets $\mathcal{X}$ and $\mathcal{W}$ are compact, convex, and contain the origin in their nonempty interiors.

□

We will now recall again the tube-based RMPC formulation presented in (6.8), and apply it to the single system case using the notation in this chapter. Defining $\Delta x := x - \hat{x}$ and $\Delta u := u - \hat{u}$, an RPI set $\Delta\mathcal{X}$ is computed for the uncertain system $\Delta x(t+1) = A_K \Delta x(t) + B_w w(t)$, which satisfies the property

$$\Delta x(t) \in \Delta\mathcal{X} \implies \Delta x(t+1) \in \Delta\mathcal{X},\ \forall\, w(t) \in \mathcal{W}.$$

This property implies that if the current system state $x(t) \in \hat{x}(t) \oplus \Delta\mathcal{X}$, and an input is computed as in (6.6), then the successive system state satisfies $x(t+1) \in \hat{x}(t+1) \oplus \Delta\mathcal{X}$, i.e., the system state always belongs to the *uncertainty tube* $\hat{x} \oplus \Delta\mathcal{X}$. Then, from (6.6) the system input always belongs to the set $\hat{u} \oplus K\Delta\mathcal{X}$.

Since the uncertainty tubes define all possible future evolutions of system (7.1), the RMPC scheme provides robust constraint satisfaction using the tightened constraint sets $\mathcal{X} \ominus \Delta\mathcal{X}$ and $U \ominus K\Delta\mathcal{X}$. These sets guarantee that, if $\hat{x}(t) \in \mathcal{X} \ominus \Delta\mathcal{X}$ and $\hat{u}(t) \in U \ominus K\Delta\mathcal{X}$, then the state and input satisfy $x(t) \in \mathcal{X}$ and $u(t) \in U$, and $x(t+1) \in \mathcal{X}$.

Assumption 6.2. *The RPI set $\Delta\mathcal{X}$ is small enough such that the origin belongs to the nonempty interior of the tightened constraint sets, i.e.,*

$$\mathbf{0} \in \operatorname{int}(\mathcal{X} \ominus \Delta\mathcal{X}), \qquad \mathbf{0} \in \operatorname{int}(U \ominus K\Delta\mathcal{X}). \tag{6.7}$$

The nominal input $\hat{u}(t)$ is computed at each time instant by measuring $x(t)$ and solving the following optimization problem [87]:

$$\min_{\mathbf{z}} \sum_{s=t}^{t+N-1} \|\hat{x}(s)\|_Q^2 + \|\hat{u}(s)\|_R^2 + \|\hat{x}(t+N)\|_P^2 \tag{6.8a}$$

$$\text{s.t. } x(t) \in \hat{x}(t) \oplus \Delta\mathcal{X}, \tag{6.8b}$$

$$\hat{x}(s+1) = A\hat{x}(s) + B\hat{u}(s), \qquad s \in \mathbb{I}_t^{t+N-1}, \tag{6.8c}$$

$$\hat{x}(s) \in \mathcal{X} \ominus \Delta\mathcal{X}, \qquad s \in \mathbb{I}_{t+1}^{t+N-1}, \tag{6.8d}$$

$$\hat{u}(s) \in U \ominus K\Delta\mathcal{X}, \qquad s \in \mathbb{I}_t^{t+N-1}, \tag{6.8e}$$

$$\hat{x}(t+N) \in \mathcal{X}^{\mathrm{t}}, \tag{6.8f}$$

where $\mathbf{z} := \{\hat{x}(t), \ldots, \hat{x}(t+N), \hat{u}(t), \ldots, \hat{u}(t+N-1)\}$. The parameters K and P are chosen to satisfy the Discrete Algebraic Riccati Equation that solves the LQR problem for the nominal system (6.5) with positive definite matrices Q and R. The terminal set $\mathcal{X}^{\mathrm{t}}$ in (6.8f) is chosen to be a positive invariant (PI) set satisfying

$$A_K\mathcal{X}^{\mathrm{t}} \subseteq \mathcal{X}^{\mathrm{t}} \subseteq \mathcal{X} \ominus \Delta\mathcal{X}, \qquad K\mathcal{X}^{\mathrm{t}} \subseteq U \ominus K\Delta\mathcal{X}, \tag{6.9}$$

which ensures that with $\hat{u}(s) = K\hat{x}(s)$ for all $s \geq t+N$, we have $\hat{x}(s) \in \mathcal{X}^{\mathrm{t}}$. Denoting the optimal solution of problem (6.8) by

$$\mathbf{z}_* := \{\hat{x}_*(t), \ldots, \hat{x}_*(t+N), \hat{u}_*(t), \ldots, \hat{u}_*(t+N-1)\},$$

the input $u(t) := \hat{u}_*(t) + K(x(t) - \hat{x}_*(t))$ is applied to the plant.

The set of all initial states $x(0)$ of system (7.1) from which the RMPC controller is recursively feasible and stabilizing [87, Proposition 2] is the *N-step controllable set* [58, Definition 2.3] defined as

$$\mathcal{K}_N(U, \mathcal{X}^{\mathrm{t}}) := \hat{\mathcal{K}}_N(U, \mathcal{X}^{\mathrm{t}}) \oplus \Delta\mathcal{X}, \tag{6.10}$$

where $\hat{\mathcal{K}}_N(U, \mathcal{X}^{\mathrm{t}})$ is the *N-step nominal controllable set*, i.e., the set of all initial states $\hat{x}(0)$ of the nominal system (6.5) for which there exists an admissible nominal control sequence that drives it to the PI terminal set $\mathcal{X}^{\mathrm{t}}$ in N-steps with an admissible nominal state evolution. Mathematically, it is defined as

$$\hat{\mathcal{K}}_N(U, \mathcal{X}^{\mathrm{t}}) := \left\{ \hat{x}(0) : \exists \left\{ \begin{array}{c} \hat{u}(t) \in U \ominus K\Delta\mathcal{X}, \\ \hat{x}(t) \in \mathcal{X} \ominus \Delta\mathcal{X}, \\ \hat{x}(t+1) = A\hat{x}(t) + B\hat{u}(t), \\ \forall\, t \in \mathbb{I}_0^{N-1} \end{array} \right\} : \hat{x}(N) \in \mathcal{X}^{\mathrm{t}} \right\}. \tag{6.11}$$

By (6.10) and (6.11), the N-step controllable set $\mathcal{K}_N(U, \mathcal{X}^{\mathrm{t}})$ is an RPI set for the RMPC scheme. Hence, by fixing the control law κ to be the RMPC scheme, we approximate the MRCI set $\mathcal{X}_\infty$ by the RPI set $\mathcal{K}_N(U, \mathcal{X}^{\mathrm{t}})$ (feasible domain of RMPC). Since all initial states $x(0)$ of system (7.1) belonging to this set can be driven to the smaller target RPI set $x(N) \in \mathcal{X}^{\mathrm{t}} \oplus \Delta\mathcal{X}$ with an admissible state and input evolution, a desired set of initial conditions Ω is stabilizable if the inclusion

$$\Omega \subseteq \mathcal{K}_N(U, \mathcal{X}^{\mathrm{t}}) \tag{6.12}$$

holds. Based on this observation, Problem 6.2 that approximates Problem 6.1 can be formulated as

$$\begin{aligned} \min_{U, N, K, \mathcal{X}^{\mathrm{t}}, \Delta\mathcal{X}} \quad & f(U) && \text{(6.13a)} \\ \text{s.t.} \quad & \Omega \subseteq \mathcal{K}_N(U, \mathcal{X}^{\mathrm{t}}), && \text{(6.13b)} \\ & A_K\mathcal{X}^{\mathrm{t}} \subseteq \mathcal{X}^{\mathrm{t}} \subseteq \mathcal{X} \ominus \Delta\mathcal{X}, && \text{(6.13c)} \\ & K\mathcal{X}^{\mathrm{t}} \subseteq U \ominus K\Delta\mathcal{X}, && \text{(6.13d)} \\ & \mathbf{0} \in \mathrm{int}(U \ominus K\Delta\mathcal{X}). && \text{(6.13e)} \end{aligned}$$

In this chapter, we assume that the feedback gain K and RPI set $\Delta\mathcal{X}$ are given a priori and optimize over U, $\mathcal{X}^{\mathrm{t}}$ and N.

Conservativeness of the proposed approach: The requirement to drive $x(N)$ to a target RPI subset of the MRCI set, and the input parametrization in (6.6) with a static linear feedback law introduce conservativeness into (6.13) as compared to (6.4). Moreover, additional conservativeness is introduced by fixing K and $\Delta\mathcal{X}$, since they affect both the uncertainty tube and the PI terminal set $\mathcal{X}^{\mathrm{t}}$. Jointly solving (6.13) also over these variables is a subject of future research. We note that (6.13) can also be formulated for the RMPC scheme proposed in [26]. Since the scheme uses exact uncertainty tubes, $\Delta\mathcal{X}$ is not present in the resulting formulation. This reduces conservativeness in the proposed approach, as we demonstrate in Example 6.5.1.

Remark 6.1. *The formulation in* (6.8) *assumes full knowledge of the state $x(t)$. In case only an estimate is available, one can enlarge the RPI set $\Delta\mathcal{X}$ to account for the estimation error, provided it is bounded. Further details of this formulation can be found in [88].* □

6.3 Computation of Sets U and $\mathcal{X}^{\mathrm{t}}$

We now discuss the computation of the smallest set U and a corresponding terminal set $\mathcal{X}^{\mathrm{t}}$ that solves Problem 6.2. To this end, we parameterize U with a finite-dimensional vector $\epsilon \in \mathbb{R}^{n_\epsilon}$ such that $U = \mathbb{U}(\epsilon)$, and define the size (or any other measure to be minimized) of U as the scalar function $\boldsymbol{f}(\epsilon) : \mathbb{R}^{n_\epsilon} \to \mathbb{R}$. We note that $\boldsymbol{f}(\epsilon) = f(\mathbb{U}(\epsilon))$, where $f(\cdot)$ is used to formulate (6.13).

Assumption 6.3. *Set $\mathbb{U}(\epsilon)$ and function $\boldsymbol{f}(\epsilon)$ satisfy:*

(a) $\mathbb{U}(\epsilon)$ is compact and convex for all ϵ; moreover, for all $\delta \geq 0$, there exists an ϵ such that $\delta\mathcal{B}_\infty^{n_u} \subseteq \mathbb{U}(\epsilon)$;

(b) The value of $\boldsymbol{f}(\epsilon)$ is a measure of the set $\mathbb{U}(\epsilon)$, i.e.,

$$\mathbb{U}(\epsilon^1) \subset \mathbb{U}(\epsilon^2) \implies 0 \leq \boldsymbol{f}(\epsilon^1) < \boldsymbol{f}(\epsilon^2) < \infty.$$

□

Assumption 6.3(a) ensures that there always exists a parameter ϵ such that $\mathbb{U}(\epsilon)$ includes any compact subset of $\mathbb{R}^{n_u}$ containing the origin. Then, in Assumption 6.3(b), we ensure that $\boldsymbol{f}(\epsilon)$ is well defined for every $\mathbb{U}(\epsilon)$, and the inequalities enforce strict monotonicity properties on $\boldsymbol{f}(\epsilon)$ with respect to $\mathbb{U}(\epsilon)$. We provide a clarifying example next.

One possible parametrization of the input constraint set is the polytope

$$\mathbb{U}(\epsilon) = \{u : F^u u \leq \epsilon\}.$$

Then, examples of the size function that satisfy Assumption 6.3(b) are:

1. *If a vector* $\mathrm{c} > \mathbf{0}$ *is such that each* c_i *denotes the unit cost of actuation in direction* i*, then*
$$\boldsymbol{f}(\epsilon) = \mathrm{c}^\top \epsilon$$
denotes the total cost of selecting the input constraint set $\mathbb{U}(\epsilon)$*;*

2. *Defining the ellipsoid* $\mathcal{M} := \{u : u^\top R u \leq 1\}$*, the size function*
$$\boldsymbol{f}(\epsilon) = \min\{\alpha : \mathbb{U}(\epsilon) \subseteq \alpha\mathcal{M}\}$$
denotes the upper bound to the largest energy input $u^\top R u$ *into system* (7.1).

In the sequel, we propose an algorithm to compute the parameter ϵ such that $\mathbb{U}(\epsilon)$ satisfies the requirements (6.7),(6.9),(6.12), and minimizes $\boldsymbol{f}(\epsilon)$. To this end, we formulate an optimization problem in Subsection 6.3.1 that is equivalent to (6.13). In Subsection 6.3.2, we develop an algorithm to solve the optimization problem, and discuss its properties in Subsection 6.3.3. In Subsection 6.3.4, we analyze the variation of the size of the optimal input constraint set with the horizon length N.

6.3.1 Input Constraint Set Computation Problem

For the finite dimensional parametrization $U = \mathbb{U}(\epsilon)$ of the input constraint set, we write the N-step controllable set defined in (6.10) as
$$\mathbb{K}_N(\epsilon, \mathcal{X}^{\mathrm{t}}) := \mathcal{K}_N(\mathbb{U}(\epsilon), \mathcal{X}^{\mathrm{t}}).$$

Then, we define the tightened constraint admissible set
$$\mathbb{C}(\epsilon) := \{\hat{x} : \hat{x} \in \mathcal{X} \ominus \Delta\mathcal{X},\ K\hat{x} \in \mathbb{U}(\epsilon) \ominus K\Delta\mathcal{X}\},$$

such that system (6.5) with nominal input $\hat{u} = K\hat{x}$ satisfies
$$\hat{x}(t+1) = A_K\hat{x}(t), \qquad \hat{x}(t) \in \mathbb{C}(\epsilon) \implies \hat{u}(t) \in \mathbb{U}(\epsilon). \tag{6.14}$$

Based on these sets, consider the following optimization problem that is equivalent to (6.13) for a fixed K, $\Delta\mathcal{X}$ and N:

$$
\begin{aligned}
(\hat{\epsilon}^N, \hat{\mathcal{T}}_{\mathrm{f}}) := \arg\min_{\epsilon, \mathcal{T}} \;& \boldsymbol{f}(\epsilon) && (6.15a)\\
\text{s.t. } & \Omega \subseteq \mathbb{K}_N(\epsilon, \mathcal{T}), && (6.15b)\\
& A_K \mathcal{T} \subseteq \mathcal{T} \subseteq \mathbb{C}(\epsilon), && (6.15c)\\
& \delta\mathcal{B}_\infty^{n_u} \subseteq \mathbb{U}(\epsilon) \ominus K\Delta\mathcal{X}, && (6.15d)
\end{aligned}
$$

where constraint (6.15c) ensures that $\mathcal{T}$ is a PI subset of $\mathbb{C}(\epsilon)$, thus satisfying (6.9); constraint (6.15b) is equivalent to (6.12); constraint (6.15d) formulated with some scalar $\delta > 0$ ensures that (6.7) is satisfied.

Since the problem defined in (6.15) involves optimizing over PI sets $\mathcal{T}$, solving it directly can be computationally challenging. To tackle this issue, we introduce the i-step feedback admissible set

$$
\mathbb{O}_i(\epsilon) := \left\{ \hat{x} : A_K^t \hat{x} \in \mathbb{C}(\epsilon), \;\; \forall\, t \in \mathbb{I}_0^i \right\},
$$

which is the set of initial states of system (6.14) that remain inside $\mathbb{C}(\epsilon)$ for i steps. Then, $\mathbb{O}_\infty(\epsilon)$ is the *Maximal Positive Invariant* (MPI) subset of $\mathbb{C}(\epsilon)$ [58, Definition 2.3]. Using this set, we propose to relax problem (6.15) by enforcing $\mathcal{T} = \mathbb{O}_\infty(\epsilon)$, thus formulating the problem:

$$
\mathbf{P}^N : \begin{cases}
\epsilon^N := \arg\min_{\epsilon} \; \boldsymbol{f}(\epsilon) & (6.16a)\\
\qquad \text{s.t. } \Omega \subseteq \mathbb{K}_N(\epsilon, \mathbb{O}_\infty(\epsilon)), & (6.16b)\\
\qquad\qquad \delta\mathcal{B}_\infty^{n_u} \subseteq \mathbb{U}(\epsilon) \ominus K\Delta\mathcal{X}. & (6.16c)
\end{cases}
$$

The constraint equivalent to (6.15c) is eliminated from the formulation of $\mathbf{P}^N$ since the inclusions

$$
A_K \mathbb{O}_\infty(\epsilon) \subseteq \mathbb{O}_\infty(\epsilon) \subseteq \mathbb{C}(\epsilon)
$$

hold by construction [44] under Assumptions 6.1 and 6.2. In the following result, we show that $\mathbf{P}^N$ is not more conservative than problem (6.15), i.e., if $\hat{\epsilon}^N$ solves $\mathbf{P}^N$ then it must also solve (6.15).

Proposition 6.1. *Suppose Assumptions 6.1, 6.2 and 6.3 hold. If problem* (6.15) *is feasible, then $\mathbf{P}^N$ is feasible and $\boldsymbol{f}(\epsilon^N) = \boldsymbol{f}(\hat{\epsilon}^N)$.* □

Proof. Feasibility of problem (6.15) implies bounded solution under Assumption 6.3. This solution satisfies

$$\hat{\mathcal{T}}_{\mathrm{f}} \subseteq \mathbb{O}_\infty(\hat{\epsilon}^N),$$

since $\mathbb{O}_\infty(\hat{\epsilon}^N)$ is the MPI subset of $\mathbb{C}(\epsilon^N)$ [58, Definition 2.2] under Assumptions 6.1 and 6.2. Hence, $\hat{\epsilon}^N$ is feasible for $\mathbf{P}^N$, which implies that the optimal value $\boldsymbol{f}(\epsilon^N) \leq \boldsymbol{f}(\hat{\epsilon}^N)$. The proof is concluded by noting that $\boldsymbol{f}(\hat{\epsilon}^N) \leq \boldsymbol{f}(\epsilon^N)$ since $(\epsilon^N, \mathbb{O}_\infty(\epsilon^N))$ is feasible for problem (6.15). □

Remark 6.2. *If Assumptions 6.1, 6.2 and 6.3 hold, and U is parametrized as $\mathbb{U}(\epsilon) = \{u : F^u u \leq \epsilon\}$, then the constraint set of problem* (6.15) *is convex. Then, if $\boldsymbol{f}(\epsilon)$ is chosen to be a convex function,* (6.15) *is a convex optimization problem. Moreover, if $\boldsymbol{f}(\epsilon)$ is a strictly convex function (for example, $\boldsymbol{f}(\epsilon) = \|\epsilon\|_2^2$), then the optimizer $\hat{\epsilon}^N$ is guaranteed to be unique. Since problem $\mathbf{P}^N$ is also convex, then $\epsilon^N = \hat{\epsilon}^N$ along with $\boldsymbol{f}(\epsilon^N) = \boldsymbol{f}(\hat{\epsilon}^N)$ if $\boldsymbol{f}(\epsilon)$ is strictly convex.* □

Remark 6.3. *In the formulation of $\mathbf{P}^N$, we assume that the dynamics of system* (7.1) *are unaffected by a change in input constraint set parameter ϵ. This assumption, however, might not be valid in certain scenarios. For example, a modification in the engine mass and inertia affect the dynamic properties of a car. In such cases, one can formulate constraint* (6.16b) *with the modified dynamical system*

$$x(t+1) = Ax(t) + Bu(t) + B_w w(t) + g(x(t), u(t), w(t), \epsilon).$$

The development of structure exploiting approaches to tackle this problem is a subject of future research. A simple approach, that we present in Example 6.5.3, models this modification as an increase in uncertainty by parametrizing the disturbance set $\mathcal{W}$ as $\mathbb{W}(\epsilon)$. This follows from the observation that $g(x, u, w, \epsilon)$ lies in a compact set for all $x \in \mathcal{X}$, $u \in \mathbb{U}(\epsilon)$ and $w \in \mathcal{W}$ under Assumptions 6.1 and 6.3. □

6.3.2 Solution Algorithm

We now present an iterative algorithm to solve problem $\mathbf{P}^N$. We require a tailored algorithm since the set $\mathbb{O}_\infty(\epsilon)$ formulating constraint (6.16b) is not known apriori. To this end, consider the variant $\mathbf{P}^{i,N}$ of $\mathbf{P}^N$ obtained by replacing the MPI set $\mathbb{O}_\infty(\epsilon)$ with the i-step feedback admissible set

Algorithm 5: Algorithm to solve $\mathbf{P}^N$ given $A, B, K, \mathcal{X}, \Delta\mathcal{X}, \Omega, N$

Initialize: Initialize $i \geq 0$;
1) Solve $\mathbf{P}^{i,N}$ for $\epsilon^{i,N}$;
2) If $\mathbb{O}_i(\epsilon^{i,N})$ is PI, stop. Else, increment i, go to Step 1;
Return: $U = \mathbb{U}(\epsilon^{i,N})$, $\mathcal{X}^{\mathrm{t}} = \mathbb{O}_\infty(\epsilon^{i,N})$

$\mathbb{O}_i(\epsilon)$ in constraint (6.16b). The resulting optimization problem is written as

$$\mathbf{P}^{i,N} : \begin{cases} \epsilon^{i,N} := \arg\min_{\epsilon} \ \boldsymbol{f}(\epsilon) & (6.17a) \\ \quad \text{s.t. } \ \Omega \subseteq \mathbb{K}_N\left(\epsilon, \mathbb{O}_i(\epsilon)\right), & (6.17b) \\ \quad \delta\mathcal{B}_\infty^{n_u} \subseteq \mathbb{U}(\epsilon) \ominus K\Delta\mathcal{X}. & (6.17c) \end{cases}$$

Problem $\mathbf{P}^{i,N}$ is related to problem $\mathbf{P}^N$ as follows: for every parameter ϵ satisfying constraint (6.16c), there exists a finite MPI set termination index [44, Theorem 4.1] given by

$$i^*(\epsilon) := \min\left\{i : A_K\mathbb{O}_i(\epsilon) \subseteq \mathbb{O}_i(\epsilon)\right\} < \infty, \tag{6.18}$$

such that $\mathbb{O}_i(\epsilon) = \mathbb{O}_\infty(\epsilon)$ for all $i \geq i^*(\epsilon)$. Labeling $\epsilon^{i,N}$ as the solution of $\mathbf{P}^{i,N}$, this implies that if $i \geq i^*(\epsilon^{i,N})$, then $\mathbb{O}_i(\epsilon^{i,N})$ is a PI set, and $\epsilon^{i,N}$ is a feasible solution to $\mathbf{P}^N$. Hence, we propose to solve a sequence of problems $\mathbf{P}^{i,N}$ for increasing values of i, and terminating the sequence at index $i = i_N$ at which the PI condition is satisfied. We summarize this procedure in Algorithm 5.

Computational considerations: We will discuss how to formulate $\mathbf{P}^{i,N}$ in practice for polyhedral sets in Section 6.4. In this case, the problem has linear constraints and a monotonic (possibly convex) cost, such that efficient algorithms can be deployed. The case of ellipsoidal sets is both more involved to analyze and more conservative, and is not discussed further in this chapter.

Remark 6.4. *Algorithm 5 follows a reasoning similar to the recursive computation of the MPI set proposed in [63, 44]. Index i is incremented until the invariance condition is satisfied. The difference is that we also recursively compute the input constraint set along with the MPI set in order to solve $\mathbf{P}^N$.* □

6.3.3 Feasibility, Convergence and Optimality of Algorithm 5

In this section, we show that Algorithm 5 solves $\mathbf{P}^N$. To this end, we will first formulate requirements on the initial-condition set Ω and horizon length N for $\mathbf{P}^{i,N}$ to be feasible. Then, we will show that Algorithm 5 terminates at some finite index i_N. Finally, we will show that $\mathbf{P}^N$ is solved at termination, i.e., $\boldsymbol{f}(\epsilon^N) = \boldsymbol{f}(\epsilon^{i_N,N})$.

Feasibility of $\mathbf{P}^{i,N}$

Problem $\mathbf{P}^{i,N}$ is feasible only if all initial-states $x(0) \in \Omega$ are controllable in N steps. In order to formalize this statement, we introduce the sets $\mathbb{O}_\infty(\infty)$ and $\mathbb{K}_N(\infty, \mathbb{O}_\infty(\infty))$, which we define using unconstrained inputs, i.e., $u \in \mathbb{R}^{n_u}$. The set $\mathbb{O}_\infty(\infty)$ is the MPI subset of $\mathbb{C}(\infty) = \mathcal{X} \ominus \Delta\mathcal{X}$ for system (6.14), and $\mathbb{K}_N(\infty, \mathbb{O}_\infty(\infty))$ is an N-step controllable set [58] with unconstrained inputs u. Using these sets, we formulate the following N-step controllability assumption:

Assumption 6.4. *All $x(0) \in \Omega$ are included in the N-step unconstrained controllable set, i.e., $\Omega \subseteq \mathbb{K}_N(\infty, \mathbb{O}_\infty(\infty))$.* □

Proposition 6.2. *Suppose Assumptions 6.1, 6.2, 6.3, and 6.4 hold. Then problem $\mathbf{P}^{i,N}$ is feasible and bounded.* □

Proof. Under Assumption 6.4, there exists a sequence of inputs

$$\{u_z(t),\ t \in \mathbb{I}_0^{N-1}\}$$

such that $x_z(N) \in \mathbb{O}_\infty(\infty) \oplus \Delta\mathcal{X}$ from each $x_z(0) = z \in \Omega$. Under Assumption 6.3, there exists an ϵ satisfying

$$\{u_z(t) \in \mathbb{U}(\epsilon),\ t \in \mathbb{I}_0^{N-1}\}, \quad \forall\, z \in \Omega$$

and $\mathbb{C}(\epsilon) = \mathcal{X} \ominus \Delta\mathcal{X}$. These conditions guarantee the existence of an $\epsilon < \infty$ such that, for all i, we have

$$\Omega \subseteq \mathbb{K}_N(\epsilon, \mathbb{O}_\infty(\infty)) \subseteq \mathbb{K}_N(\epsilon, \mathbb{O}_i(\epsilon)).$$

Hence, ϵ is feasible for $\mathbf{P}^{i,N}$, with boundedness imposed by constraint (6.16c). □

Termination of Algorithm 5

If we characterize an index ι_δ that is the maximum value of the MPI set termination index $i^*(\epsilon)$ for all parameters ϵ satisfying constraint (6.16c), then for all indices $i \geq \iota_\delta$, the solution $\epsilon^{i,N}$ of problem $\mathbf{P}^{i,N}$ satisfies the PI condition $A_K \mathbb{O}_i(\epsilon^{i,N}) \subseteq \mathbb{O}_i(\epsilon^{i,N})$. Then, there exists a termination index $i_N \leq \iota_\delta$ for Algorithm 5. However, characterizing ι_δ is not computationally possible, since the set of all ϵ satisfying constraint (6.16c) is open. In the following result, we establish an upper bound to i_N that can, in fact, be computed. To this end, consider the set

$$\mathcal{C}_\delta := \{\hat{x} : \hat{x} \in \mathcal{X} \ominus \Delta\mathcal{X}, K\hat{x} \in \delta\mathcal{B}_\infty^{n_u}\}, \tag{6.19}$$

which satisfies $\mathcal{C}_\delta \subseteq \mathbb{C}(\epsilon^{i,N})$ for all i, and the index

$$k_\delta := \min\{i : A_K^{i+1}(\mathcal{X} \ominus \Delta\mathcal{X}) \subseteq \mathcal{C}_\delta\}. \tag{6.20}$$

The existence of k_δ follows from Assumptions 6.1 and 6.2.

Proposition 6.3. *Suppose Assumptions 6.1, 6.2, 6.3, and 6.4 hold, then Algorithm 5 terminates at an index $i_N \leq k_\delta$.* □

Proof. The following inclusions hold for all i:

$$\mathbb{O}_i(\epsilon^{i,N}) \subseteq \mathbb{C}(\epsilon^{i,N}) \subseteq \mathcal{X} \ominus \Delta\mathcal{X}. \tag{6.21}$$

Then at the index k_δ, we have

$$\begin{aligned} A_K^{k_\delta+1}\mathbb{O}_{k_\delta}\left(\epsilon^{k_\delta,N}\right) &\subseteq A_K^{k_\delta+1}\mathbb{C}\left(\epsilon^{k_\delta,N}\right) \\ &\subseteq A_K^{k_\delta+1}\left(X \ominus \Delta\mathcal{X}\right) \subseteq \mathcal{C}_\delta \subseteq \mathbb{C}\left(\epsilon^{k_\delta,N}\right) \end{aligned}$$

from (6.19), (6.20), and (6.21). By definition of the feedback admissible set, the first and the last terms imply $\mathbb{O}_{k_\delta+1}(\epsilon^{k_\delta,N}) = \mathbb{O}_{k_\delta}(\epsilon^{k_\delta,N})$. Then,

$$A_K\mathbb{O}_{k_\delta}(\epsilon^{k_\delta,N}) \subseteq \mathbb{O}_{k_\delta}(\epsilon^{k_\delta,N})$$

from [44, Theorem 2.2]. □

Smaller values of the tuning factor δ result in a smaller set $\mathcal{C}_\delta$. This increases the upper bound k_δ to the termination index i_N of Algorithm 5, resulting in a larger number of iterations. However, from the formulation of $\mathbf{P}^{i,N}$, we see that a smaller value of δ results in a smaller lower bound on the optimal value of $\boldsymbol{f}(\epsilon^{i,N})$ (through constraint (6.16c)). Hence, δ dictates the trade-off between optimality and computational difficulty.

Solution to $\mathbf{P}^N$

We finally show that the termination of Algorithm 5 corresponds to the solution of $\mathbf{P}^N$, i.e., the optimal values coincide as $\boldsymbol{f}(\epsilon^{i_N,N}) = \boldsymbol{f}(\epsilon^N)$. We reason as follows : for all indices $i < i_N$, the PI condition is not satisfied, which implies $\epsilon^{i,N}$ is not feasible for $\mathbf{P}^N$. Hence, we must show that if $\mathbf{P}^{i,N}$ is solved for some $i > i_N$, then the optimal value $\boldsymbol{f}(\epsilon^{i,N})$ cannot be smaller than $\boldsymbol{f}(\epsilon^{i_N,N})$.

Proposition 6.4. *If Assumptions 6.1, 6.2, 6.3 and 6.4 hold, then* $\boldsymbol{f}(\epsilon^N) = \boldsymbol{f}(\epsilon^{i_N,N})$. □

Proof. Since the inclusion $\mathbb{O}_{i+1}(\epsilon^{i,N}) \subseteq \mathbb{O}_i(\epsilon^{i,N})$ holds for all i, the solution $\epsilon^{i+1,N}$ of $\mathbf{P}^{i+1,N}$ is feasible for $\mathbf{P}^{i,N}$. Then, the optimal values are non-decreasing as $\boldsymbol{f}(\epsilon^{i,N}) \leq \boldsymbol{f}(\epsilon^{i+1,N})$. Hence, $\boldsymbol{f}(\epsilon^{i_N,N}) \leq \boldsymbol{f}(\epsilon^{i,N})$ for all $i > i_N$, thus concluding the proof. □

Remark 6.5. *In some cases, Ω might violate the N-step controllability condition. Then, we propose to solve the optimization problem*

$$\tilde{\Omega} := \arg\min_{\tilde{\Omega}} \ \boldsymbol{d}(\Omega, \tilde{\Omega}) \ \ s.t. \ \ \tilde{\Omega} \subseteq \mathbb{K}_N(\infty, \mathbb{O}_\infty(\infty)), \tag{6.22}$$

where $\boldsymbol{d}(\Omega, \tilde{\Omega})$ is some distance metric. Since (6.22) *guarantees that $\tilde{\Omega}$ satisfies Assumption 6.4, the aforementioned properties of Algorithm 5 continue to hold for the projected initial-condition set $\tilde{\Omega}$.* □

Remark 6.6. *Since $\mathbb{O}_i(\epsilon^{i_N,N}) = \mathbb{O}_{i_N}(\epsilon^{i_N,N})$ for all $i \geq i_N$, the solution $\epsilon^{i_N,N}$ of $\mathbf{P}^{i_N,N}$ is feasible for all $\mathbf{P}^{i,N}$ with $i \geq i_N$. This implies $\boldsymbol{f}(\epsilon^{i,N}) = \boldsymbol{f}(\epsilon^{i_N,N})$ for all $i \geq i_N$. Hence, Algorithm 5 can be initialized at any index $i = i_{\text{init}} \geq 0$, and incremented in Step 3 with any $i_{\text{incr}} \geq 1$, i.e., $i \leftarrow i + i_{\text{incr}}$. Moreover, if $i_{\text{init}} = k_\delta$ from* (6.20)*, then Algorithm 5 terminates in one iteration.* □

6.3.4 Effect of the Horizon Length on the Input Constraint Set Size

In this section, we discuss the effect of the horizon length N on the optimal input constraint set size. In particular, we show that $\boldsymbol{f}(\epsilon^N)$ is monotonically non-increasing and convergent in N.

To this end, we use an auxiliary optimization problem $\tilde{\mathbf{P}}^N$ that computes the smallest input constraint set required to maintain the state of system (7.1) inside the constraint set $\mathcal{X}$ for N steps. It is formulated by replacing the target set $\mathbb{O}_\infty(\epsilon)$ in constraint (6.16b) by $\mathcal{X} \ominus \Delta\mathcal{X}$, such that $\mathbb{K}_N(\epsilon, \mathcal{X} \ominus \Delta\mathcal{X})$ is an *N-step admissible set* [58, Definition 2.11]. We label the solution of this problem as $\tilde{\epsilon}^N$, and write it as follows.

$$\tilde{\mathbf{P}}^N : \begin{cases} \tilde{\epsilon}^N := \arg\min_{\epsilon} \ \boldsymbol{f}(\epsilon) & (6.23a) \\ \qquad \text{s.t. } \ \Omega \subseteq \mathbb{K}_N(\epsilon, \mathcal{X} \ominus \Delta\mathcal{X}), & (6.23b) \\ \qquad \delta\mathcal{B}_\infty^{n_u} \subseteq \mathbb{U}(\epsilon) \ominus K\Delta\mathcal{X}. & (6.23c) \end{cases}$$

Proposition 6.5. *Suppose Assumptions 6.1, 6.2, 6.3, and 6.4 hold. Then,*

(i) $\boldsymbol{f}(\epsilon^N) \geq \boldsymbol{f}(\epsilon^{N+1})$*;*

(ii) $\lim_{N\to\infty} \boldsymbol{f}(\epsilon^N)$ *exists.*

□

Proof. For all ϵ satisfying constraint (6.16c), the N-step stabilizable and admissible sets satisfy the inclusions

$$\begin{aligned} \mathbb{K}_N(\epsilon, \mathbb{O}_\infty(\epsilon)) &\subseteq \mathbb{K}_{N+1}(\epsilon, \mathbb{O}_\infty(\epsilon)) \\ &\subseteq \mathbb{K}_{N+1}(\epsilon, \mathcal{X} \ominus \Delta\mathcal{X}) \subseteq \mathbb{K}_N(\epsilon, \mathcal{X} \ominus \Delta\mathcal{X}), \end{aligned}$$

from [58, Propositions 2.3, 2.4]. The first inclusion implies ϵ^N is feasible for $\mathbf{P}^{N+1}$, hence $\boldsymbol{f}(\epsilon^N) \geq \boldsymbol{f}(\epsilon^{N+1})$. The remaining two inclusions respectively imply $\boldsymbol{f}(\tilde{\epsilon}^N) \leq \boldsymbol{f}(\epsilon^N)$ and $\boldsymbol{f}(\tilde{\epsilon}^N) \leq \boldsymbol{f}(\tilde{\epsilon}^{N+1})$ by the same reasoning. Hence, $\{\boldsymbol{f}(\epsilon^N)\}_N$ is a non-increasing sequence, that is lower bounded by the non-decreasing sequence $\{\boldsymbol{f}(\tilde{\epsilon}^N)\}_N$. Thus, finite limits $\lim_{N\to\infty} \boldsymbol{f}(\epsilon^N)$ and $\lim_{N\to\infty} \boldsymbol{f}(\tilde{\epsilon}^N)$ exist. □

This result implies that the problem $\min_N \mathbf{P}^N$ which is equivalent to (6.13) can be solved by choosing a large enough value of N.

6.4 Polytopic Implementation of Algorithm 5

In this section we discuss the implementation of Algorithm 5 using polytopic sets

$$\mathcal{X} := \{x : H^x x \leq h^x\}, \qquad \mathcal{W} := \{w : F^w w \leq f^w\}$$

satisfying Assumption 6.1(b) with $h^x > \mathbf{0}$ and $f^w \geq \mathbf{0}$. A feedback gain K is assumed to be computed apriori. Then, we compute a polytopic RPI set

$$\Delta\mathcal{X} := \{\Delta x : H^{\Delta}\Delta x \leq h^{\Delta}\}$$

for the system $x(t+1) = A_K x(t) + B_w w(t)$ with established methods, e.g., those given in [112, 144]. Using $\Delta\mathcal{X}$, we tighten the state constraint set as $\mathcal{X} \ominus \Delta\mathcal{X} = \left\{\hat{x} : H^x\hat{x} \leq h^x - \bar{h}^x\right\}$, where each component

$$\bar{h}^x_j = \max_{\Delta x \in \Delta\mathcal{X}} H^x_j \Delta x.$$

We choose an input constraint set U parameterized as the polytope

$$\mathbb{U}(\epsilon) := \{u : F^u u \leq \epsilon\}$$

and satisfying Assumption 6.3(a). Then, the tightened input constraint set is $\mathbb{U}(\epsilon) \ominus K\Delta\mathcal{X} = \{\hat{u} : F^u\hat{u} \leq \epsilon^{\mathrm{m}}(\epsilon) - \bar{\epsilon}\}$, where each component

$$\epsilon^{\mathrm{m}}_j(\epsilon) := \max_{u \in \mathbb{U}(\epsilon)} F^u_j u, \qquad \bar{\epsilon}_j := \max_{\Delta x \in \Delta\mathcal{X}} F^u_j K \Delta x.$$

The function $\epsilon^{\mathrm{m}}(\epsilon)$ is such that if $\mathbb{U}(\epsilon^{\mathrm{m}}(\epsilon))$ contains any redundant hyperplanes, then they are tangent to the set. We now show that $\epsilon^{\mathrm{m}}(\epsilon^{i,N}) = \epsilon^{i,N}$.

Proposition 6.6. *Suppose Assumptions 6.1, 6.2, 6.3 and 6.4 hold, then we always have* $\epsilon^{\mathrm{m}}(\epsilon^{i,N}) = \epsilon^{i,N}$. □

Proof. We prove this result by contradiction. Suppose that for a given $\epsilon^{i,N}$, there exists some $\epsilon < \epsilon^{i,N}$ such that $\mathbb{U}(\epsilon^{i,N}) = \mathbb{U}(\epsilon)$. By Assumption 6.3(b), $\boldsymbol{f}(\epsilon) < \boldsymbol{f}(\epsilon^{i,N})$ follows, which is a contradiction since $\boldsymbol{f}(\epsilon^{i,N})$ is the optimal value of $\mathbf{P}^{i,N}$. □

This result permits us to directly tighten the input constraint set as

$$\mathbb{U}(\epsilon) \ominus K\Delta\mathcal{X} = \{\hat{u} : F^u\hat{u} \leq \epsilon - \bar{\epsilon}\}.$$

Using these definitions, we write the i-step feedback admissible set as $\mathbb{O}_i(\epsilon) := \{\hat{x} : S^{[i]}\hat{x} \leq q^{[i]}(\epsilon)\}$, where

$$S^{[i]} := \begin{bmatrix} \begin{bmatrix} H^x \\ F^u K \end{bmatrix} A_K^0 \\ \vdots \\ \begin{bmatrix} H^x \\ F^u K \end{bmatrix} A_K^i \end{bmatrix}, \qquad q^{[i]}(\epsilon) := \begin{bmatrix} \begin{bmatrix} h^x - \bar{h}^x \\ \epsilon - \bar{\epsilon} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} h^x - \bar{h}^x \\ \epsilon - \bar{\epsilon} \end{bmatrix} \end{bmatrix}.$$

We then use these sets to formulate problem $\mathbf{P}^{i,N}$ with a vertex notation of the initial-condition set $\Omega = \{x_0^{[k]}, k \in \mathbb{I}_1^{N_\Omega}\}$ as

$$\begin{aligned} \epsilon^{i,N} = \arg \min_{\epsilon, \mathbf{z}^{[k]} \forall k \in \mathbb{I}_1^{N_\Omega}} \; & \boldsymbol{f}(\epsilon) && (6.24) \\ \text{s.t.} \;\; & H^\Delta x_0^{[k]} \leq h^\Delta + H^\Delta \hat{x}^{[k]}(0) \\ & \hat{x}^{[k]}(s+1) = A\hat{x}^{[k]}(s) + B\hat{u}^{[k]}(s), && s \in \mathbb{I}_0^{N-1}, \\ & H^x \hat{x}^{[k]}(s) \leq h^x - \bar{h}^x, && s \in \mathbb{I}_1^{N-1}, \\ & F^u \hat{u}^{[k]}(s) \leq \epsilon - \bar{\epsilon}, && s \in \mathbb{I}_0^{N-1}, \\ & S^{[i]}\hat{x}^{[k]}(N) \leq q^{[i]}(\epsilon), \\ & \epsilon - \bar{\epsilon} \geq \delta\lambda, \end{aligned}$$

where $\mathbf{z}^{[k]} := \{\hat{x}^{[k]}(0), \ldots, \hat{x}^{[k]}(N), \hat{u}^{[k]}(0), \ldots, \hat{u}^{[k]}(N-1)\}$. The first constraint implies $x_0^{[k]} \in \hat{x}^{[k]}(0) \oplus \Delta\mathcal{X}$, and the last constraint encodes the inclusion $\delta\mathcal{B}_\infty^{n_u} \subseteq \mathbb{U}(\epsilon) \ominus K\Delta\mathcal{X}$, where

$$\lambda_j := \max_{u \in \mathcal{B}_\infty^{n_u}} F_j u.$$

The details of the formulation of $\mathbf{P}^{i,N}$ with a hyperplane notation $\Omega = \{x : R^\Omega x \leq r^\Omega\}$ is given in [100].

For the computed $\epsilon^{i,N}$, we verify if the set $\mathbb{O}_i(\epsilon^{i,N})$ is PI by checking for the existence of a matrix Λ^{p} satisfying

$$\Lambda^{\mathrm{p}} \geq \mathbf{0}, \quad \Lambda^{\mathrm{p}} q^{[i]}(\epsilon^{i,N}) \leq q^{[i]}(\epsilon^{i,N}), \quad \Lambda^{\mathrm{p}} S^{[i]} = S^{[i]} A_K,$$

which are necessary and sufficient conditions for the positive invariance inclusion $A_K \mathbb{O}_i(\epsilon^{i,N}) \subseteq \mathbb{O}_i(\epsilon^{i,N})$ to hold [127].

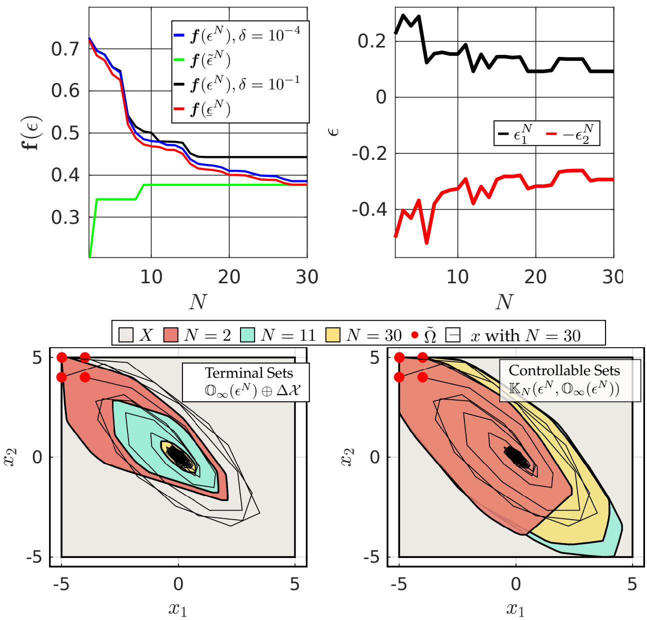


Figure 14: Numerical results for Example 1. (Upper-left plot) Optimal input constraint set size $\boldsymbol{f}(\epsilon^N)$ computed by Algorithm 5, along with the lower bound $\boldsymbol{f}(\tilde{\epsilon}^N)$. These values satisfy the monotonically convergent properties discussed in Proposition 6.5. (Upper-right plot) Optimal input constraint set parameters ϵ^N computed by Algorithm 5. The sets are not necessarily nested. (Lower-left plot) Terminal sets computed by Algorithm 5 for $N = 2, 11, 30$ (Lower-right plot) Corresponding controllable sets for $N = 2, 11, 30$. The parameters ϵ^N are computed such that the terminal set and controllable set align towards $\tilde{\Omega}$.

Remark 6.7. *Since $\mathbf{P}^N$ explicitly minimizes the input constraint set size, the resulting closed-loop performance of the RMPC controller with sets $U = \mathbb{U}(\epsilon^N)$ and $\mathcal{X}^{\mathrm{t}} = \mathbb{O}_\infty(\epsilon^N)$ might be unsatisfactory. This can be ameliorated by formulating the objective function of $\mathbf{P}^N$ with a trade-off parameter $\kappa_{\mathrm{p}} \geq 0$, e.g., as*

$$\boldsymbol{d}(\epsilon, \mathbf{z}) := \boldsymbol{f}(\epsilon) + \kappa_{\mathrm{p}} \sum_{k=1}^{N_\Omega} \left(\sum_{s=0}^{N-1} \left(\left\| \hat{x}^{[k]}(s) \right\|_Q^2 \right) + \left\| \hat{x}^{[k]}(N) \right\|_P^2 \right). \tag{6.25}$$

Then, the optimal value of $\boldsymbol{d}$ is non-increasing in N. The formulation of an objective function similar to (6.25) *if Ω is given in a hyperplane notation is a* *subject of future research.* □

6.5 Numerical Examples

In this section we present three numerical examples, with the first aimed at illustrating the properties of the approach, the second being a practical example in which we use the proposed methods to size pneumatic actuators for a force control application, and the third demonstrating the scalability of problem $\mathbf{P}^{i,N}$. We choose affine and piecewise affine size functions for these examples. This choice results in a Linear Program (LP) or Mixed-Integer LP (MILP), respectively, which we solve using the CPLEX solver [31]. We remark that other choices of size functions can be used in their formulation.

6.5.1 Illustrative system

The goal of the following example is to illustrate some fundamental properties of the approach. Consider the unstable system with dynamics

$$x(t+1) = \begin{bmatrix} 1.4 & 1 \\ -1 & 0.1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 3 \end{bmatrix} u(t) + w(t)$$

and state constraints $x \in 5\mathcal{B}_\infty^2$. We equip the system with a feedback controller K which is the solution of the Riccati equation for matrices $Q = 0.01\mathbf{I}$ and $R = 1$. The disturbance set is $\mathcal{W} = 0.1\mathcal{B}_\infty^2$, for which we compute the RPI set $\Delta\mathcal{X}$ as a polytope with 200 facets using the method presented in [144]. The initial-condition set is $\Omega = \{x_{[k]}^0, k \in \mathbb{I}_1^4\}$, where $x_{[1]}^0 = [-4, 4]^\top$, $x_{[2]}^0 = [-4, 6]^\top$, $x_{[3]}^0 = [-6, 6]^\top$ and $x_{[4]}^0 = [-6, 4]^\top$. Since this set does not satisfy Assumption 6.4, we follow the discussion in Remark 3 and compute the set $\tilde{\Omega} = \left\{\tilde{x}_{[k]}^0, k \in \mathbb{I}_1^4\right\}$ by solving the problem in (6.22) with the distance

$$\boldsymbol{d}\left(\Omega, \tilde{\Omega}\right) = \sum_{k=1}^{4} \left\| x_{[k]}^0 - \tilde{x}_{[k]}^0 \right\|_1 .$$

This choice implies problem (6.22) is an LP, and results in $\tilde{x}^0_{[1]} = [-4,4]^\top$, $\tilde{x}^0_{[2]} = [-4,5]^\top$, $\tilde{x}^0_{[3]} = [-4.9876, 4.9983]^\top$ and $\tilde{x}^0_{[4]} = [-5,4]^\top$, satisfying Assumption 6.4.

We parametrize the input constraint set U using $\epsilon \in \mathbb{R}^2$ as the saturation $\mathbb{U}(\epsilon) = \{u : -\epsilon_2 \leq u \leq \epsilon_1\}$ with $\epsilon_1, \epsilon_2 \geq 0$, and use the size function $\boldsymbol{f}(\epsilon) = \epsilon_1 + \epsilon_2$. Then, $\mathbf{P}^{i,N}$ is also an LP. We select a tuning factor of $\delta = 10^{-4}$, for which the upper bound to the termination index (6.20) of Algorithm 5 is $k_\delta = 28$. The corresponding results are shown in Figure 14. We report that Algorithm 5 converges at index $i = 2$ for $N = 3, 5$, $i = 3$ for $N = 2, 4, 11$, $i = 5$ for $N = 8, 10, 13, 16, 18$, and $i = 6$ for all other values of N. In Figure 14-upper-left plot, we see that $\boldsymbol{f}(\epsilon^N)$ is non-increasing and convergent, and is lower bounded by $\boldsymbol{f}(\tilde{\epsilon}^N)$. This follows from Proposition 6.5. We also plot $\boldsymbol{f}(\underline{\epsilon}^N)$, which is obtained by formulating Problem $\mathbf{P}^N$ with recursive feasibility conditions in [26]. Since this formulation uses exact uncertainty tubes, the resulting $\boldsymbol{f}(\underline{\epsilon}^N)$ is less conservative than $\boldsymbol{f}(\epsilon^N)$ [163]. In the upper-right plot, we see that, despite $\boldsymbol{f}(\epsilon^N)$ being non-increasing, the sets $\mathbb{U}(\epsilon^N)$ are not nested, i.e.,

$$\mathbb{U}(\epsilon^{N+1}) \not\subseteq \mathbb{U}(\epsilon^N).$$

This is because $\boldsymbol{f}(\epsilon^{N+1}) < \boldsymbol{f}(\epsilon^N)$ does not necessarily imply $\mathbb{U}(\epsilon^{N+1}) \subset \mathbb{U}(\epsilon^N)$ as per Assumption 6.3(b). However, this inclusion holds at $N = 3, 5, 11, 13, 16, 18, 26$, at which the termination index of Algorithm 5 increases ([44, Theorem 4.1]). In order to demonstrate the effect of δ on the performance of the algorithm, we plot the optimal values $\boldsymbol{f}(\epsilon^N)$ in the upper-left plot when $\delta = 10^{-1}$ is chosen in Algorithm 5. As discussed in Section 6.3.3, this choice results in conservative values of $\boldsymbol{f}(\epsilon^N)$. However, the termination index upper bound (6.20) is $k_\delta = 8$, thus resulting in reduced computational difficulty. We report that Algorithm 5 then converges at $i \leq 5$ for $N = 2, \ldots, 10$. The lower-left plot shows the terminal sets $\mathbb{O}_\infty(\epsilon^N) \oplus \Delta\mathcal{X}$, and the lower-right plot shows the feasible sets $\mathbb{K}_N(\epsilon^N, \mathbb{O}_\infty(\epsilon^N))$. We observe that these sets are not nested since they correspond to different input constraint sets. The terminal sets computed by Algorithm 5 are such that $\mathbb{K}_N(\epsilon^N, \mathbb{O}_\infty(\epsilon^N))$ are aligned in the direction of $\tilde{\Omega}$, thus minimizing the actuation effort required for stabilizing.

The closed-loop trajectories with the RMPC controller for $N = 30$ are shown. We also formulate $\mathbf{P}^N$ with the objective in (6.25) with $\kappa_{\mathrm{p}} = 0.01$ to improve closed-loop performance, which we solve using Algorithm 5 for $N = 30$. The resulting nominal state cost

$$\sum_{k=1}^{4} \sum_{t=0}^{N_{\mathrm{sim}}} \hat{x}^{[k]}(t)^\top Q \hat{x}^{[k]}(t)$$

for $N_{\mathrm{sim}} = 70$ reduces from 2.9447 to 2.8777, while the nominal input cost

$$\sum_{k=1}^{4} \sum_{t=0}^{N_{\mathrm{sim}}-1} \hat{u}^{[k]}(t)^\top R \hat{u}^{[k]}(t)$$

increases from 0.7132 to 0.7524.

6.5.2 Actuator selection

The goal of the following example is to demonstrate a simple application of the methods presented in this chapter to select a set of pneumatic actuators and corresponding compressors for the mass-spring-damper system shown in Figure 15: Actuators 1 and 2 are double-acting pneumatic cylinders, that provide force inputs to masses m_1 and m_3 respectively. The force acting on m_1 is $u_1 = A_1^{\mathrm{c}} P_{11}^{\mathrm{c}} - (A_1^{\mathrm{c}} - a_1^{\mathrm{c}}) P_{12}^{\mathrm{c}}$, where A_1^{c} and a_1^{c} are the cylinder bore and piston-rod cross-sectional areas respectively, and $(P_{11}^{\mathrm{c}}, P_{12}^{\mathrm{c}})$ are pressures acting on the left and right sides of the piston respectively. These pressures are given by the Pneumatic Valves module, which responds to a force command by adjusting the position of a 3/5-servo-valve [55]. This adjustment is performed to the pressure $\hat{P}_1$ of the airflow incoming from Compressor 1, which has the volumetric flowrate $q_1 = A_1^{\mathrm{c}} x_1^{\mathrm{v}}$ if $x_1^{\mathrm{v}} > 0$, and $q_2 = (A_1^{\mathrm{c}} - a_1^{\mathrm{c}}) x_1^{\mathrm{v}}$ otherwise (where x_1^{v} is the velocity of m_1). Hence, sizing the actuators for this system involves selecting the areas $(A_i^{\mathrm{c}}, a_i^{\mathrm{c}})$ of the actuators and compressor pressures $\hat{P}_i$, since they dictate the force limits on the system, and the compressor flowrate capacity required for actuation.

We consider a discretized model of the mass-spring-damper system, obtained using the forward Euler scheme with a time step of 0.1s. We assume that this model has additive disturbance inputs w acting on the velocity states, with $w \in W = 10^{-5}\mathcal{B}^3_\infty$. We equip the system with a static feedback gain K, that is the solution of the Riccati equation with matrices $Q = \mathbf{I}$ and $R = \mathbf{I}$. For this system, we compute an RPI set $\Delta\mathcal{X}$ as a polytope in $\mathbb{R}^6$ with 1812 faces using the method in [144]. The states of this system are constrained as

$$\mathcal{X} = \{x : |x_i^{\mathrm{p}}| \leq \hat{x}^{\mathrm{p}}, |x_i^{\mathrm{v}}| \leq \hat{x}_i^{\mathrm{v}},\ x_i^{\mathrm{p}} - x_{i+1}^{\mathrm{p}} \leq \hat{x}^{\mathrm{p}} - g^{\mathrm{p}}\},$$

where g^{p} is the minimum gap between the masses. In order to parametrize the input constraint set, we assume that the piston-rod cross-sectional area is constrained as $a_i^{\mathrm{c}} = 0.1A_i^{\mathrm{c}}$, and the compressors supply a constant pressure of $\hat{P}_i = 10^4$ N/m^2, following which we introduce the design vector $\epsilon := [A_1^{\mathrm{c}}\ A_2^{\mathrm{c}}]^\top$. Hence, U is parametrized as

$$\mathbb{U}(\epsilon) = \left\{u : \begin{bmatrix} -(A_1^{\mathrm{c}} - a_1^{\mathrm{c}})\hat{P}_1 \\ -A_2^{\mathrm{c}}\hat{P}_2 \end{bmatrix} \leq u \leq \begin{bmatrix} A_1^{\mathrm{c}}\hat{P}_1 \\ (A_2^{\mathrm{c}} - a_2^{\mathrm{c}})\hat{P}_2 \end{bmatrix}\right\}. \tag{6.26}$$

Then, to compute the vector ϵ, we use the criterion $\boldsymbol{f}(\epsilon) = \boldsymbol{f}_{\mathrm{M}}(\epsilon) + \boldsymbol{f}_{\mathrm{P}}(\epsilon)$, with $\boldsymbol{f}_{\mathrm{M}}$ and $\boldsymbol{f}_{\mathrm{P}}$ defined as follows:

- Since each A_i^{c} represents material costs, it is natural to consider the criterion $\boldsymbol{f}_{\mathrm{M}}(\epsilon) := A_1^{\mathrm{c}} + A_2^{\mathrm{c}}$ to minimize.

- Since velocities $|x_i^{\mathrm{v}}| \leq \hat{x}_i^{\mathrm{v}}$, corresponding maximum compressor flowrates are $\hat{q}_i = A_i^{\mathrm{c}}\hat{x}_i^{\mathrm{v}}$. We assume that the compressors are priced according to their maximum flowrate capacities as

Type	1	2	3	4	5
Capacity ($\times 10^{-4}$m^3/s)	0.1	0.15	0.2	0.25	0.35
Price ($\times 10^{-2}$)	1.0	1.1	1.2	1.3	1.4

  We encode this criterion as the piecewise constant function $\boldsymbol{f}_{\mathrm{P}}(\epsilon) := \beta_1 + \beta_2$, where $\beta_i = 1e^{-2}$ if $A_i^{\mathrm{c}}\hat{x}_i^{\mathrm{v}} \in [0, 0.1]e^{-4}$, $\beta_i = 1.1e^{-2}$ if $A_i^{\mathrm{c}}\hat{x}_i^{\mathrm{v}} \in (0.1, 0.15]e^{-4}$, $\beta_i = 1.2e^{-2}$ if $A_i^{\mathrm{c}}\hat{x}_i^{\mathrm{v}} \in (0.15, 0.2]e^{-4}$, $\beta_i = 1.3e^{-2}$ if $A_i^{\mathrm{c}}\hat{x}_i^{\mathrm{v}} \in (0.2, 0.25]e^{-4}$ and $\beta_i = 1.4e^{-2}$ if $A_i^{\mathrm{c}}\hat{x}_i^{\mathrm{v}} \in (0.25, 0.35]e^{-4}$. Then, the largest feasible area to ensure that the maximum feasible velocity can be reached is $\hat{A}_i^{\mathrm{c}} := 0.35e^{-4}/\hat{x}_i^{\mathrm{v}}$.

We also consider the loading effects of the actuators: compressibility of air modifies the stiffness and damping within the system [155]. Then, the modified dynamics of the system can be written as

$$x(t+1) = Ax(t) + Bu(t) + B_w w(t) + g(x(t), u(t), \epsilon).$$

To account for this modification, we follow Remark 6.3: we introduce the appended disturbance

$$\tilde{w} := B_w w + g(x, u, \epsilon),$$

such that $x(t+1) = Ax(t) + Bu(t) + \tilde{w}(t)$ is the modified system. Since $\mathcal{X}$, $\mathbb{U}(\epsilon)$ and $\mathcal{W}$ are compact, $\tilde{w}$ lies in a parametrized disturbance set, i.e., $\tilde{w} \in B_w \mathbb{W}(\epsilon)$. We use the scaling parametrization

$$\mathbb{W}(\epsilon) = (1 + \kappa_{\mathrm{d}}(\epsilon))\mathcal{W}, \;\text{ where }\; \kappa_{\mathrm{d}}(\epsilon) := 250(A_1^{\mathrm{c}} + A_2^{\mathrm{c}}).$$

In order to formulate $\mathbf{P}^{i,N}$ with the parametrized disturbance set $\mathbb{W}(\epsilon)$, we first note that the RPI set scales with $\mathbb{W}(\epsilon)$ as $(1 + \kappa_{\mathrm{d}}(\epsilon))\Delta\mathcal{X}$. Using positive homogeneity of support functions, we derive the formulation of $\mathbf{P}^{i,N}$ by replacing $h^{\Delta}, \bar{h}^{x}, \bar{\epsilon}$ and $q^{[i]}(\epsilon)$ in (6.24) by the parametrized versions

$$(1 + \kappa_{\mathrm{d}}(\epsilon))h^{\Delta}, (1 + \kappa_{\mathrm{d}}(\epsilon))\bar{h}^{x}, (1 + \kappa_{\mathrm{d}}(\epsilon))\bar{\epsilon} \;\text{ and }\; \bar{q}^{[i]}(\epsilon, \kappa_{\mathrm{d}}(\epsilon))$$

respectively. Hence, $\mathbf{P}^{i,N}$ is an optimization problem with linear constraints and a piecewise affine objective function. We formulate the resulting problem as a Mixed-Integer Linear Program (MILP) using the method presented in [36]. We consider the initial-condition set Ω to be the vertices of $\mathcal{X}$, with the modification $x_1^{\mathrm{v}}, x_3^{\mathrm{v}} = 0$. Then, we formulate and solve the projection problem in (6.22) to obtain an initial-condition set $\tilde{\Omega}$ for which $\mathbf{P}^{i,N}$ is feasible for all $N \geq 2$ as follows. Defining $\hat{\kappa}_{\mathrm{d}} := 250(\hat{A}_1^{\mathrm{c}} + \hat{A}_2^{\mathrm{c}})$, the largest possible parametrized RPI set is $(1 + \hat{\kappa}_{\mathrm{d}})\Delta\mathcal{X}$. Using this set, we compute the MPI set $\hat{\mathbb{O}}_{\infty}(\hat{\epsilon})$ and the 2-step stabilizable set $\hat{\mathbb{K}}_2(\hat{\epsilon}, \hat{\mathbb{O}}_{\infty}(\hat{\epsilon}))$ where $\hat{\epsilon} := [\hat{A}_1^{\mathrm{c}}\ \hat{A}_2^{\mathrm{c}}]^{\top}$, based on which we formulate problem in (6.22) with the constraint $\tilde{\Omega} \subseteq \hat{\mathbb{K}}_2(\hat{\epsilon}, \hat{\mathbb{O}}_{\infty}(\hat{\epsilon}))$. We choose $\tilde{\Omega}$ to be a set of vertices with the same cardinality as Ω, and the

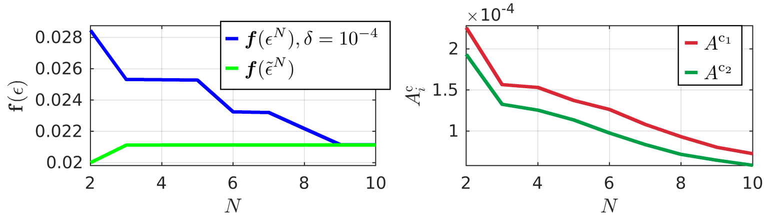


Figure 16: Numerical results for Example 2 with constraints $\hat{x}^{\mathrm{p}} = 0.8\mathrm{m}$, $\hat{x}_1^{\mathrm{v}}, \hat{x}_3^{\mathrm{v}} = 0.15\mathrm{m/s}$, $\hat{x}_2^{\mathrm{v}} = 1\mathrm{m/s}$, $g^{\mathrm{p}} = 10^{-3}\mathrm{m}$. (Left) Optimal input constraint set size $\boldsymbol{f}(\epsilon^N)$ computed by Algorithm 5, along with the lower bound $\boldsymbol{f}(\tilde{\epsilon}^N)$. (Right) Corresponding actuator areas in m^2. These values compose the parameter vector ϵ^N.

distance function to be the sum of 1-norm of the difference between the vertices, such that resulting problem is an LP. Using $\tilde{\Omega}$ and $\mathbf{P}^{i,N}$, we then compute optimal ϵ for different horizon lengths by applying Algorithm 5. The results are shown in Figure 16. We note that the upper bound to the termination index of Algorithm 5 is now computed using the largest possible parametrized disturbance set $(1 + \hat{\kappa}_{\mathrm{d}})\mathcal{W}$. For $\delta = 10^{-4}$, this bound equals $k_\delta = 1200$.

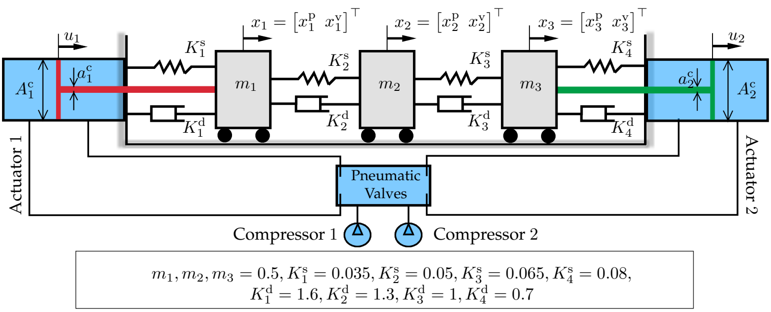


Figure 15: Schematic of the mass-spring-damper system. The Pneumatic Valves module regulates the compressor flow pressure. Units: m_i in Kg, K_i^{s} in N/m, K_i^{d} in Ns/m, $A_i^{\mathrm{c}}, a_i^{\mathrm{c}}$ in m^2.

In Figure 16 (left plot), we see that the results in Proposition 6.5 continue to hold, i.e., $\boldsymbol{f}(\epsilon^N)$ is non-increasing and convergent in N. This is because the stabilizable and admissible sets for a given input and disturbance set are nested sequences in N. This plot can be used as a tool to decide the minimum horizon length required given an upper bound on the budget allotted for pump selection. For example, if the maximum budget spent on pumps must not exceed 0.025 units, then the RMPC controller must have $N \geq 6$. In the right plot, the corresponding optimal actuator areas that can provide the required maximum flowrates $\hat{q}_i = A_i^{\mathrm{c}} \hat{x}_i^{\mathrm{v}}$ are shown, which correspond to the following optimal compressor types:

Horizon length N	2	3	4	5	6	7	8	9	10
Compressor 1 Type	5	4	4	4	3	3	2	2	2
Compressor 2 Type	5	3	3	3	2	2	2	1	1

6.5.3 Scalability of $\mathbf{P}^{i,N}$

The goal of the following example is to demonstrate scalability of problem $\mathbf{P}^{i,N}$. We consider the disturbance free system $x(t+1) = Ax(t)+u(t)$, where matrix A has diagonal components $A_{mm} = 1$, and off-diagonal components $A_{mn} = 0.01, m \neq n$. The system is subject to constraints $\mathcal{X} = \mathcal{B}_{\infty}^{n_x}$, and the initial-condition set is $\Omega = 0.2\mathcal{B}_{\infty}^{n_x}$. We parametrize the input constraint set as $\mathbb{U}(\epsilon) = \{u : -\epsilon\mathbf{1} \leq u \leq \epsilon\mathbf{1}\}$, where ϵ is a scalar, and choose $\boldsymbol{f}(\epsilon) = \epsilon$. We equip the system with a feedback controller K which is the solution of the Riccati equation for matrices $Q = \mathbf{I}$ and $R = 0.1\mathbf{I}$. Then, we use both vertex and hyperplane notations of Ω to formulate $\mathbf{P}^{i,N}$ with $i = 10$, $N \in [2, 10]$ and state-space dimension $n_x \in [2, 10]$. The resulting problems are LPs in both cases. The computational time spent by the solver and the number of variables and constraints, are shown in Figure 17.

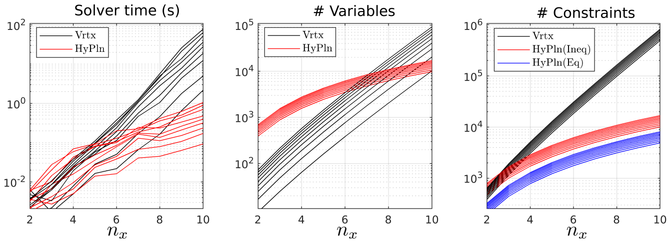


Figure 17: Numerical results for Example 3, demonstrating the solution time, number of variables, and number of constraints in the formulation of $\mathbf{P}^{i,N}$ when Ω is given in vertex and hyperplane notations. The different lines correspond to horizon lengths $N = 2, \cdots, 10$.

We observe that the dimension n_x significantly affects solver performance in the vertex notation, since the number of vertices in Ω increases exponentially with n_x. This issue is avoided if $\mathbf{P}^{i,N}$ is formulated using Ω in a hyperplane notation [100]. However, since the conditions used to encode the inclusion constraint $\Omega \subseteq \mathbb{K}_N(\epsilon, \mathbb{O}_i(\epsilon))$ are only sufficient, this might lead to conservative solutions $\boldsymbol{f}(\epsilon^{i,N})$. We report that in this example, there was no increase in conservativeness.

6.6 Comparison with parameteric sensitivity analysis-based approach

In this section, we compare our approach to computing the smallest input constraint set guaranteeing safe regulation with the parametric sensitivity analysis approach presented in [160]. We consider a slightly modified version of the approach presented in [160], by aiming to compute the smallest input constraint set rather than the largest disturbance set.

Given a *template* polytopic input constraint set $\tilde{\mathcal{U}} := \{u : F^u u \leq \mathbf{1}\}$, the approach of [160] can be adapted to compute the smallest scaling factor $\sigma \in [0, \bar{\sigma}]$ such that some given state $x \in \mathcal{X}$ in included in the MRCI set $\mathcal{X}_\infty(\sigma\tilde{\mathcal{U}})$. In other words, for each $x \in \mathcal{X}$, a function $V(x)$ is computed that satisfies

$$x \in \mathcal{X}_\infty(\sigma\tilde{\mathcal{U}}), \; \forall\, \sigma \in [V(x), \bar{\sigma}]. \tag{6.27}$$

Then, given a set of initial conditions Ω, the smallest input constraint set scaling factor required to robustly stabilize all $x(0) \in \Omega$ is given by

$$\hat{\sigma} := \max_{x \in \Omega} \; V(x).$$

In order to compute such a $V(x)$, we construct the MRCI set for the augumented system

$$\begin{aligned} &x(t+1) = Ax(t) + Bu(t) + B_w w(t), \quad u(t) \in \sigma\tilde{\mathcal{U}}, \quad w(t) \in \mathcal{W}, \\ &\sigma(t+1) = \sigma(t) \end{aligned}$$

subject to the constraints $(x, \sigma) \in \mathcal{X} \times [0, \bar{\sigma}]$. This set can be computed as the limit of the set iterations

$$\tilde{\boldsymbol{\mathcal{X}}}_0 := \left\{ \begin{pmatrix} x \\ \sigma \end{pmatrix} \;:\; H^x x \le h^x,\; \sigma \in [0, \bar{\sigma}] \right\} \tag{6.28a}$$

$$\tilde{\boldsymbol{\mathcal{X}}}_{t+1} := \left\{ \begin{pmatrix} x \\ \sigma \end{pmatrix} \in \tilde{\boldsymbol{\mathcal{X}}}_t \;:\; \exists\, u \in \sigma\tilde{\mathcal{U}} : \begin{pmatrix} Ax + Bu + B_w w \\ \sigma \end{pmatrix} \in \tilde{\boldsymbol{\mathcal{X}}}_t, \atop \forall\, w \in \mathcal{W} \right\} \tag{6.28b}$$

$$= \Pi_{(x,\sigma)} \left\{ \begin{pmatrix} x \\ \sigma \\ u \end{pmatrix} \;:\; \begin{bmatrix} \boldsymbol{M}_t^x A & \boldsymbol{M}_t^\sigma & \boldsymbol{M}_t^x B \\ \boldsymbol{0} & -\boldsymbol{1} & F^u \\ 0 & -1 & 0 \\ 0 & 1 & 0 \\ \boldsymbol{M}_t^x & \boldsymbol{M}_t^\sigma & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} x \\ \sigma \\ u \end{bmatrix} \le \begin{bmatrix} \boldsymbol{n}_t - h_{\mathcal{W}}(\boldsymbol{M}_t B_w) \\ \boldsymbol{0} \\ 0 \\ \bar{\sigma} \\ \boldsymbol{n}_t \end{bmatrix} \right\}, \tag{6.28c}$$

$$\text{where } \tilde{\boldsymbol{\mathcal{X}}}_t = \left\{ \begin{pmatrix} x \\ \sigma \end{pmatrix} \;:\; \boldsymbol{M}_t^x x + \boldsymbol{M}_t^\sigma \sigma \le \boldsymbol{n}_t \right\}. \tag{6.28d}$$

We hypothesize that the function $V(x)$ satisfying (6.27) is given by

$$V(x) = \min \left\{ \sigma : \big(x,\; \sigma\big) \in \tilde{\mathcal{X}}_\infty \right\}. \tag{6.29}$$

A proof of the aforementioned statement would require an extension to [160, Theorem 3], that we leave for future research. Then, the MRCI set for an input constraint set $\mathcal{U} = \sigma\tilde{\mathcal{U}}$ is given by

$$\mathcal{X}_\infty(\sigma\tilde{\mathcal{U}}) = \{x \;:\; \boldsymbol{M}_\infty^x x \le \boldsymbol{n}_\infty - \boldsymbol{M}_\infty^\sigma \sigma\}. \tag{6.30}$$

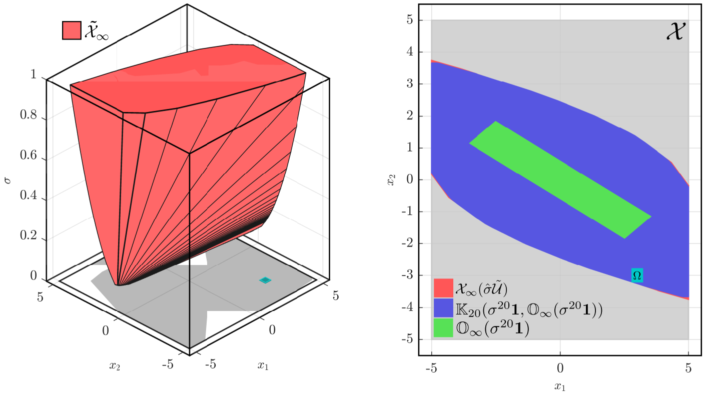


Figure 18: (Left) Set $\tilde{\mathcal{X}}_\infty$ obtained at termination of iterations in (6.28); (Right) Corresponding smallest MRCI set containing Ω. Also shown is the smallest 20-step stabilizable set that includes Ω, and driving each $x(0) \in \Omega$ to the MPI terminal set $\mathbb{O}_\infty(\sigma^{20}\mathbf{1})$ in atmost 20 timesteps.

Now considering the stabilizable set approximation of the MRCI set-based approach presented in this chapter, we note that for a given horizon length N, the smallest scaling factor required to robustly stabilize all initial states $x(0) \in \Omega$ is obtained as the solution of the optimization problem

$$\sigma^N := \min_{\sigma} \sigma \;\; \text{s.t.} \;\; \Omega \subseteq \mathbb{K}_N(\sigma\mathbf{1}, \mathbb{O}_\infty(\sigma\mathbf{1})), \; \delta\mathcal{B}_\infty^{n_u} \subseteq \sigma\tilde{\mathcal{U}} \ominus K\Delta\mathcal{X}, \qquad (6.31)$$

that can be solved exactly using a finite number of LPs according to Proposition 6.3. Moreover, σ^N is a nonincreasing sequence in N. However, since we fix the controller parameterization, we expect $\lim_{N\to\infty}\sigma^N > \hat{\sigma}$. On the other hand, defining

$$\hat{\sigma}_t = \min_{\sigma} \left\{\sigma : \boldsymbol{M}_t^x x + \boldsymbol{M}_t^\sigma \sigma \leq \boldsymbol{n}_t, \; \forall\, x \in \Omega\right\},$$

we observe that $\hat{\sigma}_t$ is a nondecreasing sequence in t, with $\hat{\sigma} = \lim_{t\to\infty}\hat{\sigma}_t$. Hence, in order to compute $\hat{\sigma}$, we require finite termination of the set iterations in (6.28).

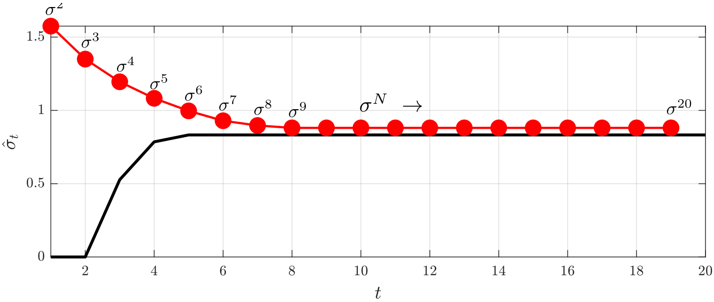


Figure 19: Comparison between the approach of [160] and the stabilizable set-based approach to compute the smallest regulating input constraint set.

Example

As an illustrative example, we consider the uncertain unstable system

$$x(t+1) \in \begin{bmatrix} 1.1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u(t) \oplus 0.1\mathcal{B}_\infty^{n_x},$$

subject to state constraints $\mathcal{X} = 5\mathcal{B}_\infty^{n_x}$. As a template input constraint set, we use the polytope $\tilde{\mathcal{U}} = \mathcal{B}_\infty^1$. In Figure 18, we plot the sets obtained using the approach of [160]. We note that due to instability, the slice of $\tilde{\boldsymbol{\mathcal{X}}}_\infty$ corresponding to $\sigma = 0$, i.e., $\{x : \boldsymbol{M}_\infty^x x \leq \boldsymbol{n}_\infty\} = \emptyset$. In order to compare with the approach presented in this chapter, we equip the system with a feedback gain K corresponding to LQR matrices $Q = \mathbf{I}$ and $R = \mathbf{I}$, and use an RPI set $\Delta\mathcal{X}$ parameterized as the minimal parameterized RPI set with 92 hyperplanes [97]. In Figure 19, we observe that as expected, σ^N approaches $\hat{\sigma}$ from above, with each σ^N being a feasible scaling factor. However, there exists a gap at termination because of the tube-MPC parameterization of the feedback control law.

Chapter 7

Data-driven synthesis of Robust Invariant Sets and Controllers

7.1 Introduction

In this chapter, we present a data-driven approach to identify an LTI model of a plant from a given dataset of state and input measurements, and synthesize an RMPC controller to robustly control the plant. In particular, we consider LTI plant models of the form

$$x(t+1) = Ax(t) + Bu(t) + w(t), \tag{7.1}$$

with state constraints $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$, input constraints $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$, and additive unknown but bounded disturbances $w \in \mathcal{W}$. This model is hence parameterized by $(A, B, \mathcal{W})$. For this model, we aim to synthesize a tube-based RMPC controller from [87], as discussed previously in Section 2.2.3. In order to synthesize this controller, in addition to the model $(A, B, \mathcal{W})$, we also require a feedback gain K and an RPI set in which the state of the system

$$x(t+1) = (A + BK)x(t) + w(t)$$

can be enforced to persistently belong. Moreover, we also require a positive invariant terminal set to guarantee recursive feasibility.

Typically, the system identification and controller synthesis phases are independent of each other. In order to perform system identification, i.e., identify a model $(A, B, \mathcal{W})$, physics-based, regression approaches and/or set-membership approaches [80, 64] can be used. Given an identified model $(A, B, \mathcal{W})$, one can then compute RPI sets using one of the several techniques available in literature. These sets can then be used to synthesize the RMPC controller. A drawback of this approach however is the following. In order to compute the RPI sets, one makes several parametric choices such as structure of the feedback laws, and representation of the RPI sets. Then, it can happen that the identified model $(A, B, \mathcal{W})$ identified a priori is not necessarily optimal for the control synthesis problem, i.e., there might exist some other model $(A, B, \mathcal{W})$ that can be identified from the dataset, and can be used to synthesize RPI sets with reduced conservativeness, this improving the controller performance. This was an observation also presented in [24], in which the authors demonstrated that by concurently selecting a system model while synthesizing RPI sets, one can obtain RPI sets with more favorable characteristics.

Based on this observation, the main contributions of the chapter are the following. Given a dataset of state and input measurements from the plant, we characterize a set of models $(A, B, \mathcal{W})$ that can describe the plant behavior, and use nonlinear matrix inequality (NLMI)-based results from [76] on RPI set computation to formulate a NonLinear Program with Matrix Inequalities (NLPMI) that selects a model (7.1) along with suitable RPI sets and a corresponding feedback matrix K. We then present a method to solve the NLPMI based on a Sequential Convex Programming (SCP) approach that we tailor to preserve feasibility of the iterates and satisfy a cost decrease condition. Finally, we demonstrate the efficacy of the method using a simple numerical example.

Alternative methods that directly compute feedback controllers using an implicit plant description based on measured trajectories were presented in [33, 9, 10, 30, 150]. In [13], these methods were used to synthesize controllers that induce robust invariance in a given polyhedral set. However, these methods cannot be used directly to select a model and synthesize RPI sets optimized for RMPC synthesis.

Notation: $\mathcal{P}(A, b) := \{x \in \mathbb{R}^n : -b \leq Ax \leq b\}$ is a symmetric polytope, and $\mathcal{E}(Q, r) := \{x \in \mathbb{R}^n : x^\top Q x \leq r\}$ is an ellipsoid. The set of m dimensional positive vectors is denoted as $\mathbb{R}^m_+$, positive definite $m \times m$ diagonal matrices is denoted as D^m_+, positive definite $m \times m$ symmetric matrices as $\mathbb{S}^m_+$. Given matrix $T \in \mathbb{R}^{n \times m}$,

$$\|T\|_\infty := \max_{i \in \mathbb{I}_1^n} \sum_{j=1}^{m} |T_{ij}|$$

is the ∞-norm of the matrix. We define $\|v\|_S^2 := v^\top S v$, and use $*$ to represent symmetrically identifiable matrix entries. We write

$$C_1\mathcal{P}(A_1, b_1) \oplus C_2\mathcal{P}(A_2, b_2) = [C_1\ C_2]\mathcal{P}(\mathrm{diag}(A_1, A_2), [b_1^\top b_2^\top]^\top),$$

where $\mathrm{diag}(A_1, A_2) := \begin{bmatrix} A_1 & \mathbf{0} \\ \mathbf{0} & A_2 \end{bmatrix}$ is a block-diagonal matrix.

7.2 Problem formulation

7.2.1 Background

We briefly recall the tube-based RMPC scheme from [87]. Given system (7.1), consider the nominal model $\hat{x}(t+1) = A\hat{x}(t) + B\hat{u}(t)$, and parameterize the plant input as

$$u(t) = \hat{u}(t) + K(x(t) - \hat{x}(t)),$$

where K is a static feedback gain. Assuming that the feedback gain is stabilizing for (A, B), the state error $\Delta x := x - \hat{x}$ with dynamics $\Delta x(t+1) = (A + BK)\Delta x(t) + w(t)$ belongs to the RPI set $\Delta\mathcal{X}$

$$\text{if } \Delta x(0) \in \Delta\mathcal{X}, \quad \text{and} \quad (A + BK)\Delta\mathcal{X} \oplus \mathcal{W} \subseteq \Delta\mathcal{X}. \tag{7.2}$$

Hence, x always belongs to the uncertainty tube with cross-section $\Delta\mathcal{X}$ around $\hat{x}$, i.e., $x(t) \in \hat{x}(t) \oplus \Delta\mathcal{X}, \forall t \geq 0$. The RMPC scheme then enforces

$\hat{x} \in \mathcal{X} \ominus \Delta\mathcal{X}$ and $\hat{u} \in \mathcal{U} \ominus K\Delta\mathcal{X}$, and computes

$$\mathbf{z} := \{\hat{x}(t), \ldots, \hat{x}(t+N), \hat{u}(t), \ldots, \hat{u}(t+N-1)\}$$

online given $x(t)$ by solving

$$\begin{aligned}
\min_{\mathbf{z}} \; & \sum_{s=t}^{t+N-1} \left\| \begin{bmatrix} \hat{x}(s)^\top & \hat{u}(s)^\top \end{bmatrix}^\top \right\|_{H_{\mathrm{Q}}}^2 + \|\hat{x}(t+N)\|_{P_{\mathrm{Q}}}^2 \\
\text{s.t. } \; & \hat{x}(s+1) = A\hat{x}(s) + B\hat{u}(s), && s \in \mathbb{I}_t^{t+N-1}, \\
& \hat{x}(s) \in \mathcal{X} \ominus \Delta\mathcal{X}, \; \hat{u}(s) \in \mathcal{U} \ominus K\Delta\mathcal{X}, && s \in \mathbb{I}_{t+1}^{t+N-1}, \\
& x(t) \in \hat{x}(t) \oplus \Delta\mathcal{X}, \;\; \hat{x}(t+N) \in \mathcal{X}^{\mathrm{t}}, && (7.3)
\end{aligned}$$

where $\mathcal{X}^{\mathrm{t}}$ is the terminal set. We assume that the set $\Delta\mathcal{X}$ is *small* enough for feasibility of Problem (7.3), i.e.,

$$\Delta\mathcal{X} \subset \mathcal{X}, \qquad K\Delta\mathcal{X} \subset \mathcal{U}, \tag{7.4}$$

and $\mathcal{X}^{\mathrm{t}}$ is a PI set for $\hat{x}(t+1) = (A+BK)\hat{x}(t)$ that satisfies

$$(A+BK)\mathcal{X}^{\mathrm{t}} \subseteq \mathcal{X}^{\mathrm{t}} \subseteq \mathcal{X} \ominus \Delta\mathcal{X}, \qquad K\mathcal{X}^{\mathrm{t}} \subseteq \mathcal{U} \ominus K\Delta\mathcal{X}. \tag{7.5}$$

Then the feasible set

$$\Omega_N := \{x : \text{(7.3) is feasible with } x(t) = x\}$$

is such that for each $x(t) \in \Omega_N$, there recursively exists an optimal solution

$$\mathbf{z}_* := \{\hat{x}_*(t), \ldots, \hat{x}_*(t+N), \hat{u}_*(t), \ldots, \hat{u}_*(t+N-1)\}$$

[87, Proposition 2]. Then, the input $u(t) := \hat{u}_*(t) + K(x(t) - \hat{x}_*(t))$ is applied to the plant. Moreover, if $(H_{\mathrm{Q}}, P_{\mathrm{Q}})$ are such that P_{Q} is the solution of the Discrete Algebraic Riccati equation (DARE) formulated using H_{Q} for the system (A, B), and K is corresponding optimal feedback gain, $\Delta\mathcal{X}$ is exponentially stable from every $x \in \Omega_N$ [87, Theorem 1].

7.2.2 Problem description

We consider a plant with dynamics

$$x(t+1) = f_{\mathrm{tr}}(x(t), u(t), v(t))$$

that is subject to bounded inputs $u(t)$ and unknown but bounded disturbances $v(t) \in \mathcal{V}_{\text{tr}} \subset \mathbb{R}^{n_v}$. Assuming that f_{tr} and $\mathcal{V}_{\text{tr}}$ are unknown, we collect a dataset

$$\mathcal{D} := \{x_{\text{D}}(t), u_{\text{D}}(t),\ t \in \mathbb{I}_1^T\}$$

of state-input measurements from the plant. Using $\mathcal{D}$, we propose a method to compute

$$(A, B, \mathcal{W}, K, \Delta\mathcal{X}, \mathcal{X}^{\text{t}})$$

that satisfy (7.1), (7.2), (7.4), (7.5) required for RMPC synthesis, while optimizing some criterion. In the sequel, we assume that we are interested in modeling the plant in a bounded subset $\Phi_* \subset \mathbb{R}^{n_x+n_u}$ of state-input vectors $[x^\top\ u^\top]^\top$. If f_{tr} is open-loop stable, then Φ_* can also represent the set of all possible state-input vectors that are reachable by the plant. Then, we assume that the set

$$\Phi_T := \{[x_{\text{D}}(t)^\top\ u_{\text{D}}(t))^\top]^\top, t \in \mathbb{I}_1^T\}$$

of measured state-input vectors is a subset of Φ_*, i.e., $\Phi_T \subseteq \Phi_*$. We also define

$$\mathcal{J}_* := \left\{ \begin{bmatrix} x \\ u \\ x_+ \end{bmatrix} : \begin{array}{c} x_+ = f_{\text{tr}}(x, u, v), \\ \forall\ [x^\top\ u^\top]^\top \in \Phi_*,\ \forall v \in \mathcal{V}_{\text{tr}} \end{array} \right\}.$$

Since Φ_*, $\mathcal{V}_{\text{tr}}$ and u are assumed to be bounded, $\mathcal{J}_*$ is also a bounded set. Finally, we denote the measured subset built using $\mathcal{D}$ as

$$\mathcal{J}_T := \{[x_{\text{D}}(t)^\top\ u_{\text{D}}(t)^\top\ x_{\text{D}}(t+1)^\top]^\top,\ t \in \mathbb{I}_1^{T-1}\} \subseteq \mathcal{J}_*$$

.

Remark 7.1. *Given (A, B, K), matrices $(H_{\text{Q}}, P_{\text{Q}})$ satisfying the DARE can be computed using the procedure in [162]. Hence, our approach involves tuning the MPC scheme. Combining our approach with other MPC tuning methods such as [86] is a subject of future research.* □

7.3 Identification based on invariant sets

To compute $(A, B, \mathcal{W}, K, \Delta\mathcal{X}, \mathcal{X}^{\text{t}})$ using the dataset $\mathcal{D}$ by optimization, we first characterize the set of models $(A, B, \mathcal{W})$ that are suitable to model the underlying plant.

7.3.1 Characterization of feasible models

We consider LTI models of the form in (7.1) to model the underlying plant, with the disturbance set parametrized as

$$\mathcal{W} := \mathcal{P}(F, d) = \{w : -d \leq Fw \leq d\}$$

We assume for simplicity that the normal vectors $\{F_i \in \mathbb{R}^{n_x},\ i \in \mathbb{I}_1^{m_W}\}$ are fixed a priori. Then, an LTI model (A, B, d) is suitable for RMPC synthesis if it can capture all possible state transitions of the plant as

$$\begin{aligned} f_{\mathrm{tr}}(x, u, v) \in Ax + Bu \oplus \mathcal{P}(F, d), \\ \forall\, [x^\top\ u^\top]^\top \in \Phi_*, \quad \forall\, v \in \mathcal{V}_{\mathrm{tr}}. \end{aligned} \tag{7.6}$$

Defining the prediction error $\zeta(A, B, z) := x_+ - Ax - Bu$ with

$$z := [x^\top\ u^\top\ x_+^\top]^\top,$$

a model (A, B, d) satisfies (7.6) if and only if

$$\zeta(A, B, z) \in \mathcal{P}(F, d), \qquad\qquad \forall\, z \in \mathcal{J}_*$$

by definition of $\mathcal{J}_*$. Hence,

$$\Sigma_* := \{(A, B, d) : \zeta(A, B, z) \in \mathcal{P}(F, d),\ \ \forall\ \ z \in \mathcal{J}_*\}$$

characterizes the set of all models (A, B, d) satisfying (7.6). However, we cannot construct Σ_* since we only access the measured subset $\mathcal{J}_T \subseteq \mathcal{J}_*$. Hence, we instead characterize

$$\Sigma_T(\theta_T) := \left\{ \begin{pmatrix} A, \\ B, \\ d \end{pmatrix} : \begin{array}{c} \zeta(A, B, z) \in \mathcal{P}(F, d - \kappa_T(A, B)\mathbf{1}) \\ \kappa_T(A, B) = \|F[-A\ \ -B\ \mathbf{I}]\|_\infty \theta_T, \\ d > \kappa_T(A, B)\mathbf{1},\ \forall\, z \in \mathcal{J}_T \end{array} \right\}$$

using $\mathcal{J}_T$, where $\theta_T := \mathbf{d}_\infty(\mathcal{J}_*, \mathcal{J}_T)$ is the Hausdorff distance between the sets $\mathcal{J}_T$ and $\mathcal{J}_*$ in ∞-norm, and is given by

$$\mathbf{d}_\infty(\mathcal{J}_*, \mathcal{J}_T) := \max_{z_* \in \mathcal{J}_*} \min_{z \in \mathcal{J}_T} \|z - z_*\|_\infty$$

since the inclusion $\mathcal{J}_T \subseteq \mathcal{J}_\infty$ holds for every $T > 0$.

Assumption 7.1. *(a) $\Sigma_T(\theta_T) \neq \emptyset$; (b) For all scalars $\theta > 0$, there exists some $\tilde{T} < \infty$ such that $\mathbf{d}_\infty(\mathcal{J}_*, \mathcal{J}_{\tilde{T}}) \leq \theta$.* $\square$

Assumption 7.1 implies that as $T \to \infty$, the set $\mathcal{J}_*$ is densely covered by $\mathcal{J}_T$: this is an assumption on the persistence of excitation of inputs, and bound-exploring property of the disturbances acting on the underlying plant [142, Section 3.2].

Theorem 7.1. *If Assumption 1 holds, then the inclusion* (7.6) *holds for all models $(A, B, d) \in \Sigma_T(\theta_T)$.* $\square$

Proof. We show that $\zeta(A, B, z) \in \mathcal{P}(F, d)$ for all $z \in \mathcal{J}_*$ and for all models $(A, B, d) \in \Sigma_T(\theta_T)$. For any $(A, B, d) \in \Sigma_T(\theta_T)$, clearly

$$\zeta(A, B, z) \in \mathcal{P}(F, d - \kappa_T(A, B)\mathbf{1}) \subset \mathcal{P}(F, d), \qquad \forall\, z \in \mathcal{J}_T.$$

By definition of the Hausdorff distance, for every remaining vector

$$\bar{z} \in \mathcal{J}_* \setminus \mathcal{J}_T := \{\tilde{z} : \tilde{z} \in \mathcal{J}_*, \tilde{z} \notin \mathcal{J}_T\},$$

there exists some measured vector $z \in \mathcal{J}_T$ satisfying $||z - \bar{z}||_\infty \leq \theta_T$. Then, for any $(A, B, d) \in \Sigma_T(\theta_T)$,

$$\begin{aligned}
F\zeta(A, B, \bar{z}) &= F(\zeta(A, B, \bar{z}) - \zeta(A, B, z) + \zeta(A, B, z)) \\
&\leq F[-A \ \ -B\ \mathbf{I}](\bar{z} - z) + d - \kappa_T(A, B)\mathbf{1} \\
&\leq ||F[-A \ \ -B\ \mathbf{I}](\bar{z} - z)||_\infty \mathbf{1} + d - \kappa_T(A, B)\mathbf{1} \\
&\leq ||F[-A \ \ -B\ \mathbf{I}]||_\infty \, \|(\bar{z} - z)\|_\infty \, \mathbf{1} + d - \kappa_T(A, B)\mathbf{1} \\
&\leq ||F[-A \ \ -B\ \mathbf{I}]||_\infty \theta_T \mathbf{1} + d - \kappa_T(A, B)\mathbf{1} = d,
\end{aligned}$$

where the second step follows from definition of $\Sigma_T(\theta_T)$, and third step from the definition of ∞-norm, the fourth step from the Cauchy-Schwarz inequality, and the fifh step from the the inequality $||\bar{z} - z||_\infty \leq \theta_T$. Using similar arguments, the condition $-d \leq F\zeta(A, B, \bar{z})$ follows, thus concluding that $\zeta(A, B, \bar{z}) \in \mathcal{P}(F, d), \forall\, \bar{z} \in \mathcal{J}_* \setminus \mathcal{J}_T$. $\square$

Theorem 7.1 implies that every $(A, B, d) \in \Sigma_T(\theta_T) \subseteq \Sigma_*$ is a feasible model for RMPC synthesis. However, $\Sigma_T(\theta_T)$ cannot be constructed from data since θ_T is unknown. To tackle this issue, we follow the standard approach of inflating the disturbance set using some parameter (e.g., [142]): we propose to select some $\hat{\theta}_T > 0$, and approximate $\Sigma_T(\theta_T)$ with $\hat{\Sigma}_T := \Sigma_T(\hat{\theta}_T)$ under the following assumption.

Assumption 7.2. $\hat{\theta}_T \geq \theta_T = \mathbf{d}_\infty(\mathcal{J}_*, \mathcal{J}_T)$. □

Under Assumption 7.2, we have $\hat{\Sigma}_T \subseteq \Sigma_T(\theta_T)$. Hence, every uncertain LTI model $(A, B, d) \in \hat{\Sigma}_T$ is suitable for RMPC synthesis. We assume in the sequel that Assumption 7.2 is satisfied by some user-defined $\hat{\theta}_T$. We then encode $\hat{\Sigma}_T$ with linear constraints as

$$\hat{\Sigma}_T = \left\{ \begin{pmatrix} A, \\ B, \\ d \end{pmatrix} : \begin{array}{c} \zeta(A, B, z) \in \mathcal{P}(F, d - \lambda\hat{\theta}_T \mathbf{1}), d > \lambda\hat{\theta}_T \mathbf{1}, \\ -\mathcal{Z} \leq F[-A \ \ -B \ \mathbf{I}] \leq \mathcal{Z}, \mathcal{Z} \geq \mathbf{0}, \\ \sum_{j=1}^{2n_x+n_u} \mathcal{Z}_{ij} \leq \lambda, \forall\, i \in \mathbb{I}_1^{m_W}, \forall\, z \in \mathcal{J}_T \end{array} \right\}$$

using the definition of ∞-norm for matrices, where $\mathcal{Z} \in \mathbb{R}_+^{m_W \times (2n_x+n_u)}$ is a slack variable matrix. We reiterate that since Assumption 7.2 cannot be verified directly using data, robustness guarantees with respect to the underlying plant can only be provided in theory. However, if Assumption 7.1(b) holds, the distance $\theta_T \to 0$ for large T. Hence, guessing some $\hat{\theta}_T \approx 0$ can satisfy Assumption 7.2 for large datasets. Moreover, the validity of a given $\hat{\theta}_T$ can be checked by verifying the existence of a model $(A, B, d) \in \hat{\Sigma}_T$ explaining a validation dataset. On the other hand, computation of a $\hat{\theta}_T$ satisfying Assumption 7.2 is a fundamental issue in data-driven methods: while statistical techniques such as, e.g., bootstrapping can be used, the development of such methods is a future research subject.

Remark 7.2. *(a) In [142], an optimal LTI model set is first computed, from which a model is selected and then feedback controllers are synthesized. We combine all three phases in the current work; (b) In [10], the closed-loop dynamics of an unknown LTI plant with a known disturbance set is characterized in terms of the measured dataset, and parametrized by unknown but bounded disturbance sequences. Then, a controller is synthesized for all feasible LTI models. We instead use a model-dependent disturbance set. While the assumption of a known disturbance set is as strict as Assumption 7.2, comparison with [10] is a subject of future research.* □

7.3.2 Robust PI set design

We will now compute a feedback gain K and corresponding invariant sets $\Delta\mathcal{X}$ and $\mathcal{X}^{\mathrm{t}}$ for some $(A, B, d) \in \hat{\Sigma}_T$. To this end, we parametrize the

RPI set as

$$\Delta\mathcal{X} = \mathcal{P}(\underline{P}, \underline{b}), \underline{b} \in \mathbb{R}^{m}_{+},$$

the PI terminal set as

$$\mathcal{X}^{\mathrm{t}} = \mathcal{P}(\bar{P}, \bar{b}), \bar{b} \in \mathbb{R}^{\bar{m}}_{+},$$

and assume that the constraint sets are

$$\mathcal{X} = \mathcal{P}(V^x, v^x), v^x \in \mathbb{R}^{m_x}_{+}, \qquad \mathcal{U} = \mathcal{P}(V^u, v^u), v^u \in \mathbb{R}^{m_u}_{+}.$$

Then, for some $(A, B, d) \in \hat{\Sigma}_T$, if $(K, \underline{P}, \underline{b}, \bar{P}, \bar{b})$ satisfies

$$(A + BK)\mathcal{P}(\underline{P}, \underline{b}) \oplus \mathcal{P}(F, d) \subseteq \mathcal{P}(\underline{P}, \underline{b}), \tag{7.7a}$$

$$(A + BK)\mathcal{P}(\bar{P}, \bar{b}) \subseteq \mathcal{P}(\bar{P}, \bar{b}), \tag{7.7b}$$

$$\mathcal{P}(\underline{P}, \underline{b}) \oplus \mathcal{P}(\bar{P}, \bar{b}) \subseteq \mathcal{X}, \tag{7.7c}$$

$$K\mathcal{P}(\underline{P}, \underline{b}) \oplus K\mathcal{P}(\bar{P}, \bar{b}) \subseteq \mathcal{U}, \tag{7.7d}$$

it can be used to synthesize the RMPC scheme (since (7.7a) implies (7.2), (7.7b) implies (7.5), (7.7c)-(7.7d) imply the constraint inclusions in (7.4)-(7.5)). We encode (7.7a)-(7.7d) using Theorem 7.2.

Theorem 7.2. *[76, Theorem 2] For some $C \in \mathbb{R}^{n \times n^c}$, $M^c \in \mathbb{R}^{m^c \times n}$, $b^c \in \mathbb{R}^{m^c}_{+}$, $M^0 \in \mathbb{R}^{m^0 \times n}$, $b^o \in \mathbb{R}^{m^o}_{+}$, the inclusion $C\mathcal{P}(M^c, b^c) \subseteq \mathcal{P}(M^0, b^0)$ holds if*

$$\forall\, i \in \mathbb{I}_1^{m^0}, \qquad \exists L^c_{[i]} \in \mathbb{D}^{m^c}_{+}$$

such that

$$\begin{bmatrix} 2b^0_i - b^{c\top} L^c_{[i]} b^c & M^0_i C \\ * & M^{c\top} L^c_{[i]} M^c \end{bmatrix} \succ 0.$$

□

Remark 7.3. *The condition in Theorem 7.2 is necessary and sufficient for the inclusion $C\mathcal{P}(M^c, b^c) \subseteq \mathcal{P}(M^0, b^0)$ if a non-strict inequality $\succeq$ is used. However, we only use the sufficiency property given by $\succ$ for numerical robustness.* □

Hence, (7.7a) $\Longleftarrow$ $\forall i \in \mathbb{I}_1^{m}, \exists \underline{D}_{[i]} \in D^{m}_{+}, W_{[i]} \in D^{mw}_{+}$ s.t.

$$\begin{bmatrix} 2\underline{b}_i - \underline{b}^\top \underline{D}_{[i]} \underline{b} - d^\top W_{[i]} d & \underline{P}_i & \underline{P}_i(A + BK) \\ * & F^\top W_{[i]} F & \mathbf{0} \\ * & * & \underline{P}^\top \underline{D}_{[i]} \underline{P} \end{bmatrix} \succ 0, \tag{7.8}$$

(7.7b) $\Longleftarrow\ \forall\, i \in \mathbb{I}_1^{\bar{m}}, \exists\, \bar{D}_{[i]} \in D_+^{\bar{m}}$ s.t.

$$\begin{bmatrix} 2\bar{b}_i - \bar{b}^\top \bar{D}_{[i]}\bar{b} & \bar{P}_i(A+BK) \\ * & \bar{P}^\top \bar{D}_{[i]}\bar{P} \end{bmatrix} \succ 0, \tag{7.9}$$

(7.7c) $\Longleftarrow\ \forall\, i \in \mathbb{I}_1^{m_x}, \exists\, \underline{S}_{[i]} \in D_+^{\underline{m}}, \bar{S}_{[i]} \in D_+^{\bar{m}}$ s.t.

$$\begin{bmatrix} 2v_i^x - \underline{b}^\top \underline{S}_{[i]}\underline{b} - \bar{b}^\top \bar{S}_{[i]}\bar{b} & V_i^x & V_i^x \\ * & \underline{P}^\top \underline{S}_{[i]}\underline{P} & \mathbf{0} \\ * & * & \bar{P}^\top \bar{S}_{[i]}\bar{P} \end{bmatrix} \succ 0, \tag{7.10}$$

(7.7d) $\Longleftarrow\ \forall\, i \in \mathbb{I}_1^{m_u}, \exists\, \underline{R}_{[i]} \in D_+^{\underline{m}}, \bar{R}_{[i]} \in D_+^{\bar{m}}$ s.t.

$$\begin{bmatrix} 2v_i^u - \underline{b}^\top \underline{R}_{[i]}\underline{b} - \bar{b}^\top \bar{R}_{[i]}\bar{b} & V_i^u K & V_i^u K \\ * & \underline{P}^\top \underline{R}_{[i]}\underline{P} & \mathbf{0} \\ * & * & \bar{P}^\top \bar{R}_{[i]}\bar{P} \end{bmatrix} \succ 0. \tag{7.11}$$

We now formulate a criterion to select the variables formulating (7.8)-(7.11) along with $(A, B, d) \in \hat{\Sigma}_T$, leading to an optimization problem. In this formulation, we assume that the matrices $(\underline{P}, \bar{P}, F)$ are known a priori. While this assumption increases conservativeness in our approach, it simplifies the solution procedure. We note that a good set of hyperplanes $(\underline{P}, \bar{P})$ can *guessed* for some initial (A, B, F, d) using [76], and kept constant for our approach. Moreover, our approach can be extended to optimize over $(\underline{P}, \bar{P})$ using the results in [73]. However, we skip further details here.

7.3.3 Identification criterion

For RMPC synthesis, it is desirable to compute a small RPI set $\Delta\mathcal{X}$ to reduce constraint tightening, and to regulate the system to a small neighborhood of the origin [87]. Hence, we minimize $\|\underline{b}\|_1$, since it corresponds to computing the smallest (in an inclusion sense) RPI set represented by fixed hyperplanes $\underline{P}$ [118, Corollary 1].

Moreover, we know from [87, Proposition 2] that a large terminal set $\mathcal{X}^{\mathrm{t}}$ maximizes the region of attraction Ω_N. Hence, we maximize the size of $\mathcal{X}^{\mathrm{t}}$ by minimizing a distance metric between $\mathcal{X}^{\mathrm{t}}$ and the state constraint set $\mathcal{X}$ as follows: let $\mathcal{B}(\bar{\epsilon}) := \mathcal{P}(\bar{E}, \bar{\epsilon}) \subset \mathbb{R}^{n_x}$ with $\bar{\epsilon} \in \mathbb{R}_+^{m_\epsilon}$ and $\bar{E}$ fixed a priori; then, we minimize $\|\bar{\epsilon}\|_1$ subject to the inclusion $\mathcal{X} \subseteq \mathcal{P}(\bar{P}, \bar{b}) \oplus \mathcal{B}(\bar{\epsilon})$. Assuming to know the vertices $\{x_{[i]}, i \in \mathbb{I}_1^{m_x^v}\}$ of $\mathcal{X}$, $(\bar{b}, \bar{\epsilon}) \in \bar{\mathcal{S}}$ implies $\mathcal{X} \subseteq \mathcal{P}(\bar{P}, \bar{b}) \oplus \mathcal{B}(\bar{\epsilon})$ where

$$\bar{\mathcal{S}} := \{(\bar{b}, \bar{\epsilon}) : \; x_{[i]} \in \mathcal{P}(\bar{P}, \bar{b}) \oplus \mathcal{P}(\bar{E}, \bar{\epsilon}), \forall\, i \in \mathbb{I}_1^{m_x^v}\}.$$

Finally, since the performance matrices $(H_{\mathrm{Q}}, P_{\mathrm{Q}})$ formulating the RMPC controller in (7.3) are fixed by (A, B, K) as noted in Remark 7.1, we introduce a way to tune the closed-loop performance: We evaluate the performance using the system $\hat{x}(t+1) = A\hat{x}(t) + B\hat{u}(t)$ inside the terminal set as

$$\hat{x}(0) \in \mathcal{X}^{\mathrm{t}}, \; \hat{u}(t) = K\hat{x}(t), \; \sum_{t=0}^{\infty} \|\hat{x}(t)\|_{\tilde{Q}}^2 + \|\hat{u}(t)\|_{\tilde{R}}^2 \leq \tilde{r}, \tag{7.12}$$

where $\tilde{Q} \in \mathbb{S}_+^{n_x}$ and $\tilde{R} \in \mathbb{S}_+^{n_u}$ are user-defined performance matrices, and we minimize $\tilde{r}$. Then, if $\tilde{\Theta} \in \mathbb{S}_+^{n_x}$ satisfies

$$(A + BK)^\top \tilde{\Theta}(A + BK) - \tilde{\Theta} + \tilde{Q} + K^\top \tilde{R} K \prec 0, \tag{7.13}$$

the left-hand-side of the inequality in (7.12) is upper bounded by $\|\hat{x}(0)\|_{\tilde{\Theta}}^2$ [65]. Hence, (7.12) is satisfied if the inclusion $\mathcal{P}(\bar{P}, \bar{b}) \subseteq \mathcal{E}(\tilde{\Theta}, \tilde{r})$ holds, thus imposing an upper bound on the size of the terminal set. Following the S-procedure [19, Section 2.6.3], the inclusion $\mathcal{P}(\bar{P}, \bar{b}) \subseteq \mathcal{E}(\tilde{\Theta}, \tilde{r})$ holds if

$$\exists \tilde{M} \in D_+^{\bar{m}} \quad \text{s.t.} \quad \bar{P}^\top \tilde{M} \bar{P} - \tilde{\Theta} \succ 0, \quad \tilde{r} - \bar{b}^\top \tilde{M} \bar{b} > 0. \tag{7.14}$$

Based on these considerations, we formulate the identification problem as the following NLPMI

$$\begin{aligned} &\min_{Z_{\mathrm{NL}}} \alpha \|\underline{b}\|_1 + \beta \|\bar{\epsilon}\|_1 + \gamma \tilde{r} \\ &\quad \text{s.t. } (A, B, d) \in \hat{\Sigma}_T, \; (\bar{b}, \bar{\epsilon}) \in \bar{\mathcal{S}}, \; (7.8) - (7.11), (7.13) - (7.14) \end{aligned} \tag{7.15}$$

where $\alpha, \beta, \gamma \geq 0$ are user-defined weights, and

$$Z_{\mathrm{NL}} := \begin{pmatrix} A, B, d, \mathcal{Z}, \lambda, K, \underline{b}, \bar{b}, \tilde{\Theta}, \tilde{r}, \tilde{M}, \bar{\epsilon}, \{\underline{D}_{[i]}, W_{[i]}, i \in \mathbb{I}_1^{m}\}, \\ \{\bar{D}_{[i]}, i \in \mathbb{I}_1^{\bar{m}}\}, \{\underline{S}_{[i]}, \bar{S}_{[i]}, i \in \mathbb{I}_1^{m_x}\}, \{\underline{R}_{[i]}, \bar{R}_{[i]}, i \in \mathbb{I}_1^{m_u}\} \end{pmatrix}.$$

7.3.4 Feasible SCP for Problem (7.15)

In order to solve Problem (7.15), a standard SCP approach can be adopted, in which a sequence of SDPs approximating (7.15) are solved. However, to guarantee feasibility of the iterates, we adopt the following SCP procedure. Starting from an initial feasible iterate Z_{NL}, we solve a sequence of SDPs formulated using sufficient LMI conditions for the constraints of Problem (7.15), such that the method produces feasible iterates. The sufficient LMI conditions are formulated using convex underestimates [143] of the NLMI constraints at the current iterate. Moreover, the objective value of (7.15) is non-increasing over the iterates, such that globalization is unnecessary and we terminate when the objective value does not reduce further.

Convex SDP approximation

Given a feasible iterate Z_{NL} for Problem (7.15), we formulate sufficient LMI conditions for (7.8)-(7.11), (7.13), (7.14) using the following result.

Proposition 7.1. *[76, Lemma 2.1] Let matrices $\boldsymbol{L}, L \in \mathbb{R}^{m\times n}$ and $\boldsymbol{D}, D \in \mathbb{S}_+^m$, and define the matrix functions*

$$\mathcal{L}_{\boldsymbol{L},\boldsymbol{D}}^{L,D} := \boldsymbol{L}^\top D^{-1}L + L^\top D^{-1}\boldsymbol{L} - L^\top D^{-1}\boldsymbol{D}D^{-1}L,$$

and

$$\mathcal{N}_{\boldsymbol{L},\boldsymbol{D}} := \boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{L}.$$

Then, $\mathcal{N}_{\mathbf{L},\mathbf{D}} \succeq \mathcal{L}_{\mathbf{L},\mathbf{D}}^{L,D}$ and $\mathcal{N}_{L,D} = \mathcal{L}_{L,D}^{L,D}$. Hence, if there exists (L,D) such that $\mathcal{N}_{L,D} \succ 0$, then there exists $(\boldsymbol{L},\boldsymbol{D})$ such that $\mathcal{N}_{\boldsymbol{L},\boldsymbol{D}} \succeq \mathcal{L}_{\boldsymbol{L},\boldsymbol{D}}^{L,D} \succ 0$. □

This result implies that if $\mathcal{N}_{L,D} \succ 0$, then the LMI $\mathcal{L}_{\boldsymbol{L},\boldsymbol{D}}^{L,D} \succ 0$ is a convex underestimate and a sufficient condition for $\mathcal{N}_{\boldsymbol{L},\boldsymbol{D}} \succ 0$. We will now use this property to formulate sufficient LMIs for (7.8)-(7.11), (7.13), (7.14). The claimed SCP feasibility and cost decrease are then obtained as a corollary.

Theorem 7.3. *Suppose that Z_{NL} is feasible for* (7.15). *Then:*

(i) RPI condition (7.8)*: For each $i \in \mathbb{I}_1^m$, there exists $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{d}, \boldsymbol{K}, \underline{\boldsymbol{b}}, \hat{\underline{\boldsymbol{D}}}_{[i]}, \hat{\boldsymbol{W}}_{[i]})$ satisfying the LMI*

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & -\boldsymbol{B}^\top \underline{P}_i^\top & \mathbf{0} & \boldsymbol{K} \\ * & \hat{\underline{\boldsymbol{D}}}_{[i]} & \mathbf{0} & \underline{\boldsymbol{b}} & \mathbf{0} & \mathbf{0} \\ * & * & \hat{\boldsymbol{W}}_{[i]} & \boldsymbol{d} & \mathbf{0} & \mathbf{0} \\ * & * & * & 2\underline{\boldsymbol{b}}_i + \mathcal{L}_{\boldsymbol{B}^\top \underline{P}_i^\top, \mathbf{I}}^{B^\top \underline{P}_i^\top, \mathbf{I}} & \underline{P}_i & \underline{P}_i \boldsymbol{A} \\ * & * & * & * & \mathcal{L}_{F, \hat{\boldsymbol{W}}_{[i]}}^{F, W_{[i]}^{-1}} & \mathbf{0} \\ * & * & * & * & * & \mathcal{L}_{\underline{P}, \hat{\underline{\boldsymbol{D}}}_{[i]}}^{\underline{P}, \underline{D}_{[i]}^{-1}} + \mathcal{L}_{\boldsymbol{K}, \mathbf{I}}^{K, \mathbf{I}} \end{bmatrix} \succ 0, \qquad (7.16)$$

and $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{d}, \boldsymbol{K}, \underline{\boldsymbol{b}}, \hat{\underline{\boldsymbol{D}}}_{[i]}^{-1}, \hat{\boldsymbol{W}}_{[i]}^{-1})$ *satisfies* (7.8).

(ii) PI condition (7.9)*: The following LMI is satisfied by some $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{K}, \bar{\boldsymbol{b}}, \hat{\bar{\boldsymbol{D}}}_{[i]})$ for each $i \in \mathbb{I}_1^{\bar{m}}$:*

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & -\boldsymbol{B}^\top \bar{P}_i^\top & \boldsymbol{K} \\ * & \hat{\bar{\boldsymbol{D}}}_{[i]} & \bar{\boldsymbol{b}} & \mathbf{0} \\ * & * & 2\bar{\boldsymbol{b}}_i + \mathcal{L}_{\boldsymbol{B}^\top \bar{P}_i^\top, \mathbf{I}}^{B^\top \bar{P}_i^\top, \mathbf{I}} & \bar{P}_i \boldsymbol{A} \\ * & * & * & \mathcal{L}_{\bar{P}, \hat{\bar{\boldsymbol{D}}}_{[i]}}^{\bar{P}, \bar{D}_{[i]}^{-1}} + \mathcal{L}_{\boldsymbol{K}, \mathbf{I}}^{K, \mathbf{I}} \end{bmatrix} \succ 0, \qquad (7.17)$$

and $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{K}, \bar{\boldsymbol{b}}, \hat{\bar{\boldsymbol{D}}}_{[i]}^{-1})$ *satisfies* (7.9).

(iii) Constraint inclusions (7.10), (7.11)*: For each $i \in \mathbb{I}_1^{m_x}$, there exists $(\underline{\boldsymbol{b}}, \bar{\boldsymbol{b}}, \hat{\underline{\boldsymbol{S}}}_{[i]}, \hat{\bar{\boldsymbol{S}}}_{[i]})$, and for each $i \in \mathbb{I}_1^{m_u}$, there exists $(\boldsymbol{K}, \underline{\boldsymbol{b}}, \bar{\boldsymbol{b}}, \hat{\underline{\boldsymbol{R}}}_{[i]}, \hat{\bar{\boldsymbol{R}}}_{[i]})$ satisfy-*

ing the LMIs

$$\begin{bmatrix} \underline{\hat{\boldsymbol{S}}}_{[i]} & \mathbf{0} & \underline{\boldsymbol{b}} & \mathbf{0} & \mathbf{0} \\ * & \hat{\bar{\boldsymbol{S}}}_{[i]} & \bar{\boldsymbol{b}} & \mathbf{0} & \mathbf{0} \\ * & * & 2v_i^x & V_i^x & V_i^x \\ * & * & * & \mathcal{L}_{\underline{P},\underline{\hat{\boldsymbol{S}}}_{[i]}}^{\underline{P},\underline{S}_{[i]}^{-1}} & \mathbf{0} \\ * & * & * & * & \mathcal{L}_{\bar{P},\hat{\bar{\boldsymbol{S}}}_{[i]}}^{\bar{P},\bar{S}_{[i]}^{-1}} \end{bmatrix} \succ 0, \tag{7.18}$$

$$\begin{bmatrix} \underline{\hat{\boldsymbol{R}}}_{[i]} & \mathbf{0} & \underline{\boldsymbol{b}} & \mathbf{0} & \mathbf{0} \\ * & \hat{\bar{\boldsymbol{R}}}_{[i]} & \bar{\boldsymbol{b}} & \mathbf{0} & \mathbf{0} \\ * & * & 2v_i^u & V_i^u\boldsymbol{K} & V_i^u\boldsymbol{K} \\ * & * & * & \mathcal{L}_{\underline{P},\underline{\hat{\boldsymbol{R}}}_{[i]}}^{\underline{P},\underline{R}_{[i]}^{-1}} & \mathbf{0} \\ * & * & * & * & \mathcal{L}_{\bar{P},\hat{\bar{\boldsymbol{R}}}_{[i]}}^{\bar{P},\bar{R}_{[i]}^{-1}} \end{bmatrix} \succ 0, \tag{7.19}$$

and $(\underline{\boldsymbol{b}}, \bar{\boldsymbol{b}}, \underline{\hat{\boldsymbol{S}}}_{[i]}^{-1}, \hat{\bar{\boldsymbol{S}}}_{[i]}^{-1})$ *,* $(\boldsymbol{K}, \underline{\boldsymbol{b}}, \bar{\boldsymbol{b}}, \underline{\hat{\boldsymbol{R}}}_{[i]}^{-1}, \hat{\bar{\boldsymbol{R}}}_{[i]}^{-1})$ *satisfy* (7.10)*,* (7.11)*.*

(iv) Dissipativity condition (7.13)*: There exists* $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{K}, \tilde{\boldsymbol{\Theta}})$ *satisfying the LMI*

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & -\boldsymbol{B}^\top & \boldsymbol{K} \\ * & \tilde{Q}^{-1} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ * & * & \tilde{R}^{-1} & \mathbf{0} & \boldsymbol{K} \\ * & * & * & \mathcal{L}_{\mathbf{I},\tilde{\boldsymbol{\Theta}}}^{\mathbf{I},\tilde{\Theta}} + \mathcal{L}_{\boldsymbol{B}^\top,\mathbf{I}}^{B^\top,\mathbf{I}} & \boldsymbol{A} \\ * & * & * & * & \tilde{\boldsymbol{\Theta}} + \mathcal{L}_{\boldsymbol{K},\mathbf{I}}^{K,\mathbf{I}} \end{bmatrix} \succ 0, \tag{7.20}$$

and $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{K}, \tilde{\boldsymbol{\Theta}})$ *satisfies* (7.13)*.*

(v) Performance ellipsoid inclusion condition (7.14)*: There exists* $(\bar{\boldsymbol{b}}, \tilde{\boldsymbol{\Theta}}, \tilde{\boldsymbol{r}}, \hat{\tilde{\boldsymbol{M}}})$ *satisfying the LMI*

$$\mathcal{L}_{\bar{P},\hat{\tilde{\boldsymbol{M}}}}^{\bar{P},\tilde{M}^{-1}} - \tilde{\boldsymbol{\Theta}} \succ 0, \quad \begin{bmatrix} \hat{\tilde{\boldsymbol{M}}} & \bar{\boldsymbol{b}} \\ * & \tilde{\boldsymbol{r}} \end{bmatrix} \succ 0, \tag{7.21}$$

and $(\bar{\boldsymbol{b}}, \tilde{\boldsymbol{\Theta}}, \tilde{\boldsymbol{r}}, \hat{\tilde{\boldsymbol{M}}}^{-1})$ *satisfies* (7.14)*.*

Proof. The proof follows by using the Schur complement and Proposition 7.1 on (7.8)-(7.11). We detail the proof of Part (i), since Parts (ii)-(v)

follow with similar arguments.
Part $(i):$ As $(A, B, K, \underline{b}, d, \underline{D}_{[i]}, W_{[i]})$ in Z_{NL} satisfy (7.8), we take a Schur complement of the $(1,1)$ block to obtain

$$\begin{bmatrix} \underline{D}_{[i]}^{-1} & \mathbf{0} & \underline{b} & \mathbf{0} & \mathbf{0} \\ * & W_{[i]}^{-1} & d & \mathbf{0} & \mathbf{0} \\ * & * & 2\underline{b}_i & \underline{P}_i & \underline{P}_i A + \underline{P}_i B K \\ * & * & * & F^\top W_{[i]} F & \mathbf{0} \\ * & * & * & * & \underline{P}^\top \underline{D}_{[i]} \underline{P} \end{bmatrix} \succ 0. \tag{7.22}$$

Defining $\hat{W}_{[i]} := W_{[i]}^{-1}$ and $\hat{\underline{D}}_{[i]} := \underline{D}_{[i]}^{-1}$, Eq. (7.22) is nonlinear in variables $(B, K, \hat{\underline{D}}_{[i]}, \hat{W}_{[i]})$ in the blocks $(4,4)$, $(5,5)$, $(3,5)$ and $(5,3)$. Then, we write (7.22) as

$$\begin{bmatrix} \hat{\underline{D}}_{[i]} & \mathbf{0} & \underline{b} & \mathbf{0} & \mathbf{0} \\ * & \hat{W}_{[i]} & d & \mathbf{0} & \mathbf{0} \\ * & * & \underline{\mathcal{N}}_{i,33} & \underline{P}_i & \underline{P}_i A \\ * & * & * & \mathcal{N}_{F,\hat{W}_{[i]}} & \mathbf{0} \\ * & * & * & * & \underline{\mathcal{N}}_{i,55} \end{bmatrix} - \underline{\mathcal{K}}_i^\top \underline{\mathcal{K}}_i \succ 0, \tag{7.23}$$

where $\underline{\mathcal{N}}_{i,33} := 2\underline{b}_i + \mathcal{N}_{B^\top \underline{P}_i^\top, \mathbf{I}}$, $\underline{\mathcal{N}}_{i,55} := \mathcal{N}_{\underline{P}, \hat{\underline{D}}_{[i]}} + \mathcal{N}_{K,\mathbf{I}}$, and $\underline{\mathcal{K}}_i := [\mathbf{0}\ \ \mathbf{0}\ \ -B^\top \underline{P}_i^\top\ \ \mathbf{0}\ \ K]$ (with the function $\mathcal{N}_{\cdot,\cdot}$ defined in Proposition 7.1). Taking Schur complement of (7.23),

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & -B^\top \underline{P}_i^\top & \mathbf{0} & K \\ * & \hat{\underline{D}}_{[i]} & \mathbf{0} & \underline{b} & \mathbf{0} & \mathbf{0} \\ * & * & \hat{W}_{[i]} & d & \mathbf{0} & \mathbf{0} \\ * & * & * & 2\underline{b}_i + \mathcal{N}_{B^\top \underline{P}_i^\top, \mathbf{I}} & \underline{P}_i & \underline{P}_i A \\ * & * & * & * & \mathcal{N}_{F,\hat{W}_{[i]}} & \mathbf{0} \\ * & * & * & * & * & \mathcal{N}_{\underline{P}, \hat{\underline{D}}_{[i]}} + \mathcal{N}_{K,\mathbf{I}} \end{bmatrix} \succ 0 \tag{7.24}$$

results, with all nonlinear components collected in the diagonal blocks. Using Proposition 7.1 on these components, we conclude that (7.16) is a sufficient LMI condition for (7.24). □

Corollary 7.1. *Suppose that Z_{NL} is feasible for Problem* (7.15)*. Then, the solution of the SDP*

$$\begin{aligned} &\min_{\boldsymbol{Z}}\ \alpha \left\|\underline{\boldsymbol{b}}\right\|_1 + \beta \left\|\bar{\boldsymbol{\epsilon}}\right\|_1 + \gamma \tilde{\boldsymbol{r}} \\ &\textit{s.t. } (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{d}) \in \hat{\Sigma}_T,\ \ (\bar{\boldsymbol{b}}, \bar{\boldsymbol{\epsilon}}) \in \bar{\mathcal{S}},\ \ (7.16) - (7.21), \end{aligned} \tag{7.25}$$

Algorithm 6: Update solution of Problem (7.15)

1) Obtain an initial feasible solution Z_{NL} for (7.15);
2) Solve SDP (7.25) for $\boldsymbol{Z}$, recover Z_{NL} from the solution;
2) Evaluate the objective value $\alpha \|\underline{b}\|_1 + \beta \|\bar{\epsilon}\|_1 + \gamma\tilde{r}$;
4) If there is a reduction from previous iteration, repeat Step 2 using Z_{NL} for linearization. Else, terminate.

$$\boldsymbol{Z} := \begin{pmatrix} \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{d}, \boldsymbol{\mathcal{Z}}, \boldsymbol{\lambda}, \boldsymbol{K}, \underline{\boldsymbol{b}}, \bar{\boldsymbol{b}}, \tilde{\boldsymbol{\Theta}}, \tilde{\boldsymbol{r}}, \hat{\hat{\boldsymbol{M}}}, \bar{\boldsymbol{\epsilon}}, \{\hat{\underline{\boldsymbol{D}}}_{[i]}, \hat{\boldsymbol{W}}_{[i]}, i \in \mathbb{I}_1^{\underline{m}}\}, \\ \{\hat{\bar{\boldsymbol{D}}}_{[i]}, i \in \mathbb{I}_1^{\bar{m}}\}, \{\hat{\underline{\boldsymbol{S}}}_{[i]}, \hat{\bar{\boldsymbol{S}}}_{[i]}, i \in \mathbb{I}_1^{m_x}\}, \{\hat{\underline{\boldsymbol{R}}}_{[i]}, \hat{\bar{\boldsymbol{R}}}_{[i]}, i \in \mathbb{I}_1^{m_u}\} \end{pmatrix},$$

is feasible for Problem (7.15)*, and satisfies the cost decrease condition*

$$\alpha \|\underline{\boldsymbol{b}}\|_1 + \beta \|\bar{\boldsymbol{\epsilon}}\|_1 + \gamma\tilde{\boldsymbol{r}} \leq \alpha \|\underline{b}\|_1 + \beta \|\bar{\epsilon}\|_1 + \gamma\tilde{r}.$$

□

Proof. The feasibility of $\boldsymbol{Z}$ for Problem (7.15) follows from Theorem 7.3, and the cost decrease condition holds since Z_{NL} is feasible for Problem (7.25). □

We propose the procedure in Algorithm 6 to solve Problem (7.15). The convergence of this algorithm can be studied using the results in [143, Chapter 4], and is left for future research.

Initialization procedure

We propose the following procedure to compute an initial feasible solution Z_{NL}.

(i) Select some $\hat{\theta}_T > 0$ through a guess to characterize $\hat{\Sigma}_T$.

(ii) Solve the LP
$$\min_{A,B,d} \{\|d\|_1 \text{ s.t. } (A, B, d) \in \hat{\Sigma}_T\}$$
for an initial feasible model (A, B, d).

(iii) Use the method in [76] to compute an initial RPI set $\Delta\mathcal{X} = \mathcal{P}(\underline{P}, \underline{b})$ satisfying (7.7a) along with a feedback gain K, while enforcing $\mathcal{P}(\underline{P}, \underline{b}) \subset \mathcal{X}$ and $K\mathcal{P}(\underline{P}, \underline{b}) \subset \mathcal{U}$.

(*iv*) Compute the tightened constraint set

$$\mathcal{O}_0 := \{x : x \in \mathcal{X} \ominus \Delta\mathcal{X}, Kx \in \mathcal{U} \ominus K\Delta\mathcal{X}\},$$

and then compute a PI set $\mathcal{X}^{\mathrm{t}} = \mathcal{P}(\bar{P}, \bar{b})$ using the method in [76] for $x(t+1) = (A + BK)x(t)$.

(*v*) Compute the remaining variables formulating Problem (7.15) by solving

$$\min_{Z_{\mathrm{I}}} \{\tilde{r} \text{ s.t. } (7.8) - -(7.11), (7.13) - -(7.14)\},$$

with the optimization variables

$$Z_{\mathrm{I}} := \begin{pmatrix} \{\underline{D}_{[i]}, W_{[i]}, i \in \mathbb{I}_1^m\}, \{\bar{D}_{[i]}, i \in \mathbb{I}_1^{\bar{m}}\}, \{\underline{S}_{[i]}, \bar{S}_{[i]}, i \in \mathbb{I}_1^{m_x}\}, \\ \{\underline{R}_{[i]}, \bar{R}_{[i]}, i \in \mathbb{I}_1^{m_u}\}, \tilde{\Theta}, \tilde{r}, \tilde{M} \end{pmatrix}.$$

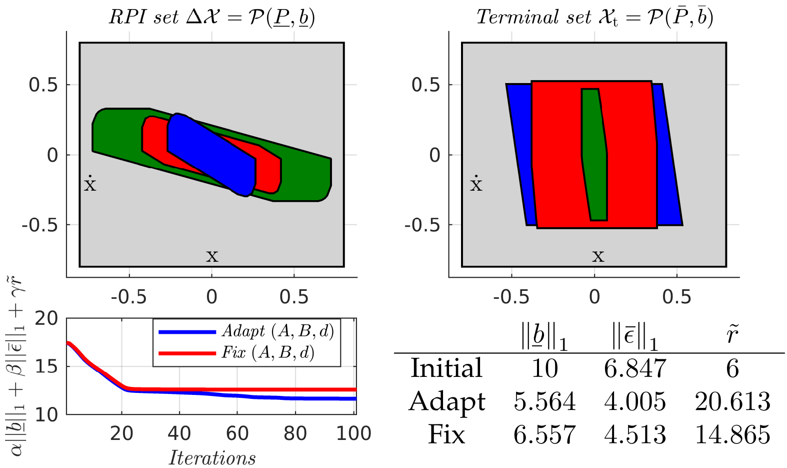
Remark 7.4. *In Steps (ii),(iii), the methods in [76] guarantee the feasibility of the SDP in Step (v), since they are also formulated using Theorem 7.2. Other methods, e.g. [144, 44], can also be used if the feasibility of Step (v) is ensured.* □

7.4 Numerical example

We consider a nonlinear mass-spring-damper system with dynamics

$$\mathrm{F} = \mathrm{m}\ddot{\mathrm{x}} + (\mathrm{K}\mathrm{x} + \mathrm{K}_{\mathrm{NL}}\mathrm{x}^2) + (\mathrm{c}\dot{\mathrm{x}} + \mathrm{c}_{\mathrm{NL}}\dot{\mathrm{x}}^2) + \mathrm{F}_\delta,$$

where $u = \mathrm{F}$, $x = [\mathrm{x}\ \dot{\mathrm{x}}]^\top$, and constraints $\mathcal{X} = \{x : ||x||_\infty \leq 0.8\}$, $\mathcal{U} = \{u : ||u||_\infty \leq 2.5\}$. We simulate the plant using $\mathrm{ode45}$ integration to build the dataset $\mathcal{D}$ with $T = 1000$ at a $0.1s$ time interval. We set $\mathrm{K}_{\mathrm{NL}}, \mathrm{c}_{\mathrm{NL}} = 0.12$, and uniformly sample the parameters $\mathrm{m}, \mathrm{K}, \mathrm{c}$ in $(0.44, 0.56)$ and F_δ in $(-0.12, 0.12)$ at every timestep $0.1s$. We then use Algorithm 6 to synthesize a model and RPI sets required for RMPC synthesis. To this end, we follow the initialization procedure described in Section 7.3-4(b) to obtain an initial feasible Z_{NL} for Problem (7.15). We first parametrize the disturbance set $\mathcal{W}$ with $m_W = 10$ hyperplanes.



	$\|\underline{b}\|_1$	$\|\bar{\epsilon}\|_1$	$\tilde{r}$
Initial	10	6.847	6
Adapt	5.564	4.005	20.613
Fix	6.557	4.513	14.865

Figure 20: Results of Algorithm 1. Gray sets-$\mathcal{X}$, Green sets-Initialization, Blue sets-(A, B, d) as optimization variables, Red sets - Fix (A, B, d) to initial values.

Then, following Step (i), we characterize the set $\hat{\Sigma}_T$ with $\hat{\theta}_T = 1 \cdot 10^{-3}$. Then, we compute the initial model

$$A = \begin{bmatrix} 0.9967 & 0.0951 \\ -0.0637 & 0.9036 \end{bmatrix}, \; B = \begin{bmatrix} 0.0098 \\ 0.1914 \end{bmatrix}$$

and $\|d\|_1 = 0.5816$ following Step (ii). We initialize the feedback gain as $K = [-0.4140 \; -2.3734]$ which is the optimal LQR gain corresponding to matrices $\tilde{Q} = \text{diag}(1, 15)$ and $\tilde{R} = 1$. Then, we compute an RPI set $\Delta\mathcal{X} = (\underline{P}, \underline{b})$ following Step (iii) with $\underline{m} = 10$ hyperplanes using [144]. Similarly, we compute the PI terminal set $\mathcal{X}^{\text{t}} = \mathcal{P}(\bar{P}, \bar{b})$ following Step (iv) with $\bar{m} = 15$ hyperplanes using [44]. Finally, with Step (v) we compute the remaining variables formulating Z_{NL}. We parameterize $\mathcal{B}(\bar{\epsilon})$ with $m_\epsilon = 10$ for terminal set maximization. The results obtained with Algorithm 6 with weights $\alpha = 1, \beta = 1, \gamma = 0.1$ using the MOSEK SDP solver [93] in MATLAB are shown in Figure 20. For the purpose of comparison, we also plot the results when the model (A, B, d) is fixed to the initial value. We observe that by allowing Algorithm 6 to adapt the system model using $\hat{\Sigma}_T$, we obtain a lower objective value with a larger terminal set $\mathcal{X}^{\mathrm{t}}$ and a smaller RPI set $\Delta\mathcal{X}$. The model at termination is

$$A = \begin{bmatrix} 0.9967 & 0.0951 \\ -0.0625 & 0.8990 \end{bmatrix},\ B = \begin{bmatrix} 0.0098 \\ 0.1958 \end{bmatrix}$$

and $\|d\|_1 = 0.5833$, and the computed feedback gain is $K = [-1.7062\ -2.6306]$: the model consists of a larger disturbance set than the initialized value, with (A, B, d) optimizing (7.15) instead of best fitting the data. In case the model is fixed to the initial value, the feedback gain at termination is $K = [-1.0033\ -3.0882]$. In order to study the effect of the parameter $\hat{\theta}_T$ characterizing $\hat{\Sigma}_T$, we run Algorithm 1 for increasing values of $\hat{\theta}_T$. The objective values at termination are 11.631 for $\hat{\theta}_T = 1 \cdot 10^{-3}$, 11.659 for $\hat{\theta}_T = 1.2 \cdot 10^{-3}$, 11.668 for $\hat{\theta}_T = 1.3 \cdot 10^{-3}$, 13.4376 for $\hat{\theta}_T = 1.5 \cdot 10^{-3}$: we observe that conservativeness increases with $\hat{\theta}_T$, while increasing robustness with respect to the underlying plant. Note that this trend is not guaranteed since Problem (7.15) is an NLPMI.

Computational Complexity: The SDP in (7.25) consists of an LMI constraint with $\underline{m}(2n_x + m_W + n_u + \underline{m} + 1) + \bar{m}(n_x + n_u + \bar{m} + 1) + (m_x + m_u)(2n_x + \underline{m} + \bar{m} + 1) + (3n_x + 2n_u) + (n_x + \bar{m} + 1) + n_x = 1086$ rows, $n_x m_x^v = 8$ linear equality constraints, and $2m_W T + 2m_W(2n_x + n_u) + (2n_x + n_u) + m_W + 2m_x^v(\bar{m} + m_\epsilon) = 20275$ linear inequality constraints over $2(n_x^2 + n_x n_u + n_x m_x^v + 1) + m_W(2n_x + n_u + 1) + \underline{m}(\underline{m} + m_W + 1 + m_x + m_u) + \bar{m}(\bar{m} + 2 + m_x + m_u) + m_\epsilon = 638$ variables. Over multiple runs, the average solving time for the SDP in (7.25) on a laptop with an Intel i7-7500U processor and 16GB of RAM running Ubuntu 16.04 is approximately 1.57s when the model is allowed to adapt, and 0.78s when the model is fixed. We note that the number of LMI constraints and variables scale quadratically in $\bar{m}$ and $\underline{m}$. Hence, the approach can be computationally expensive if a large number of hyperplanes are required for RPI set representation. Comparing our approach to [24] using data from a real-world system is a subject of future work.

Chapter 8

Conclusions

The main objective of this thesis was methods to synthesize disturbance and input constraint sets for linear time-invariant (LTI) systems subject to constraints. To this end, optimization problems were formulated to compute desirable large/small distrubance and input constraint sets, as dictated by the problem specifications. A central tool in the development of these methods was the Robust Positive Invariant (RPI) set, and computation of RPI sets was embedded in the aforementioned optimization problems. Hence, the problems were designed to co-synthesize disturbance/input constraint sets along with the corresponding RPI sets. In order to facilitate the *formulation* of such optimization problems, new results regarding RPI sets were presented. In order to *solve* the resulting optimization problems, structure-exploiting solvers were developed that were demonstrated numerically to outperform conventional methods.

In Chapter 3, the problem of computing a safe disturbance set for an LTI system subject to output constraints was considered. Essentially, a safe disturbance set ensures that the corresponding reachable set of outputs is included in the output constraint set. Then, an optimization problem was formulated to compute a safe disturbance set that minimizes the distance between the reachable set of outputs and the constraint set. The reachable set of outputs is characterized by the minimal Robust Positive Invariant (mRPI) set, such that the optimization problem constrains the mRPI set inside the constraint set. Unfortunately, computing an explicit representation of the mRPI is generally impossible, implying that the formulated optimization problem cannot be solved exactly. In order to tackle this issue, an RPI set parameterized as a polytope with fixed normal vectors was used to approximate the mRPI set. New results were developed proving that if an RPI set with the user-specified normal vectors exists, then the smallest such RPI set is unique, and satisfies a functional equality formulated using support functions over polytopes. This equality was used to formulate an optimization problem to compute a safe disturbance set parameterized as a polytope. In order to solve the resulting optimization problem, a smoothening-based interior-point solver was developed. The development of a specialized solver was necessary since the optimization problem was formulated using support functions that are inherently nonsmooth. The smoothening approach was developed by adopting results from parameteric optimization theory, and treating support functions as implicit functions of their parameters. Feasibility of the smoothened optimization problem was proven, and numerical efficacy as compared to alternative formulations was demonstrated.

In Chapter 4, the methods developed in Chapter 3 were used to synthesize a decentralized Model Predictive Control (DeMPC) scheme for a linear system composed of dynamically coupled subsystems and subject to coupled constraints. This scheme was developed by considering each dynamic coupling as an additive disturbance, such that the disturbance set is defined by the state constraint sets of the neighbors. In order to satisfy the coupled constraint, the state constraint set of each subsystem was designed by explicitly minimizing the conservativeness using the approach of Chapter 3, and locally regulating tube-based Robust MPC (RMPC) controllers were designed for robust constraint satisfaction. A numerical example was presented demonstrating these aspects, and the closed-loop behavior was compared against a centralized MPC scheme.

In Chapter 5, the problem of computing safe disturbance sets was reconsidered. An alternative approach based on implicit RPI sets was presented to tackle the problem, in which the mRPI set within the optimization problem framework was approximated with an RPI set whose approximation error with respect to the mRPI set could be specified a priori. This tackles the first issue encountered in the approach in Chapter 3. New results were developed to facilitate embedding the computation of such an RPI set within the optimization problem. Then, a novel disturbance set parameterization based on convex hull of polytopes was presented. This parameterization allows for the computation of disturbance sets without a priori fixing the orientation of the normal vectors, thus reducing conservativeness. Moreover, it also permits the characterization of the set of safe disturbance sets as a polyhedron by exploiting new results concerning support functions over convex hull of polytopes. This tackles the second issue posed by the approach of Chapter 3. In order to solve the resulting optimization problem, off-the-shelf solvers can be used. New approximate solution methods based on linear programing were also presented. The proposed method was demonstrated to vastly outperform the approach of Chapter 3, both with respect to conservativeness in the solution, and computational complexity. Using the methods, a reduced-order MPC scheme was developed.

In Chapter 6, the problem of computing the smallest input constraint set that guarantees robust stabilizability of a given set of initial conditions of a constrained uncertain LTI system was presented. The size of the input constraint set was defined by a function that could represent, for example, economic considerations, such that the proposed techniques can be used to solve practical problems such as actuator selection/design. In order to tackle the problem, it is assumed that the system is equipped with a tube-based RMPC controller, formulated with unknown input constraint set and terminal set. Then, an optimization problem was formulated with these sets as the optimization variables, and the input constraint set size as the objective. The constraints were defined using robust stabilization properties of the RMPC scheme. In order to solve the optimization problem, an iterative procedure was developed. Finite-time termination and optimality of the iterative procedure was shown. Also, the effect of RMPC horizon length on the solution of the procedure was discussed. Numerical examples demonstrating the efficacy of the procedure were presented. The methods were also extended to consider the effect of variation of the input constraint set on the system dynamics. These methods were applied to perform actuator selection in a practical problem.

In Chapter 7, a data-driven framework for RPI set synthesis was presented. In particular, it was assumed that an input-state dataset was collected from a plant. Using this dataset, a method was presented to select an uncertain LTI model of the plant that is characterized by the system matrices along with the disturbance set, and co-synthesize RPI sets along with corresponding invariance-inducing static feedback laws. Theoretical guarantees were provided to ensure that the identified model set accounts for unseen data, i.e, any model in the identified model set can robustly represent all possible plant behaviors. An optimization problem based on NonLinear Matrix Inequalities (NLMI) was formulated, with the objective being a system identification criterion optimizing for tube-based RMPC synthesis. The resulting optimization problem was solved using Semidefinite Programing (SDP) formulated with LMI-based convex underestimators from a feasible initial point. Reduced conservativeness of the proposed method as opposed to a sequential system identification and controller design scheme was demonstrated using a numerical example.

8.1 Future Work

There exist several directions of future research. Broadly, they are (a) clarifying the issues associated with the current methods, (b) extending the methods to accommodate a broader class of systems, and (c) the development of efficient numerical optimization algorithms.

Chapter 3: The first research question is the characterization of set of matrices E of pre-specified complexity satisfying Assumption 3.2. These matrices parameterize RPI sets of the uncertain LTI system, and they can currently be characterized using bilinear conditions through LP duality. However, characterizing them using linear conditions can simplify the identification of such matrices, thus expanding their practical applicability. The second research question is the a priori specification of approximation error of an RPI set parameterized with matrix E with respect to the mRPI set. In order to facilitate such a specification, an additional constraint enforcing the inclusion of the RPI set inside a μ-RPI set can be formulated, where μ is the approximation error. This can be done by adopting the results in Chapter 5, and inclusion encoding can be performed using existing results from literature. However, a more attractive direction is the characterization of matrices E that guarantee an upper-bound on the approximation error. With respect to the smoothening-based optimization algorithm, efficient Hessain computation/approximation methods can be derived, since the current method is very demanding with respect to memory requirements. Finally, extension of the method to accommodate the computation of a static feedback law, and development of results regarding polytopic sets can also be considered.

Chapter 4: Extension of the method to co-synthesize local static linear feedback laws satisfying Assumption 4.2 can be considered. Moreover, the extensions of the scheme to exploit partial communication between the controllers is a viable research direction, and explicit enforcement of RMPC feasibility from a given set of initial states can improve the practical applicability of the scheme.

Chapter 5: Future research can be directed towards the computation of safe disturbance sets for polytopic systems. This requires the extension of results in the chapter to ensure that the μ-RPI condition is satisfied for all systems belonging to a given polytopic set, as well as development of efficient inclusion encoding techniques inside the output constraint set. We believe that the methods in this chapter present a much improved approach (both with respect to conservativeness and computational complexity) as compared to those in Chapter 3 for safe disturbance set computation, and hence should be adopted in practice if additional requirements such as those in Assumption 5.4 are not present. Thus, the development of inclusion encoding techniques to satisfy requirements such as Assumption 5.4 by construction also presents an attractive research direction. Such techniques can also be used to accommodate the inclusion corresponding to the distance condition, thus eliminating nonlinearities in the eventual optimization problem formulation. Alternatively, the configuration-constrained polytopic parameterization presented recently in [149] provides a very attractive opportunity, owing to the fact that they permit encoding both inner and outer inclusions using linear inequalities. Such properties also allow for the design of DeMPC constraint sets presented in Chapter 4 using implicit RPI sets through linear programs, thus reducing conservativeness further, while also being considerably cheaper in terms of computational requirements.

Chapter 6: Results permitting the co-synthesis of a linear static feedback law, along with the input constraint set and terminal set can be developed. Moreover, the formulation can be enhanced to explicitly account for modification in the dynamics while performing actuator selection. At the moment, we lump this modification into the uncertainty set. However, by exploiting the structure of the modification, solutions with reduced conservativeness can be computed.

Chapter 7: Efficient methods to estimate the robustness parameter $\hat{\theta}_T$ satisfying Assumption 7.2 can be developed. Moreover, modifications to Theorem 7.1 in order to reduce the conservativeness in the model set can be considered. While the proposed methods can be extended using existing results from literature to accommodate the identification of LTI models with parametric uncertainty, development of a result similar to Theorem 7.1 for robustness guarantees in the presence of parametric uncertainty is a future research subject.

Appendix A

Illustrative example details

The matrices used in Example (A) are the following.

$$
\begin{aligned}
A &= \begin{bmatrix} 1.3 & 1 \\ 0 & 1 \end{bmatrix}, \\
B &= \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, \\
C &= \begin{bmatrix} 0.8871 & 1.2831 \\ 0.1371 & -2.5575 \end{bmatrix}, \\
D &= \begin{bmatrix} -1.8067 \\ -0.5819 \end{bmatrix}, \\
B_w &= \begin{bmatrix} 0.5316 & 1.1882 \\ 1.0656 & 0.3695 \end{bmatrix}, \\
D_w &= \begin{bmatrix} 0.4630 & 1.3621 \\ 1.0368 & -1.8349 \end{bmatrix}, \\
F &= \begin{bmatrix} 38.4106 & -8.5980 & -9.3035 & -2.0243 & 2.7624 & 8.2472 \\ 27.9082 & 14.8918 & -6.7597 & -19.2643 & -26.2888 & -25.3847 \end{bmatrix}^\top, \\
\mathcal{W} &= \{w \;:\; Fw \leq \mathbf{1}\}, \\
Q &= \mathrm{diag}(1, 10), \quad R = 1, \quad K = \begin{bmatrix} -0.6543 & -1.2067 \end{bmatrix}.
\end{aligned}
$$

Appendix B

Implementation of Constraints (5.65)

Defining $\mathbf{P}_{[l]} := \begin{bmatrix} CA^{l-1}B & CA^{l-2}B & \dots & CAB & CB & D \end{bmatrix}$, we implement constraints $\mathbf{C_w w} + \mathbf{C_z z} = \mathbf{h}$ and $\mathbf{D_x x} + \mathbf{D_{\bar{w}} \bar{w}} \leq \mathbf{0}$ as

$$\overbrace{\begin{bmatrix} \mathbf{P}_{[l]} & & \\ & \ddots & \\ & & \mathbf{P}_{[l]} \end{bmatrix}}^{\mathbf{C_w}} \overbrace{\begin{bmatrix} \mathrm{w}^1_{[10]} \\ \vdots \\ \mathrm{w}^1_{[1l-1]} \\ \mathrm{w}^2_{[1]} \\ \vdots \\ \mathrm{w}^1_{[v_{\mathcal{Y}}0]} \\ \vdots \\ \mathrm{w}^1_{[v_{\mathcal{Y}}l-1]} \\ \mathrm{w}^2_{[v_{\mathcal{Y}}]} \end{bmatrix}}^{\mathbf{w}} + \overbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} & & \\ \vdots & & \ddots & \\ \mathbf{0} & & & \mathbf{I} \end{bmatrix}}^{\mathbf{C_z}} \mathbf{z} = \overbrace{\begin{bmatrix} \mathrm{y}_{[1]} \\ \vdots \\ \mathrm{y}_{[v_{\mathcal{Y}}]} \end{bmatrix}}^{\mathbf{h}},$$

$$\overbrace{\begin{bmatrix} \mathbf{I}_N \otimes \begin{bmatrix} -\mathbf{I} \\ -\mathbf{I} \end{bmatrix} & \mathbf{I}_N \otimes \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \end{bmatrix} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{I}_N \otimes \begin{bmatrix} -\mathbf{I} \\ -\mathbf{I} \end{bmatrix} & \mathbf{I}_N \otimes \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \end{bmatrix} & \mathbf{0} & \mathbf{0} \end{bmatrix}}^{\mathbf{D_x}} \mathbf{x} + \overbrace{\begin{bmatrix} \mathbf{I}_{N(l+1)} \otimes \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} & & \\ & \ddots & \\ & & \mathbf{I}_{N(l+1)} \otimes \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \end{bmatrix}}^{\mathbf{D_{\bar{w}}}} \bar{\mathbf{w}} \leq \mathbf{0},$$

where $\bar{\mathbf{w}} := \begin{bmatrix} \bar{\mathfrak{w}}_{[1]}^\top & \dots & \bar{\mathfrak{w}}_{[v_\mathcal{Y}]}^\top \end{bmatrix}^\top$, with each element defined as

$$\bar{\mathfrak{w}}_{[i]} := \begin{bmatrix} \bar{\mathrm{w}}^{1\top}_{[i01]} \dots \bar{\mathrm{w}}^{1\top}_{[i0N]} \dots \bar{\mathrm{w}}^{1\top}_{[i\ l-1\ 1]} \dots \bar{\mathrm{w}}^{1\top}_{[i\ l-1\ N]} & \bar{\mathrm{w}}^{2\top}_{[i1]} \dots \bar{\mathrm{w}}^{2\top}_{[iN]} \end{bmatrix}^\top.$$

We implement constraint $\mathbf{T}_{\boldsymbol{\beta}}\boldsymbol{\beta} = \mathbf{1}$ with $\mathbf{T}_{\boldsymbol{\beta}} = \mathbf{I}_{v_\mathcal{Y}(l+1)} \otimes \mathbf{1}_{1\times N}$, where $\boldsymbol{\beta} := \begin{bmatrix} \bar{\boldsymbol{\beta}}_{[1]}^\top & \dots & \bar{\boldsymbol{\beta}}_{[v_\mathcal{Y}]}^\top \end{bmatrix}^\top$ with each element defined as

$$\bar{\boldsymbol{\beta}}_{[i]} := \begin{bmatrix} \beta^{1\top}_{[i01]} \dots \beta^{1\top}_{[i0N]} \dots \beta^{1\top}_{[i\ l-1\ 1]} \dots \beta^{1\top}_{[i\ l-1\ N]} & \beta^{2\top}_{[i1]} \dots \beta^{2\top}_{[iN]} \end{bmatrix}^\top.$$

Denoting $\hat{\mathbf{I}}_N := \begin{bmatrix} \mathbf{I}_{m_\mathcal{Y}} & \cdots \mathbf{I}_{m_\mathcal{Y}} \end{bmatrix}^\top \in \mathbb{R}^{m_\mathcal{Y} N \times m_\mathcal{Y}}$, we implement constraint $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ as

$$\overbrace{\left[\begin{array}{ccccccc} \mathbf{I}_N \otimes |\bar{G}_{[0]}B| & \mathbf{I}_N \otimes \bar{G}_{[0]}B & -\hat{\mathbf{I}}_N & & & & \\ \mathbf{I}_N \otimes |\bar{G}_{[1]}B| & \mathbf{I}_N \otimes \bar{G}_{[1]}B & & -\hat{\mathbf{I}}_N & & & \\ \vdots & \vdots & & & \ddots & & \\ \mathbf{I}_N \otimes |\bar{G}_{[s-1]}B| & \mathbf{I}_N \otimes \bar{G}_{[s-1]}B & & & & -\hat{\mathbf{I}}_N & \\ \mathbf{I}_N \otimes |GD| & \mathbf{I}_N \otimes GD & & & & & -\hat{\mathbf{I}}_N \\ & & \mathbf{I}_{m_\mathcal{Y}} & \mathbf{I}_{m_\mathcal{Y}} & \dots & \mathbf{I}_{m_\mathcal{Y}} & \mathbf{I}_{m_\mathcal{Y}} \\ \hline \mathbf{I}_N \otimes |\tilde{\mathbf{I}}_{n_x}B| & \mathbf{I}_N \otimes \tilde{\mathbf{I}}_{n_x}B & & & & & \\ \hline \begin{bmatrix} -\mathbf{I}_{n_w} & \mathbf{0} \end{bmatrix} & \begin{bmatrix} -\mathbf{I}_{n_w} & \mathbf{0} \end{bmatrix} & & & & & \\ \begin{bmatrix} -\mathbf{I}_{n_w} & \mathbf{0} \end{bmatrix} & \begin{bmatrix} \mathbf{I}_{n_w} & \mathbf{0} \end{bmatrix} & & & & & \end{array}\right]}^{\mathbf{A}} \overbrace{\begin{bmatrix} \epsilon^w_{[1]} \\ \vdots \\ \epsilon^w_{[N]} \\ \bar{w}_{[1]} \\ \vdots \\ \bar{w}_{[N]} \\ \mathbf{Q}_{[0]} \\ \vdots \\ \mathbf{Q}_{[s-1]} \\ \mathbf{r} \end{bmatrix}}^{\mathbf{x}} \leq \overbrace{\left[\begin{array}{c} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \\ g - \lambda \sum_{t=0}^{s-1} |\bar{G}_{[t]}|\mathbf{1} \\ \hline \gamma\mathbf{1} \\ \hline \mathbf{0} \\ \mathbf{0} \end{array}\right]}^{\mathbf{b}}.$$

Finally, we implement constraint $\mathbf{E}_\mathbf{z}\mathbf{z} \leq \mathbf{0}$ with $\mathbf{E}_\mathbf{z} = \begin{bmatrix} -\mathbf{I} & H & & \\ \vdots & & \ddots & \\ -\mathbf{I} & & & H \end{bmatrix}$.

Bibliography

[1] A. Alessio et al. "Squaring the circle: An algorithm for generating polyhedral invariant sets from ellipsoidal ones". In: *Automatica* 43.12 (2007), pp. 2096–2103. ISSN: 0005-1098.

[2] Aaron D. Ames et al. "Control Barrier Function Based Quadratic Programs for Safety Critical Systems". In: *IEEE Transactions on Automatic Control* 62.8 (2017), pp. 3861–3876.

[3] Aaron D. Ames et al. "Control Barrier Functions: Theory and Applications". In: *2019 18th European Control Conference (ECC)*. 2019, pp. 3420–3431.

[4] G. Anandalingam and D. J. White. "A solution method for the linear static Stackelberg problem using penalty functions". In: *IEEE Transactions on Automatic Control* 35.10 (1990), pp. 1170–1173.

[5] Tzanis Anevlavis et al. *Controlled invariant sets: implicit closed-form representations and applications*. 2021.

[6] Mato Baotic. "Polytopic Computations in Constrained Optimal Control". In: *Automatika* 50 (Apr. 2009), pp. 119–134.

[7] Jonathan F. Bard. *Practical Bilevel Optimization: Algorithms and Applications (Nonconvex Optimization and Its Applications)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0792354583.

[8] Alberto Bemporad and Davide Barcelli. "Decentralized Model Predictive Control". In: *Networked Control Systems*. Ed. by Alberto Bemporad, Maurice Heemels, and Mikael Johansson. London: Springer London, 2010, pp. 149–178. ISBN: 978-0-85729-033-5.

[9] Julian Berberich et al. "Data-Driven Model Predictive Control With Stability and Robustness Guarantees". In: *IEEE TAC* 66.4 (Apr. 2021), pp. 1702–1717. ISSN: 2334-3303.

[10] Julian Berberich et al. “Robust data-driven state-feedback design”. In: *Proc. of the 2020 ACC*. 2020, pp. 1532–1538.

[11] D. Bertsekas. “Infinite time reachability of state-space regions by using feedback control”. In: *IEEE Transactions on Automatic Control* 17.5 (Oct. 1972), pp. 604–613. ISSN: 2334-3303.

[12] Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Jan. 1998.

[13] Andrea Bisoffi, Claudio De Persis, and Pietro Tesi. “Controller design for robust invariance from noisy data”. In: *arXiv e-prints*, arXiv:2007.13181 (July 2020), arXiv:2007.13181.

[14] F. Blanchini. “Constrained control for uncertain linear systems”. In: *Journal of Optimization Theory and Applications* 71.3 (Dec. 1991), pp. 465–484.

[15] F. Blanchini. “Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions”. In: *IEEE Transactions on Automatic Control* 39.2 (1994), pp. 428–433.

[16] Franco Blanchini and Stefano Miani. *Set-Theoretic Methods in Control*. Jan. 2007. ISBN: 9780817632557.

[17] Toni Barjas Blanco, Mark Cannon, and Bart De Moor. “On efficient computation of low-complexity controlled invariant sets for uncertain linear systems”. In: *International Journal of Control* 83.7 (2010), pp. 1339–1346.

[18] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[19] S. Boyd et al. *Linear Matrix Inequalities in System and Control Theory*. SIAM studies in applied mathematics: 15, 1994.

[20] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[21] Stephen Boyd et al. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers”. In: *Found. Trends Mach. Learn.* 3.1 (Jan. 2011), pp. 1–122. ISSN: 1935-8237.

[22] M.S. Branicky. “Multiple Lyapunov functions and other analysis tools for switched and hybrid systems”. In: *IEEE Transactions on Automatic Control* 43.4 (1998), pp. 475–482.

[23] Christof Büskens and Helmut Maurer. "Sensitivity Analysis and Real-Time Optimization of Parametric Nonlinear Programming Problems". In: *Online Optimization of Large Scale Systems*. Ed. by Martin Grötschel, Sven O. Krumke, and Jörg Rambau. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 3–16. ISBN: 978-3-662-04331-8.

[24] Yuxiao Chen and Necmiye Ozay. "Data-Driven Computation of Robust Control Invariant Sets With Concurrent Model Selection". In: *IEEE Transactions on Control Systems Technology* (2021), pp. 1–12.

[25] Graziano Chesi. "Rational Lyapunov functions for estimating and controlling the robust domain of attraction". In: *Automatica* 49.4 (2013), pp. 1051–1057. ISSN: 0005-1098.

[26] L. Chisci, J.A. Rossiter, and G. Zappa. "Systems with persistent disturbances: predictive control with restricted constraints". In: *Automatica* 37.7 (2001), pp. 1019–1028. ISSN: 0005-1098.

[27] Glen Chou et al. "Using Control Synthesis to Generate Corner Cases: A Case Study on Autonomous Driving". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37.11 (2018), pp. 2906–2917.

[28] Christian Conte et al. "Distributed synthesis and control of constrained linear systems". In: *2012 American Control Conference (ACC)*. 2012, pp. 6017–6022.

[29] Andres Cotorruelo et al. "Reference dependent invariant sets: Sum of squares based computation and applications in constrained control". In: *Automatica* 129 (2021), p. 109614. ISSN: 0005-1098.

[30] Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-Enabled Predictive Control: In the Shallows of the DeePC". In: *Proc. of the 2019 ECC*. 2019, pp. 307–312.

[31] IBM ILOG Cplex. "V12. 1: User's Manual for CPLEX". In: *International Business Machines Corporation* 46.53 (2009), p. 157.

[32] M. S. Darup and M. Mönnigmann. "On general relations between null-controllable and controlled invariant sets for linear constrained systems". In: *53rd IEEE Conference on Decision and Control*. 2014, pp. 6323–6328.

[33] Claudio De Persis and Pietro Tesi. "Formulas for Data-Driven Control: Stabilization, Optimality, and Robustness". In: *IEEE TAC* 65.3 (2020), pp. 909–924.

[34] S. Di Cairano and A. Bemporad. "Model predictive controller matching: Can MPC enjoy small signal properties of my favorite linear controller?" In: *2009 European Control Conference (ECC)*. 2009, pp. 2217–2222.

[35] M. Farina and R. Scattolini. "Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems". In: *Automatica* 48.6 (2012), pp. 1088–1096. ISSN: 0005-1098.

[36] G. Ferrari-Trecate, P. Letizia, and M. Spedicato. "Optimization with Piecewise Affine Cost Functions". In: (2001).

[37] Mirko Fiacchini and Mazen Alamir. *Computing control invariant sets in high dimension is easy*. 2018.

[38] "Introduction to Sensitivity and Stability Analysis in Nonlinear Programming". In: ed. by Anthony V. Fiacco. Vol. 165. Mathematics in Science and Engineering. Elsevier, 1983, pp. 67–90.

[39] J.V. Flores, D. Eckhard, and J.M. Gomes da Silva. "On the Tracking Problem for Linear Systems subject to Control Saturation". In: *IFAC Proceedings Volumes* 41.2 (2008). 17th IFAC World Congress, pp. 14168–14173. ISSN: 1474-6670.

[40] Emanuele Garone, Stefano Di Cairano, and Ilya Kolmanovsky. "Reference and command governors for systems with constraints: A survey on theory and applications". In: *Automatica* 75 (2017), pp. 306–328. ISSN: 0005-1098.

[41] Emanuele Garone, Francesco Tedesco, and Alessandro Casavola. "Sensorless supervision of linear dynamical systems: The Feed-Forward Command Governor approach". In: *Automatica* 47.7 (2011), pp. 1294–1303. ISSN: 0005-1098.

[42] R. Ghaemi, J. Sun, and I. V. Kolmanovsky. "Robust Control of Constrained Linear Systems With Bounded Disturbances". In: *IEEE Transactions on Automatic Control* 57.10 (2012), pp. 2683–2688.

[43] Kasra Ghasemi, Sadra Sadraddini, and Calin Belta. "Compositional Synthesis of Decentralized Robust Set-Invariance Controllers for Large-scale Linear Systems". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. 2019, pp. 2054–2059.

[44] E. G. Gilbert and K. T. Tan. "Linear systems with state and control constraints: the theory and application of maximal output admissible sets". In: *IEEE Transactions on Automatic Control* 36.9 (Sept. 1991), pp. 1008–1020.

[45] Giorgio Giorgi. *Nonnegative Square Matrices: Irreducibility, Reducibility, Primitivity and Some Economic Applications*. DEM Working Papers Series 175. University of Pavia, Department of Economics and Management, Oct. 2019.

[46] P. Grieder et al. "Invariant sets for switched discrete time systems subject to bounded disturbances". In: *IFAC Proceedings Volumes* 38.1 (2005). 16th IFAC World Congress, pp. 115–120. ISSN: 1474-6670.

[47] Leonidas J. Guibas, An Nguyen, and Li Zhang. "Zonotopes as Bounding Volumes". In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '03. Baltimore, Maryland: Society for Industrial and Applied Mathematics, 2003, pp. 803–812. ISBN: 0898715385.

[48] Ankit Gupta and Paolo Falcone. "Full-Complexity Characterization of Control-Invariant Domains for Systems With Uncertain Parameter Dependence". In: *IEEE L-CSS* 3.1 (2019), pp. 19–24.

[49] Ankit Gupta, Hakan Köroğlu, and Paolo Falcone. "Computation of low-complexity control-invariant sets for systems with uncertain parameter dependence". In: *Automatica* 101 (2019), pp. 330–337. ISSN: 0005-1098.

[50] Ankit Gupta et al. *Computation of Parameter Dependent Robust Invariant Sets for LPV Models with Guaranteed Performance*. 2022.

[51] LLC Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2019.

[52] Didier Henrion and Milan Korda. "Convex Computation of the Region of Attraction of Polynomial Control Systems". In: *IEEE Transactions on Automatic Control* 59.2 (2014), pp. 297–312.

[53] M. Herceg et al. "Multi-Parametric Toolbox 3.0". In: *Proc. of the European Control Conference*. Zürich, Switzerland, July 2013, pp. 502–510.

[54] Leslie Hogben, ed. *Handbook of Linear Algebra*. Boca Raton, FL, USA: CRC Press, 2006.

[55] Achim Ilchmann, Oliver Sawodny, and Stephan Trenn. "Pneumatic cylinders: Modelling and feedback force-control". In: *International Journal of Control* 79 (Apr. 2005).

[56] Hartmut Janocha. *Actuators: Basics and Applications*. Jan. 2004. ISBN: 978-3-642-08266-5.

[57] Zhong-Ping Jiang and Yuan Wang. "Input-to-state stability for discrete-time nonlinear systems". In: *Automatica* 37.6 (2001), pp. 857–869. ISSN: 0005-1098.

[58] E. C. Kerrigan and J. M. Maciejowski. "Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control". In: *Proceedings of the 39th IEEE Conference on Decision and Control*. Vol. 5. 2000, 4951–4956 vol.5.

[59] Hassan K Khalil. *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002.

[60] H. Kiendl, J. Adamy, and P. Stelzner. "Vector norms as Lyapunov functions for linear systems". In: *IEEE Transactions on Automatic Control* 37.6 (1992), pp. 839–842.

[61] Youngdae Kim, Sven Leyffer, and Todd Munson. *MPEC methods for bilevel optimization problems*. Argonne National Laboratory, MCS Division Preprint, ANL/MCS-P9195-0719.

[62] M. Kögel and R. Findeisen. "Improved Robust Decentralized MPC". In: *2018 European Control Conference (ECC)*. 2018, pp. 312–318.

[63] Ilya Kolmanovsky and Elmer G. Gilbert. "Theory and computation of disturbance invariant sets for discrete-time linear systems". In: *Mathematical Problems in Engineering* 4.4 (1998), pp. 317–367. ISSN: 0005-1098.

[64] R.L. Kosut, M.K. Lau, and S.P. Boyd. "Set-membership identification of systems with parametric and nonparametric uncertainty". In: *IEEE TAC* 37.7 (1992), pp. 929–941.

[65] Mayuresh V. Kothare, Venkataramanan Balakrishnan, and Manfred Morari. "Robust constrained model predictive control using linear matrix inequalities". In: *Automatica* 32.10 (1996), pp. 1361–1379. ISSN: 0005-1098.

[66] K.I. Kouramas et al. "On the Minimal Robust Positively Invariant Set for Linear Difference Inclusions". In: *Proceedings of the 44th IEEE Conference on Decision and Control*. 2005, pp. 2296–2301.

[67] Basil Kouvaritakis and Mark Cannon. "Model Predictive Control: Classical, Robust and Stochastic". In: 2015.

[68] Mohammed-Tahar Laraba et al. "A bilevel optimization approach for D-invariant set design". In: *6th symposium on system structure and control (SSSC)*. Vol. 49. Proceedings of the IFAC joint Conference 10. Istanbul, Turkey, June 2016, pp. 235–240.

[69] Mircea Lazar and Andrej Jokić. "On infinity norms as Lyapunov functions for piecewise affine systems". In: ACM Press, 2010.

[70] D. Limon, T. Alamo, and E.F. Camacho. "Enlarging the domain of attraction of MPC controllers". In: *Automatica* 41.4 (2005), pp. 629–635. ISSN: 0005-1098.

[71] D. Limon et al. "Robust tube-based MPC for tracking of constrained linear systems with additive disturbances". In: *Journal of Process Control* 20.3 (2010), pp. 248–260. ISSN: 0959-1524.

[72] D. Limon Marruedo, T. Alamo, and E.F. Camacho. "Stability analysis of systems with bounded additive uncertainties based on invariant sets: Stability and feasibility of MPC". In: *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*. Vol. 1. 2002, 364–369 vol.1.

[73] Chengyuan Liu. "Robust model predictive control: robust control invariant sets and efficient implementation". PhD thesis. Imperial College London, 2017.

[74] Chengyuan Liu and Imad M. Jaimoukha. "The computation of full-complexity polytopic robust control invariant sets". In: *2015 54th IEEE CDC*. 2015, pp. 6233–6238.

[75] Chengyuan Liu, Furqan Tahir, and Imad M. Jaimoukha. "Full-complexity polytopic robust control invariant sets for uncertain linear discrete-time systems". In: *International Journal of Robust and Nonlinear Control* 29.11 (2019), pp. 3587–3605.

[76] Chengyuan Liu, Furqan Tahir, and Imad M. Jaimoukha. "Full-complexity polytopic robust control invariant sets for uncertain linear discrete-time systems". In: *International Journal of Robust and Nonlinear Control* 29.11 (2019), pp. 3587–3605.

[77] Xinwei Liu, Georgia Perakis, and Jie Sun. "A Robust SQP Method for Mathematical Programs with Linear Complementarity Constraints". In: *Computational Optimization and Applications* 34.1 (May 2006), pp. 5–33. ISSN: 1573-2894.

[78] Zexiang Liu and Necmiye Ozay. "On the Convergence of the Backward Reachable Sets of Robust Controlled Invariant Sets For Discrete-time Linear Systems". In: (2022).

[79] Lennart Ljung. *System Identification (2nd Ed.): Theory for the User*. USA: Prentice Hall PTR, 1999. ISBN: 0136566952.

[80] Lennart Ljung. *System Identification: Theory for the User*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1986. ISBN: 0-138-81640-9.

[81] Joseph Lorenzetti and Marco Pavone. "A Simple and Efficient Tube-based Robust Output Feedback Model Predictive Control Scheme". In: *2020 European Control Conference (ECC)*. 2020, pp. 1775–1782.

[82] Jos M. Maestre and Rudy R. Negenborn. *Distributed Model Predictive Control Made Easy*. Springer Publishing Company, Incorporated, 2013. ISBN: 9400770057.

[83] L. Magni et al. "A stabilizing model-based predictive control algorithm for nonlinear systems". In: *Automatica* 37.9 (2001), pp. 1351–1362. ISSN: 0005-1098.

[84] O. L. Mangasarian and R. R. Meyer. "Nonlinear Perturbation of Linear Programs". In: *SIAM Journal on Control and Optimization* 17.6 (1979), pp. 745–752.

[85] John J. Martínez. "Minimal RPI sets computation for polytopic systems using the Bounded-real lemma and a new shrinking procedure". In: *IFAC-PapersOnLine* 48.26 (2015). 1st IFAC Workshop on Linear Parameter Varying Systems LPVS 2015, pp. 182–187. ISSN: 2405-8963.

[86] Daniele Masti, Mario Zanon, and Alberto Bemporad. "Tuning LQR Controllers: A Sensitivity-Based Approach". In: *IEEE L-CSS* 6 (2022), pp. 932–937.

[87] D.Q. Mayne, M.M. Seron, and S.V. Raković. "Robust model predictive control of constrained linear systems with bounded disturbances". In: *Automatica* 41.2 (2005), pp. 219–224. ISSN: 0005-1098.

[88] D.Q. Mayne et al. "Robust output feedback model predictive control of constrained linear systems". In: *Automatica* 42.7 (2006), pp. 1217–1222. ISSN: 0005-1098.

[89] Nimrod Megiddo. "Pathways to the Optimal Set in Linear Programming". In: *Progress in Mathematical Programming: Interior-Point and Related Methods*. Ed. by Nimrod Megiddo. New York, NY: Springer New York, 1989, pp. 131–158. ISBN: 978-1-4613-9617-8.

[90] N. Meslem, N. Loukkas, and J.J. Martinez. "Using set invariance to design robust interval observers for discrete-time linear systems". In: *International Journal of Robust and Nonlinear Control* 28.11 (2018), pp. 3623–3639.

[91] S. Miani and C. Savorgnan. "MAXIS-G: a software package for computing polyhedral invariant sets for constrained LPV systems". In: *Proceedings of the 44th IEEE Conference on Decision and Control*. 2005, pp. 7609–7614.

[92] Basílio E.A. Milani. "Piecewise-affine Lyapunov functions for discrete-time linear systems with saturating controls". In: *Automatica* 38.12 (2002), pp. 2177–2184. ISSN: 0005-1098.

[93] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.* 2019.

[94] S. K. Mulagaleti, A. Bemporad, and M. Zanon. "Computation of Least-Conservative State-Constraint Sets for Decentralized MPC With Dynamic and Constraint Coupling". In: *IEEE Control Systems Letters* 5.1 (2021), pp. 235–240.

[95] S.K. Mulagaleti, Mario Zanon, and Alberto Bemporad. "Dynamic output disturbance models for robust model predictive control". In: *Proceedings of the 21st IFAC World Congress* (2020). to be published.

[96] Sampath Kumar Mulagaleti, Alberto Bemporad, and Mario Zanon. "A Smoothening-based Algorithm for the Computation of Input Disturbance Sets Guaranteeing Constrained Output Reachability". submitted. 2022.

[97] Sampath Kumar Mulagaleti, Alberto Bemporad, and Mario Zanon. "Computation of Input Disturbance Sets for Constrained Output Reachability". In: *IEEE Transactions on Automatic Control* (2022), pp. 1–8.

[98] Sampath Kumar Mulagaleti, Alberto Bemporad, and Mario Zanon. "Computation of Safe Disturbance sets using Implicit RPI sets". submitted. 2022.

[99] Sampath Kumar Mulagaleti, Alberto Bemporad, and Mario Zanon. "Data-Driven Synthesis of Robust Invariant Sets and Controllers". In: *IEEE Control Systems Letters* 6 (2022), pp. 1676–1681.

[100] Sampath Kumar Mulagaleti, Alberto Bemporad, and Mario Zanon. "Input Constraint Sets for Robust Regulation of Linear Systems". In: *IEEE Transactions on Automatic Control* (2021), pp. 1–1.

[101] Petter Nilsson and Necmiye Ozay. "Synthesis of separable controlled invariant sets for modular local control design". In: *2016 American Control Conference (ACC)*. 2016, pp. 5656–5663.

[102] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. second. New York, NY, USA: Springer, 2006.

[103] B.D. O'Dell and E.A. Misawa. "Semi-ellipsoidal controlled invariant sets for constrained linear systems". In: *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*. Vol. 3. 2000, 1779–1783 vol.3.

[104] S. Olaru, J.A. De Doná, and M.M. Seron. "Positive invariant sets for fault tolerant multisensor control schemes". In: *IFAC Proceedings Volumes* 41.2 (2008). 17th IFAC World Congress, pp. 1224–1229. ISSN: 1474-6670.

[105] F. Di Palma and L. Magni. "A multi-model structure for model predictive control". In: *Annual Reviews in Control* 28.1 (2004), pp. 47–52. ISSN: 1367-5788.

[106] B. Pluymers et al. "The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty". In: *Proceedings of the 2005, American Control Conference, 2005.* 2005, 804–809 vol. 2.

[107] M. J. D. Powell. "A fast algorithm for nonlinearly constrained optimization calculations". In: *Numerical Analysis*. Ed. by G. A. Watson. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 144–157. ISBN: 978-3-540-35972-2.

[108] S. Prajna, A. Papachristodoulou, and P.A. Parrilo. "Introducing SOSTOOLS: a general purpose sum of squares programming solver". In: *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.* Vol. 1. 2002, 741–746 vol.1.

[109] Arvind Raghunathan and Lorenz Biegler. “An Interior Point Method for Mathematical Programs with Complementarity Constraints (MPCCs)”. In: *SIAM Journal on Optimization* 15 (Mar. 2005), pp. 720–750.

[110] D.M. Raimondo, L. Magni, and R. Scattolini. “Decentralized MPC of nonlinear systems: An input-to-state stability approach”. In: *International Journal of Robust and Nonlinear Control* 17.17 (2007), pp. 1651–1667.

[111] S. V. Raković and K. I. Kouramas. “The Minimal Robust Positively Invariant Set for Linear Discrete Time Systems: Approximation Methods and Control Applications”. In: *Proceedings of the 45th IEEE Conference on Decision and Control*. Dec. 2006, pp. 4562–4567.

[112] S. V. Raković et al. “Invariant approximations of the minimal robust positively Invariant set”. In: *IEEE Transactions on Automatic Control* 50.3 (Mar. 2005), pp. 406–410. ISSN: 0018-9286.

[113] S.V. Raković, E. De Santis, and P. Caravani. “Invariant Equilibria of Polytopic Games via Optimized Robust Control Invariance”. In: *Proceedings of the 44th IEEE Conference on Decision and Control*. 2005, pp. 7686–7691.

[114] S.V. Raković et al. “Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances”. In: *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*. Vol. 2. 2004, 1418–1423 Vol.2.

[115] S.V. Raković et al. “Homothetic tube model predictive control”. In: *Automatica* 48.8 (2012), pp. 1631–1638. ISSN: 0005-1098.

[116] S.V. Raković et al. “Optimized robust control invariant sets for constrained linear discrete-time systems”. In: *IFAC Proceedings Volumes* 38.1 (2005). 16th IFAC World Congress, pp. 584–589. ISSN: 1474-6670.

[117] Saša V. Raković, Benjamin Kern, and Rolf Findeisen. “Practical set invariance for decentralized discrete time systems”. In: *49th IEEE Conference on Decision and Control (CDC)*. 2010, pp. 3283–3288.

[118] Saša V. Raković, Basil Kouvaritakis, and Mark Cannon. “Equinormalization and exact scaling dynamics in homothetic tube model predictive control”. In: *Systems & Control Letters* 62.2 (2013), pp. 209–217. ISSN: 0167-6911.

[119] J Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Jan. 2009.

[120] S. Riverso, M. Farina, and G. Ferrari-Trecate. "Plug-and-Play Decentralized Model Predictive Control for Linear Systems". In: *IEEE Transactions on Automatic Control* 58.10 (Oct. 2013), pp. 2608–2614. ISSN: 2334-3303.

[121] L. Rodrigues. "Stability Analysis Using Controlled Invariant Sets for Piecewise-Affine Systems". In: *2003 4th International Conference on Control and Automation Proceedings*. 2003, pp. 198–202.

[122] Damiano Rotondo and Gianluca Rizzello. "On the optimization of actuator saturation limits for LTI systems: an LMI-based invariant ellipsoid approach". In: *IFAC-PapersOnLine* 53.2 (2020). 21th IFAC World Congress, pp. 5567–5572. ISSN: 2405-8963.

[123] Daniel Rubin, Hoai-Nam Nguyen, and Per-Olof Gutman. "Computation of Polyhedral Positive Invariant Sets via Linear Matrix Inequalities". In: *2018 European Control Conference (ECC)*. 2018, pp. 2941–2946.

[124] Daniel Y. Rubin, Hoai-Nam Nguyen, and Per-Olof Gutman. "Yet Another Algorithm for the Computation of Polyhedral Positive Invariant Sets". In: *2018 IEEE Conference on Control Technology and Applications (CCTA)*. 2018, pp. 698–703.

[125] M. Rungger and P. Tabuada. "Computing Robust Controlled Invariant Sets of Linear Systems". In: *IEEE Transactions on Automatic Control* 62.7 (July 2017), pp. 3665–3670. ISSN: 2334-3303.

[126] M. Rungger and P. Tabuada. "Computing Robust Controlled Invariant Sets of Linear Systems". In: *IEEE Transactions on Automatic Control* 62.7 (2017), pp. 3665–3670.

[127] Sadra Sadraddini and Russ Tedrake. "Linear Encodings for Polytope Containment Problems". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. 2019, pp. 4367–4372.

[128] M. Saunders and J. Tomlin. "Solving regularized linear programs using barrier methods and KKT systems". In: 1996.

[129] Lukas Schafer, Felix Gruber, and Matthias Althoff. *Scalable Computation of Robust ControlInvariant Sets of Nonlinear Systems*.

[130] R. M. Schaich and M. Cannon. "Robust positively invariant sets for state dependent and scaled disturbances". In: *2015 54th IEEE Conference on Decision and Control (CDC)*. Dec. 2015, pp. 7560–7565.

[131] Rolf Schneider. *Convex Bodies: The Brunn–Minkowski Theory*. 2nd ed. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2013.

[132] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley Series in Discrete Mathematics & Optimization. Wiley, 1998. ISBN: 9780471982326.

[133] Moritz Schulze Darup, Rainer Schaich, and Mark Cannon. "How scaling of the disturbance set affects robust positively invariant sets for linear systems". In: *International Journal of Robust and Nonlinear Control* 27.16 (2017), pp. 3236–3258.

[134] Francesco Scibilia, Sorin Olaru, and Morten Hovd. "On feasible sets for MPC and their approximations". In: *Automatica* 47.1 (2011), pp. 133–139. ISSN: 0005-1098.

[135] Maria M. Seron et al. "On Finitely Determined Minimal Robust Positively Invariant Sets". In: *2019 Australian & New Zealand Control Conference (ANZCC)*. 2019, pp. 157–162.

[136] V. L. Silva, L. F. O. Chamon, and A. Ribeiro. "Model Predictive Selection: A Receding Horizon Scheme for Actuator Selection". In: *2019 American Control Conference (ACC)*. 2019, pp. 347–353.

[137] Pantelis Sopasakis, Daniele Bernardini, and Alberto Bemporad. "Constrained model predictive control based on reduced-order models". In: *52nd IEEE Conference on Decision and Control*. 2013, pp. 7071–7076.

[138] F. Tahir and I. M. Jaimoukha. "Low-Complexity Polytopic Invariant Sets for Linear Systems Subject to Norm-Bounded Uncertainty". In: *IEEE Transactions on Automatic Control* 60.5 (2015), pp. 1416–1421.

[139] Furqan Tahir. "Efficient computation of Robust Positively Invariant sets with linear state-feedback gain as a variable of optimization". In: *2010 7th International Conference on Electrical Engineering Computing Science and Automatic Control*. 2010, pp. 199–204.

[140] Furqan Tahir and Imad M. Jaimoukha. "Robust feedback model predictive control of constrained uncertain systems". In: *Journal of Process Control* 23.2 (2013). IFAC World Congress Special Issue, pp. 189–200. ISSN: 0959-1524.

[141] Junbo Tan et al. “A Novel RPI Set Computation Method for Discrete-time LPV Systems with Bounded Uncertainties”. In: *2019 IEEE 15th International Conference on Control and Automation (ICCA)*. 2019, pp. 946–951.

[142] Enrico Terzi et al. “Learning-based predictive control for linear systems: A unitary approach”. In: *Automatica* 108 (2019), p. 108473. ISSN: 0005-1098.

[143] Quoc Tran-Dinh. “Sequential Convex Programming and Decomposition Approaches for Nonlinear Optimization”. PhD thesis. Nov. 2012.

[144] P. Trodden. “A One-Step Approach to Computing a Polytopic Robust Positively Invariant Set”. In: *IEEE Transactions on Automatic Control* 61.12 (Dec. 2016), pp. 4100–4105. ISSN: 0018-9286.

[145] Paul A. Trodden and J.M. Maestre. “Distributed predictive control with minimization of mutual disturbances”. In: *Automatica* 77 (2017), pp. 31–43. ISSN: 0005-1098.

[146] V. Tzoumas et al. “Minimal actuator placement with optimal control constraints”. In: *2015 American Control Conference (ACC)*. 2015, pp. 2081–2086.

[147] Vasileios Tzoumas, Ali Jadbabaie, and George J. Pappas. “Minimal Reachability Problems”. In: *arXiv e-prints*, arXiv:1503.07021 (Mar. 2015), arXiv:1503.07021.

[148] Renée Vidal et al. “Controlled Invariance of Discrete Time Systems”. In: *Hybrid Systems: Computation and Control*. Ed. by Nancy Lynch and Bruce H. Krogh. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 437–451. ISBN: 978-3-540-46430-3.

[149] Mario E. Villanueva, Matthias A. Müller, and Boris Houska. *Configuration-Constrained Tube MPC*. 2022.

[150] Henk J. van Waarde, M. Kanat Camlibel, and Mehran Mesbahi. “From noisy data to feedback controllers: non-conservative design via a matrix S-lemma”. In: *IEEE TAC* (2020), pp. 1–1.

[151] Andreas Wächter and Lorenz T. Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical Programming* 106.1 (Apr. 2005), pp. 25–57.

[152] Shijie Wang, Zhikun She, and Shuzhi Sam Ge. "Estimating Minimal Domains of Attraction for Uncertain Nonlinear Systems". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51.12 (2021), pp. 7776–7787.

[153] Zheming Wang, Raphaël M. Jungers, and Chong-Jin Ong. "Computation of the maximal invariant set of linear systems with quasi-smooth nonlinear constraints". In: *2019 18th European Control Conference (ECC)*. 2019, pp. 3803–3808.

[154] D. White and Anand Anandalingam. "A penalty function approach for solving bi-level linear programs". In: *Journal of Global Optimization* 3 (Dec. 1993), pp. 397–419.

[155] Xiangrong Shen and M. Goldfarb. "Independent Stiffness and Force Control of Pneumatic Actuators for Contact Stability during Robot Manipulation". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. 2005, pp. 2697–2702.

[156] Mengwei Xu, Jane J. Ye, and Liwei Zhang. "Smoothing SQP Methods for Solving Degenerate Nonsmooth Constrained Optimization Problems with Applications to Bilevel Programs". In: *SIAM Journal on Optimization* 25.3 (2015), pp. 1388–1410.

[157] Bai Xue and Naijun Zhan. "Robust Invariant Sets Computation for Discrete-Time Perturbed Nonlinear Systems". In: *IEEE Transactions on Automatic Control* 67.2 (2022), pp. 1053–1060.

[158] Bai Xue and Naijun Zhan. *Robust Invariant Sets Computation for Switched Discrete-Time Polynomial Systems*. 2021.

[159] Liren Yang and Necmiye Ozay. "Scalable Zonotopic Under-Approximation of Backward Reachable Sets for Uncertain Linear Systems". In: *IEEE Control Systems Letters* 6 (2022), pp. 1555–1560.

[160] Liren Yang et al. "Parameter Sensitivity Analysis of Controlled Invariant Sets via Value Iteration". In: *2020 American Control Conference (ACC)*. 2020, pp. 4737–4744.

[161] Bin Yu, John E. Mitchell, and Jong-Shi Pang. *Solving Linear Programs with Complementarity Constraints using Branch-and-Cut*. 2018.

[162] Mario Zanon and Alberto Bemporad. *Constrained Control and Observer Design by Inverse Optimality*. 2021.

[163] Mario Zanon and Sébastien Gros. "On the Similarity Between Two Popular Tube MPC Formulations". In: *2021 European Control Conference (ECC)*. 2021, pp. 651–656.

[164] Mengjia Zhu, Alberto Bemporad, and Dario Piga. "Preference-based MPC calibration". In: *2021 European Control Conference (ECC)*. 2021, pp. 638–645.