Finite Horizon Robustness with Applications to Missile Engagements

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Jyot R. Buch

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Peter J. Seiler

December, 2021

Acknowledgements

I thank my advisor, Prof. Peter Seiler, for providing a stimulating environment to conduct research and guiding me through graduate school. Without his support, this thesis could not have taken shape. During all these graduate years, he provided necessary critical feedback, which has been very valuable. Prof. Seiler is an excellent advisor, and he deeply cares about students' success. He played an essential role in shaping my thinking by providing the right advice at the right time. I also thank him for hosting me as a visiting research scholar at the University of Michigan, Ann Arbor, during the last two years of my graduate school.

I thank Prof. Demoz Gebre-Egziabher, Prof. Ryan Caverly, Prof. Rajesh Rajmani for agreeing to be on my doctoral advising committee. Prof. Demoz advised me in the first year of graduate school and helped me sort out my career path in the initial stage of the program. I appreciate all the valuable discussions with him during the aerospace navigation and flight dynamics & controls class. I also thank Prof. Ryan Caverly for being a committee chair and helpful one-on-one conversations throughout graduate school. His comments during my research presentations have been precious. I thank Prof. Rajesh Rajmani for being part of the review committee, reading this thesis, and providing constructive feedback.

I acknowledge the U.S. Office of Naval Research (ONR) for funding and support through the grant N00014-18-1-2209. This project was jointly funded with Prof. Andrew Packard and Prof. Murat Arcak's research group at the University of California, Berkeley, under the title "Finite-Horizon Robustness: Moving Beyond Traditional Stability Analysis". During my first year of research, Prof. Packard provided valuable feedback and comments during our weekly team meetings. On September 30, 2019, he passed away after a long battle with cancer. Prof. Arcak has been a second advisor and

a great mentor during this project. I had an opportunity to collaborate with him and his research group. I also thank Dr. Douglas Philbrick from the U.S. Naval Air Warfare Center Weapons Division (NAWCWD) at China Lake for being a mentor and advisor during this project. His comments have been helpful and very valuable.

In addition to the committee members at Minnesota, I had excellent interactions with other faculties through courses and research/project discussions. I express my sincere gratitude to Dale Enns, Maziar Hemati, Derya Aksaray, Yohannes Ketema, Yousef Saad, Mingy Hong, Richard Linares, Andrew Lamperski, and Joseph Mueller for their time and valuable discussions. Special thanks to Dale Enns for sharing his expertise and insights on aerospace vehicle controls. At Michigan, I thank Prof. Dennis Bernstein for helpful discussions on adjoint operators. Finally, I thank Prof. Kamesh Subbarao for valuable discussions on capture region conditions and Prof. Harald Pfifer for his comments and correction on the robust synthesis work in this dissertation.

I thank Kate Schweidel at UC Berkeley and Shih-Chi Liao at the University of Michigan for our valuable collaboration and discussions during this project. Moreover, I would like to thank Emmanuel Sin, Alex Devonport, Galaxy Yin, and Adnane Saoud for helpful discussions during our research update meetings. At Minnesota, I thank fellow graduate students Alex, Manash, Narendra, Kerry, Raghu, Daniel, William, Pingyen, Jordan, Sanjana, Harish, Thanasi, Inchara, Peng, Aniketh, Debraj, Gunjan, Abhineet, Grace (and other colleagues whom I may have missed) for valuable academic discussions and fun moments during our shared time.

Finally, I thank my parents, brother Kushal and other family members for their endless support and patience during these years.

Dedication

To my parents Rashesh and Meena.

Abstract

This dissertation advances theoretical and computational tools for robustness analysis and synthesis on a finite time horizon. A motivation for this work is the reliable assessment of missile interceptor system performance, which will also enable the robust design of such systems. Typical performance metrics are infinite-horizon in nature, centered on stability, and rely on frequency-domain concepts, such as gain/phase margins. Such metrics can be inadequate for systems operating on finite-time horizons, as in many launch scenarios. Instead, this thesis focuses on time-domain metrics, e.g., bounds on the system's state at the final time of the horizon, while considering the impact of disturbances, model uncertainty/variability, and initial conditions. The proposed approach is to numerically linearize the dynamics along the trajectory to obtain a Linear Time-Varying (LTV) system. The analysis or synthesis is then performed on a linearized system that captures the dynamics up to the first-order perturbations around the nominal trajectory. This sacrifices some accuracy over the original nonlinear model but enables the use of linear system tools. The proposed worst-case LTV analysis also provides specific bad disturbances and uncertain parameters that can be further studied in high-fidelity nonlinear simulation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Missile Defense: The threat environment is rapidly evolving in many unprecedented ways, predominantly due to the enhancement of existing missile capabilities and more agile unmanned aerial vehicles. The primary goal of any missile defense system is to protect its homeland, civilization, and strategic assets (e.g., aircraft carriers). These complex engineering systems must detect, track, and engage with incoming threat missiles to destroy them before they reach their respective targets. Currently, one of the most common approaches is to use an interceptor missile to disable the threat by either colliding with it (i.e., hit-to-kill) or by exploding near it (i.e., directed fragmentation).

Current Limitations: The performance of a single interceptor-threat engagement may degrade due to multiple factors, including external disturbances (e.g., wind gusts), unmodeled flexible dynamics, sensor noise, tracking inaccuracies, actuator saturation, threat's evasive maneuvers, etc. This poses a significant concern on the accuracy of a single interceptor system. Therefore multiple interceptors are launched to increase the likelihood of success. However, this may not always be feasible; for example, a small navy ship may only have limited missile resources onboard. The alternative approaches include counter rocket, artillery, and mortar (C-RAM) systems or CIWS radar-controlled rapid-fire gun that fires multiple rounds until the threat is successfully identified as

destroyed. Such defense capabilities may easily get overwhelmed when attacked simultaneously by multiple threats. Some threat missiles can be highly maneuverable with multiple decoys and countermeasures, making them hard to be intercepted. Moreover, failure to react in a short amount of time can be catastrophic. In summary, current multi-layered missile defense systems heavily suffer from the lack of performance guarantees.

1.2 Motivation

Gap in Existing Analysis Paradigms: State of the art methods for achieving guaranteed stability for both deterministic and uncertain control systems are based on well-developed mathematical tools such as gain margin [4], disk margins [5], μ-analysis [6], etc. A typical paradigm is to linearize the nonlinear dynamics about an equilibrium point and use these methods for infinite horizon robustness analysis of the resulting linear time-invariant (LTI) system. Such analysis also provides specific bad uncertainties and disturbances that can be further analyzed in a high-fidelity nonlinear simulation. The low computational cost and availability of these tools in well-documented software such as MATLAB's Robust Control Toolbox [7] make them attractive for engineers and scientists to use in their workflow. Moreover, the quantities produced by these methods are commonly accepted as evidence that a system will perform as desired. For example, 6 dB of gain margin and 45^0 of phase margin requirements are typical for rigid body control loops such as aircraft pitch rate. However, these methodologies ultimately answer the same question: will an individual system be stable? Whether a particular system will adequately coordinate with other systems and achieve a mission objective must be answered by other means. The state-of-the-art method for answering this question is to obtain an empirical performance estimate through extensive system testing with random parameter variations. This approach is known as Monte-Carlo analysis. An advantage of this method is that it can account for nonlinearities, time-varying uncertainties, initial conditions, etc. The disadvantages of this method are that it can require an impractically large number of full mission simulations to cover enough parameter combinations. The confidence level in the result is only as good as the refinement of the high-dimensional parameter grid. There is always a possibility of missing some edge-case

behavior in which system performance is compromised. Overall, there is a significant gap between existing linearized analysis tools and high-fidelity nonlinear Monte-Carlo approaches, both in mathematical framework and numerically reliable computational tools.

Demand for System-Level Analysis Tools: Today's complex autonomous systems are being designed and developed by scientists and engineers to meet safety, stability, and critical performance requirements. With the growing use of onboard optimization and machine learning-based methods, it is critical to verify and validate (V&V) if the system will perform as desired in a worst-case situation. Moreover, there may be thousands of engineers involved in a particular project to design components and subsystems based on specific requirements individually. There is a need to verify the entire system's performance when everything is integrated. Thus, it is a prime requirement to investigate and develop more general system-level analysis tools to quickly assess the worst-case performance and minimize design iterations.

Broader Scope: In addition to missile interceptors, many uncertain nonlinear systems follow a finite horizon trajectory such as space launch, robotic manipulators, flexible aircraft, entry-descent-and-landing systems, autonomous vehicles, etc. When these systems fail to perform, the consequences range from a car accident to a failure to stop a nuclear strike. It is important to analyze the performance of these systems in the presence of unmodeled dynamics, external disturbances, sensor noise, uncertain initial conditions, and parametric/non-parametric model uncertainties. Thus, to embrace a broader scope, a two-link robot arm example is also used in this dissertation to demonstrate some newly developed capabilities.

1.3 Objective

Goal: The main objective of this research is to develop theoretical and computational tools for robustness analysis of systems that operate on a finite time horizon. The focus is to quickly and reliably compute an appropriate robustness metric to determine the worst-case performance. Such analysis can be used to complement existing Monte-Carlo approaches to discover edge-cases early in the design iterations or to determine binary outcomes (such as mission success or a failure in a worst-case setting).

Challenges: Overall, with many uncertainties, disturbances, and parameter variations, the worst-case analysis problem is nonlinear and nonconvex. There aren't any numerically reliable tools available for such analysis. Even if such tool exists, their scope is likely limited because they are either computationally slow with no convergence guarantees or apply to only academic examples. For instance, consider a nonlinear power algorithm applied on an F-16 aircraft for worst-case trajectory analysis [8]. This algorithm not only suffers from a lack of convergence guarantees but can also be computationally slow. It took approximately the same amount of time (4 to 4.5 hours) to yield a worst-case combination of parameters and wind gusts as the Monte-Carlo simulations.

Approach: Our approach is focused on numerically linearizing the dynamics of a system along the nominal trajectory and assessing the robustness of the resulting Linear Time-Varying (LTV) system. This linearized system captures only the first-order perturbations around the nominal trajectory. We leverage the linearity of a system to provide formal guarantees on the LTV performance by solving convex optimization problems. However, this is achieved at some sacrifice on the accuracy concerning the original uncertain nonlinear system (i.e., trading accuracy for computational benefit). Such approximate analysis requires only one nonlinear simulation and is significantly faster. Proposed linearized analysis on a finite horizon also provides a certificate of worst-case performance (e.g., specific "bad" disturbances, parameters, etc.) that can further be analyzed in a nonlinear simulation.

1.4 Thesis Overview

Each chapter of this dissertation starts with a short introduction of the underlying problem and a summary of the closely related work. Simulations and numerical examples are provided towards the end of each chapter to support the discussion.

To begin, Chapter 2 formulates the LTV analysis problem and introduces the two examples of LTV toolchains using MATLAB/Simulink tools. The examples include (a) an interceptor missile trying to hit a threat missile using standard guidance and control law and (b) a two-link robot arm following a pre-specified trajectory for the end-effector. Overall, this chapter gives a high-level overview of the LTVTools [9] software and how it can be used for robustness analysis and synthesis on a finite horizon.

Chapter 3 discusses two nominal performance metrics for the LTV system: (a) finite horizon H_2 norm, and (b) induced gains. The computational conditions for both are reviewed, and related issues are discussed for induced gain computation using existing methods. A faster algorithm is developed that combines complementary benefits of the existing computational methods. Numerical examples are provided to illustrate the effectiveness of the new algorithm.

Chapter 4 focuses on the robust performance of the finite-horizon uncertain LTV system. This chapter only considers worst-case induced gain as a metric; however, the approach can be extended to the worst-case H_2 norm. The model uncertainties are characterized using Integral Quadratic Constraints (IQCs), and a worst-case upper bound on the induced gain is computed using the existing algorithm in [10]. A power iteration lower bound algorithm is presented for the uncertain LTV system. Furthermore, a simpler and faster first-order sensitivity analysis approach is discussed for parametric uncertainties. This chapter concludes with an example of robustness degradation in output-feedback setting, which motivates the robust synthesis work presented in the next chapter.

Chapter 5 develops a new robust synthesis algorithm for uncertain LTV systems on finite horizons. The goal is to design a finite horizon output-feedback LTV controller that minimizes the impact of external disturbances and model uncertainties. The proposed algorithm iterates between a nominal synthesis step and an IQC-analysis step presented in the previous chapter. These steps are combined using the construction of a scaled plant. The robustness and performance trade-offs are demonstrated for the nonlinear two-link robot arm example.

Chapter 6 applies the robustness analysis tools developed in the previous chapters to the single interceptor and threat engagement. The worst-case interceptor disturbance is computed using LTV analysis, and the resulting impact is compared with the nonlinear Monte-Carlo simulations. The linear and nonlinear analysis gap is further studied by scaling the size of a worst-case disturbance. Finally, first-order sensitivity analysis is presented for the uncertainties in aerodynamic parameters for the interceptor missile.

Chapter 7 presents safety-critical control with sector-bounded uncertainties at the plant input. The approach characterizes the input-output behavior of input nonlinearities using point-wise in time quadratic constraints. Robust control barrier functions

(RCBFs) are presented to provide safety guarantees for the entire uncertainty set. This is done by ensuring that a safe action exists for all nonlinearities at the modeled uncertainty level. This combines a traditional robust control approach with the CBF methods for safety-critical control. As a result, we obtain more cautious trajectories close to the unsafe region. This approach is demonstrated using a lateral vehicle control example for obstacle avoidance.

Chapter 8 provides concluding remarks on this dissertation with some recommendations for future research.

1.5 Thesis Contributions

The key technical contributions [11, 12, 13, 14] of this dissertation are as follows:

1. **Nominal Analysis:** Chapter 3 proposes a new and more efficient algorithm to compute induced norms of finite-horizon LTV systems that utilizes the complementary benefits of the power iteration and RDE bisection methods (Section 3.8). This algorithm provides guaranteed bounds and is roughly 8 to 10 times faster than the existing RDE bisection algorithm for higher-order LTV systems. It is also shown that the power iteration algorithm converges monotonically to the induced system norm (Section 3.6). This is a stronger convergence result than existing results for power iterations on Hilbert spaces [15]. Finally, in this chapter, we show that a simpler computational condition can be obtained for a special case of induced $\mathcal{L}_2$-to-Euclidean gain (Section 3.8.2) using existing time-varying Gramian results (Section 22 of [16]). The proposed analysis tools are used in Section 6.3 to compute worst-case disturbances for the interceptor to miss the threat.

2. **Robust Analysis:** Chapter 4 reviews existing theoretical and computational conditions for the worst-case gain IQC upper bound [10, 17]. This chapter discusses the power iteration algorithm for computation of the worst-case gain lower bound, which is a special case of the work in [8] and is analogous to the skewed-μ lower bound problem [6, 18, 19, 20] for the structured singular value of the infinite horizon LTI system. This chapter also presents first-order trajectory sensitivity analysis for systems with uncertain parameters. This approach is applied

to the missile interceptor example in Section 6.4 to study the impact of uncertain aerodynamic parameters.

3. **Robust Synthesis:** There are three main technical contributions of Chapter 5. First, a new iterative algorithm is presented in Section 5.4.2 for the robust synthesis of LTV systems on finite horizon [12]. We use the dynamic IQC multipliers for the proposed algorithm, whereas our prior work in [11] used the memoryless IQCs and related classes of uncertainties. Second, we use a time-varying IQC factorization to construct a scaled plant. This step ensures that the worst-case gain at each iteration is monotonically non-increasing. Finally, this chapter provides all details and technical proofs regarding the proposed approach.

4. **LTVTools Software:** The computational tools developed in this thesis are available in the LTVTools [9] MATLAB toolbox. The initial (beta) version was first developed as part of the work in [17, 10]. During this research, several new capabilities are added and existing functions are enhanced for better user experience. A complete list of available functions is provided in Appendix D. The key contributions include new LTV analysis (e.g., `tvnorm`, `tvwcgain`) and synthesis functions (e.g., `tvrobsyn`) that link back to the algorithms developed in this dissertation. Other contributions include improved user interface for time-varying matrices (`tvmat`) and state-space systems (`tvss`) objects, plotting tools, additional demo examples, bug fixes, new Simulink library for LTV systems etc.

5. **Robust Safety-Critical Control:** The main contributions of Chapter 7 are as follows: First, a new robust control barrier function (RCBF) approach to handle sector-bounded uncertainties at the plant input is presented. This allows us to handle nonlinearities and time-varying memoryless uncertainties described by a quadratic constraint. Second, an optimization problem is formulated that minimally alters the baseline control command to guarantee safety in the presence of modeled input uncertainty. This optimization problem can be rewritten using a second-order cone program (SOCP) to be solved online. For a scalar input case, it is shown that the proposed SOCP is locally Lipschitz continuous. Finally, the proposed approach is demonstrated using a lateral vehicle control example to study robust safety and design trade-offs therein.

Chapter 2

Problem Formulation

2.1 Notation

Let $\mathbb{R}^{n\times m}$ and $\mathbb{S}^n$ denote the sets of n-by-m real matrices and n-by-n real, symmetric matrices. The inner product for a vector $w \in \mathbb{R}^m$ in Euclidean space is defined as $\langle w, w\rangle := w^\top w$. This defines a Euclidean norm as $\|w\|_2 := \sqrt{\langle w, w\rangle}$. Let $\mathcal{L}_2^n[0,T]$ denote the Hilbert space of Lebesgue integrable signals $v : [0,T] \to \mathbb{R}^n$ with inner product $\langle v, v\rangle := \int_0^T v(t)^\top v(t)\, dt$. This inner product defines a finite horizon $\mathcal{L}_2[0,T]$ norm of a signal as $\|v\|_{2,[0,T]} := \sqrt{\langle v, v\rangle}$. If $\|v\|_{2,[0,T]} < \infty$ then $v \in \mathcal{L}_2^n[0,T]$. The expected value of a random variable $v \in \mathbb{R}^n$ is denoted by $\mathcal{E}(v)$. The notation $\mathbb{RL}_\infty$ denotes the set of rational functions with real coefficients that are proper and have no poles on the imaginary axis. $\mathbb{RH}_\infty \subset \mathbb{RL}_\infty$ contains functions that are analytic in the closed right-half of the complex plane. The notations $\mathcal{F}_l(G,K)$ and $\mathcal{F}_u(N,\Delta)$ represents lower and upper Linear Fractional Transformations (LFTs) respectively as standard in robust control literature [21, 22]. The notation $G^\sim$ denotes the adjoint of a linear dynamical system G as formally defined in Section 3.2.4 of [23].

2.2 Uncertain Nonlinear Systems

Consider a continuous-time, nonlinear, time-varying, uncertain dynamical system described by the following ordinary differential equations:

$$\begin{aligned} \dot{\eta}(t) &= f(\eta(t), w(t), p(t), d(t), t), \qquad \eta(0) = \eta_0 \\ v(t) &= g_1(\eta(t), w(t), p(t), d(t), t) \\ \xi(t) &= g_2(\eta(t), w(t), p(t), d(t), t) \\ w(t) &= \Delta(v(t), t) \end{aligned} \tag{2.1}$$

where $\eta(t) \in \mathbb{R}^{n_\eta}$ is the nonlinear state, $w(t) \in \mathbb{R}^{n_w}$ is the uncertain input, $p(t) \in \mathbb{R}^{n_p}$ is the uncertain parameter vector, $d(t) \in \mathbb{R}^{n_d}$ is the external disturbance input, $v(t) \in \mathbb{R}^{n_v}$ is the uncertain output, $\xi(t) \in \mathbb{R}^{n_\xi}$ is the performance output of interest. Let $(\bar{\eta}, \bar{w} = 0, \bar{p}, \bar{d} = 0, \bar{v}, \bar{\xi})$ represent the nominal values of the respective variables along the trajectory. Our goal is to quickly analyze the performance of this system against the most problematic components such as unmodeled dynamics $\Delta \in \mathbf{\Delta}$, uncertain parameters $p \in \mathcal{P}$, stochastic and/or deterministic disturbances $d \in \mathcal{L}_2^{n_d}[0, T]$, and uncertain initial conditions $\eta(0) \in \{\eta_0 \,|\, (\eta_0 - c)^\top E_0 (\eta_0 - c) \leq 1, E_0 = E_0^\top \succ 0, c \in \mathbb{R}^{n_\eta}\}$.

2.3 Summary of Related Work

The most related work for robustness analysis of uncertain nonlinear systems along the trajectory is summarized here. Many metrics can be defined to measure the performance of a system. The most common performance metric is a mission success or a failure, often defined by achieving a certain outcome at the end of the trajectory. As noted earlier, with many uncertainties, external disturbances, and uncertain initial conditions, the performance analysis problem is high-dimensional and non-convex. A typical analysis paradigm is to use Monte-Carlo simulations [24, 25, 26]. This provides probabilistic guarantees on the performance of a system. An alternative but the complementary approach is to perform worst-case analysis. There are only a few numerically reliable tools available for worst-case analysis. One of the related works is the forward reachability analysis [27]. The work in [27] uses IQCs and Sum-of-Squares (SOS) techniques to compute the tightest outer bound on the robust reachable set.

However, these algorithms do not scale well with the state dimension. The work in [28] focuses on the parametric uncertainties and proposes a second-order trajectory sensitivity analysis approach. This work assumes analytical expressions of the sensitivity matrix, which may not be available for the complex dynamical system. Most related works in LTV worst-case analysis are [10, 17, 29] and [30, 31], which will be reviewed later in a specific chapter. Another popular direction is to use machine learning and data-driven approaches to provide guarantees on the performance of a nonlinear system by designing a classifier [2] or creating a surrogate model [32]. These approaches suffer from a lack of formal guarantees, which is also an area of active research [33].

2.4 Finite Horizon LTV Approximations

The proposed approach in this dissertation considers numerical linearization of nonlinear dynamics along the trajectory to obtain an LTV system. The worst-case LTV behavior is individually analyzed for disturbances, uncertainties, and uncertain initial conditions. For example to study the impact of disturbances, define a nominal trajectory ($\bar{\eta}$, $\bar{d} = 0$) on a time horizon $[0, T]$ and consider the Taylor series expansion of Equation (2.1) to obtain the following approximation:

$$\begin{aligned} \delta\dot{\eta}(t) &= \left.\frac{\partial f}{\partial \eta}\right|_{(\bar{\eta},\, \bar{d}=0)} \delta\eta(t) + \left.\frac{\partial f}{\partial d}\right|_{(\bar{\eta},\, \bar{d}=0)} d(t), \qquad \delta\eta(0) = 0 \\ \delta\xi(t) &= \left.\frac{\partial g}{\partial \eta}\right|_{(\bar{\eta},\, \bar{d}=0)} \delta\eta(t) + \left.\frac{\partial g}{\partial d}\right|_{(\bar{\eta},\, \bar{d}=0)} d(t) \end{aligned} \tag{2.2}$$

where $\delta\eta := \eta - \bar{\eta}$, $\delta\xi := \xi - \bar{\xi}$ represents the first order perturbations in state and performance output respectively. Equation (2.2) defines a linear time-varying system which can be analyzed using linear analysis tools presented in Chapter 3. For uncertain LTV systems in Chapter 4 and 5, we continue to use the Linear Fractional Transformations (LFT) framework as in standard in robust control literature [34, 22, 21]. The notation $x := \delta\eta$ and $e := \delta\xi \in \mathbb{R}^{n_e}$ (with $n_e \equiv n_\xi$) are used to denote the linearized system state vector and error outputs of interests in Chapter 3, 4 and 5. It is important to note that a shift in the nominal trajectory must be accounted for when analyzing the linearized interconnection of the uncertain system to reasonably match the analysis

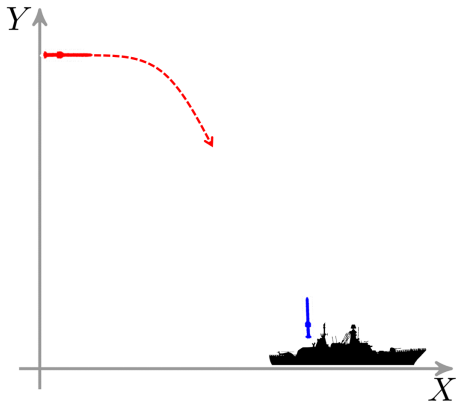

Figure 2.1: Planar Engagement Setup

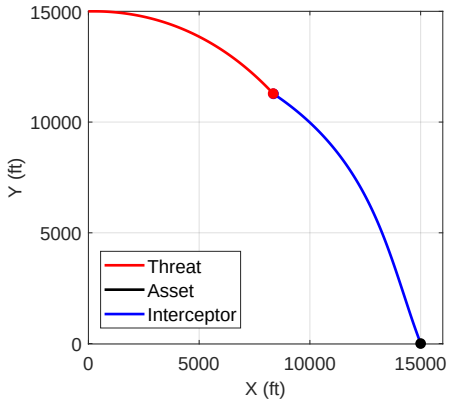

Figure 2.2: Nominal Trajectories

with the original uncertain nonlinear system. This shift can be captured in the non-zero initial conditions, which results in an affine LTV system from input to the output. The work in this dissertation assumes zero initial conditions for the LTV system and does not explicitly handle non-zero initial condition cases. See reference [31] for more details.

2.5 Example 1: Single Interceptor-Threat Engagement

2.5.1 Model Development

Consider a planar engagement scenario as shown in the Figure 2.1. A threat missile (red) is headed towards a static asset (black ship). The interceptor missile (blue) can be launched from the ship to engage the threat. Both threat and interceptor missiles are assumed to have identical models, guidance, and control loops but different initial conditions. A planar version of the generic missile model from [1] is used for both missiles. The motion of a threat or an interceptor missile (in the body fixed xz-plane Figure 2.4) is governed by the following equations:

$$\dot{\alpha} = \frac{1}{mV_m}\left(F_z \cos\alpha - F_x \sin\alpha\right) + q \tag{2.3}$$

$$\dot{q} = \frac{M_y}{I_{yy}} \tag{2.4}$$

$$\dot{V}_m = \frac{1}{mV_m}\left(F_z \sin\alpha + F_x \cos\alpha\right) \tag{2.5}$$

where α is the angle of attack, q is the pitch rate, and V_m is the airspeed. The forces and moments are calculated as follows:

$$F_x = -\frac{1}{2}\rho V_m^2 A \left(C_{\text{dap}} + C_D\right) \tag{2.6}$$

$$F_z = \frac{1}{2}\rho V_m^2 A C_{Nz} \tag{2.7}$$

$$M_y = \frac{1}{2}\rho V_m^2 A l \left(C_{My} + \frac{C_{Mtd} l}{2V_m} q\right) \tag{2.8}$$

and the aerodynamic coefficients are given by:

$$C_{Nz}(\alpha, \delta) = -21\alpha - 24.5\alpha^3 - \frac{12}{\pi}\delta \tag{2.9}$$

$$C_{dap}(\alpha, \delta) = 0.5(\alpha + \delta)^2 \tag{2.10}$$

$$C_{My}(\alpha, \delta) = -15.75\alpha - 36.75\alpha^3 - \frac{84}{\pi}\delta \tag{2.11}$$

$$C_{Mtd} = -720 \tag{2.12}$$

$$C_D = 0.3 \tag{2.13}$$

where $\frac{1}{2}\rho V_m^2$ is the dynamic pressure, A and l are the reference area and distance, respectively, and δ is the fin deflection angle. A standard two-loop design (Figure 2.3) is used for both missiles in which inner loop is a control loop and outer loop is a proportional navigation (PN) guidance law. The PN guidance law sends acceleration commands to the interceptor [35, 36, 37]. The acceleration command prescribed by PN is $\eta_c = -N\dot{R}\dot{\lambda}$, where η_c is the commanded acceleration, N is the PN ratio, R is the range between the threat and interceptor, and λ is the Line-of-Sight (LOS) angle. These quantities are shown in Figure 2.5, where all angles are measured with respect to the inertial frame.

The acceleration command from the guidance law is converted into control surface deflections by the missile autopilot. A standard three-loop topology is used for the autopilot [38], which is a controller for the interceptor's lateral acceleration. It is designed using a linearization of the missile about the equilibrium point at a fixed angle of attack and airspeed. Gains are scheduled on these variables and stored in interpolation tables.

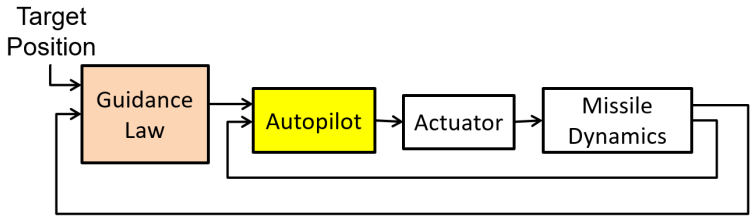

Figure 2.3: Block Diagram of the Missile with Autopilot and Guidance Law

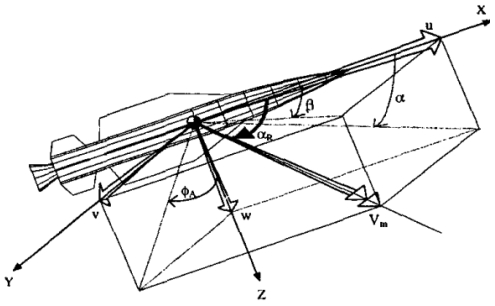

Figure 2.4: Missile Body-fixed Axis [1]

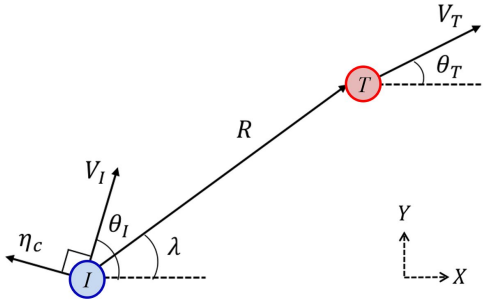

Figure 2.5: Inertial Frame [2]

The autopilot design parameters are open-loop crossover frequency, closed-loop damping ratio, and ratio (in the magnitudes) of the complex and real pole of the closed loop. The fin-actuator is subject to the rate and saturation limits. Air density is assumed constant for all altitudes. The effect of earth's gravity is ignored in this model.

The entire engagement can be modeled* using Simulink (Figure 2.6) to represent a specific head-on engagement scenario. In this model, a threat missile uses PN guidance to hit the asset located at $(15000, 0)$ in Figure 2.2. The interceptor launched from the asset uses PN guidance to hit the threat. All other properties such as autopilot and actuator parameters are assumed the same for both missiles. In simulation, the interceptor missile is assumed to have higher velocity magnitude ($V_I = 2750$ ft/s) compared to the threat missile ($V_T = 2000$ ft/s). Such maneuvering advantage is required for the interceptor to capture the threat successfully using PN guidance law [36].

Simulation can be stopped when a missile is within a specified capture radius (e.g., 20 feet) of its respective target. Figure 2.2 shows the nominal hit of an interceptor with

*Developed in collaboration with K. Schweidel, A. Packard, P. Seiler, M. Arcak and D. Philbrick.

the threat missile at $T = 5.208$ seconds. A more advanced stopping condition can also be used, in which the model can be stopped when the range rate ($\dot{R}$) is positive between the missile and its target.

2.5.2 LTV Toolchain

To demonstrate the LTV toolchain, consider analyzing the plant-input disturbances for the interceptor. We are interested in studying the impact of such disturbances on the relative position of the interceptor and threat at the final time of intercept. The following code snippet in MATLAB demonstrates how a model can be linearized around the nominal trajectory to perform an LTV analysis. The linearization input and outputs are shown using a red circle in Figure 2.6.

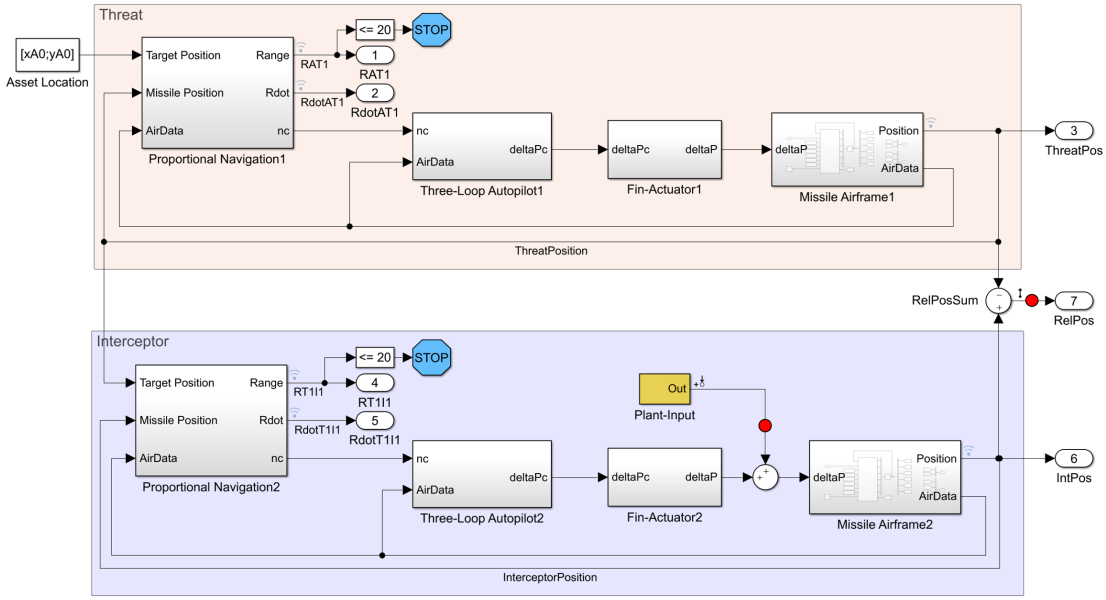

Figure 2.6: Planar Engagement Simulink Model

We will revisit this example in Chapter 6 to compute an appropriate metric and related worst-case disturbance input that maximizes the perpendicular component of the Line-of-Sight (LOS) vector at the final time (i.e., nominal intercept time). Such

```
% Load Simulink model with nominal parameters
mdl = 'mPlanarSim';
load_system(mdl);

% Perform nominal simulation
timeGrid = sim(mdl);

% Define linearization inputs and outputs
io(1) = linio([mdl '/Interceptor/Plant-Input'],1,'input');
io(2) = linio([mdl '/RelPosSum'],1,'output');

% Perform snapshot linearization
sys = linearize(mdl,io,timeGrid);

% LTV system with 1 input and 2 outputs
G = tvss(sys,sys.SamplingGrid.Time);

% Perform LTV analysis by computing induced system norm
[gE,dwc] = tvnorm(G,2);
```

Figure 2.7: MATLAB Code for LTV Analysis

disturbances can be scaled appropriately to induce a desired miss-distance in the LTV simulation.

Remark 2.5.1. *The engagement model in Figure 2.6 is of order* 34*, which includes states for a threat and the interceptor missile. This example (used only for demonstration purposes) is realistic in the sense that the threat has nonlinear dynamics with its guidance and control law, as opposed to a simple double integrator model considered in [29]. We assume that the perfect information for the state is available. However, the analysis workflow can be extended to more realistic approximations of the engagement, e.g., the interceptor model can include propulsion, gravity model, sensor dynamics, and navigation filters, etc. The assumptions on a threat missile can also be relaxed, e.g., it can simply be a projectile with some uncertainty bounds (depending on the accuracy of a tracking radar), which can be propagated along the trajectory to study their impact on the endgame. Similarly, the engagement model can also include the dynamics for the asset (e.g., aircraft carrier, fighter jet, satellite).*

2.6 Example 2: Two-Link Robot

2.6.1 Model Development

Consider an example of a two-link robot arm as shown in Figure 2.8. This example was considered in [10, 17] and is further studied in this research. The mass and moment of inertia of the i^{th} link are denoted by m_i and I_i. The robot properties are $m_1 = 3kg$, $m_2 = 2kg$, $l_1 = l_2 = 0.3m$, $r_1 = r_2 = 0.15m$, $I_1 = 0.09kg \cdot m^2$, and $I_2 = 0.06kg \cdot m^2$. The nonlinear equations of motion [3] for the robot are given by:

$$\begin{bmatrix} \alpha + 2\beta\cos(\theta_2) & \delta + \beta\cos(\theta_2) \\ \delta + \beta\cos(\theta_2) & \delta \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -\beta\sin(\theta_2)\dot{\theta}_2 & -\beta\sin(\theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \\ \beta\sin(\theta_2)\dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

$$\begin{aligned} \text{with } \alpha &:= I_1 + I_2 + m_1 r_1^2 + m_2(l_1^2 + r_2^2) = 0.4425\, kg \cdot m^2 \\ \beta &:= m_2 l_1 r_2 = 0.09\, kg \cdot m^2 \\ \delta &:= I_2 + m_2 r_2^2 = 0.105\, kg \cdot m^2. \end{aligned}$$

The state and input are $\eta := [\theta_1\ \theta_2\ \dot{\theta}_1\ \dot{\theta}_2]^\top$ and $\tau := [\tau_1\ \tau_2]^\top$, where τ_i is the torque applied to the base of link i. A trajectory $\bar{\eta}$ of duration 5 second was selected for the tip of the arm to follow. This trajectory is shown in solid black (Figure 2.9).

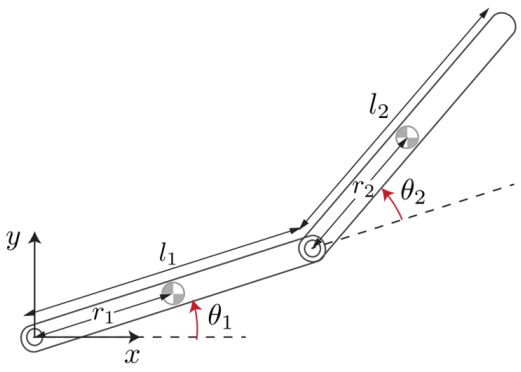

Figure 2.8: Two-link Planar Robot [3]

The equilibrium input torque $\bar{\tau}$ can be computed using inverse kinematics. The robot should track this trajectory in the presence of small torque disturbances d. The

input torque vector is $\tau = \bar{\tau} + u + d$ where u is an additional control torque to reject the disturbances. The nonlinear dynamics are linearized around the trajectory $(\bar{\eta}, \bar{\tau})$ to obtain an LTV system G:

$$\dot{x}(t) = A(t)x(t) + B(t)\left(u(t) + d(t)\right) \tag{2.14}$$

where $x(t) := \eta(t) - \bar{\eta}(t)$ is the deviation from the equilibrium trajectory. This is an example of a "gridded" LTV system which often arises from a linearization of a nonlinear dynamic model along a trajectory. Linear interpolation is used between the grid points to approximate the system dynamics during integration. A finite horizon time-varying LQR controller is designed to reject input disturbances. The details for the control design can be found in [10, 17]. The nominal trajectory in the joint space is also shown

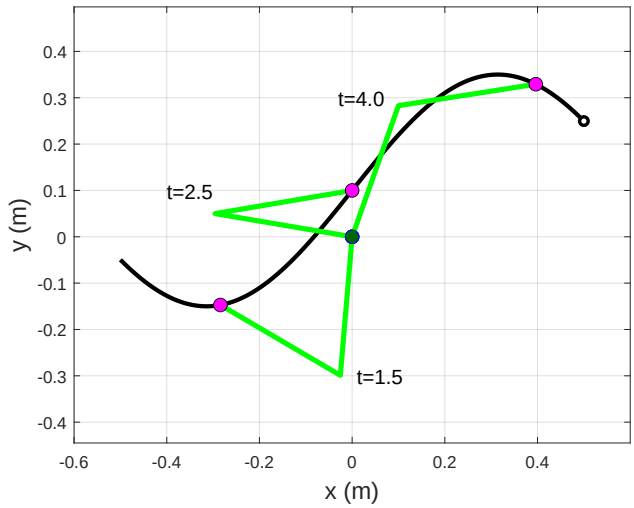

Figure 2.9: Nominal Trajectory in Cartesian Space with Snapshot Positions

in Figure 2.10. Analysis performed in the joint space can be translated to the Cartesian (task) space using forward kinematics of the robot.

2.6.2 LTV Toolchain

Consider the Simulink model of the closed-loop system as shown in Figure 2.11. The pre-computed (nominal) values of the state and input are $\bar{\eta}$ and $\bar{\tau}$ respectively. Moreover,

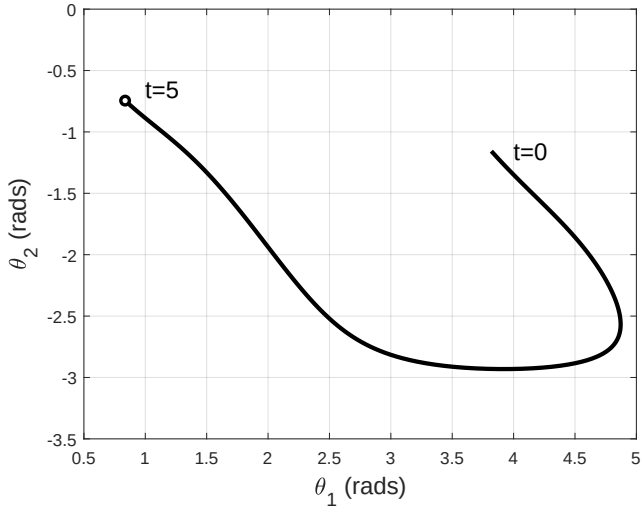

Figure 2.10: Nominal Trajectory in Joint Space

the disturbance $d(t) = \left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right]$ and controller $K(t) = 0_{2\times 4}$ are initialized for any $t \in [0, 5]$. This results into an open-loop model to be simulated on the horizon of $T = 5$ seconds. The MATLAB code in Figure 2.12 demonstrates the LTV synthesis workflow [9]. A specific example of designing finite horizon LQR controller is shown, however alternative synthesis methods (including the one presented in Chapter 5) can be considered.

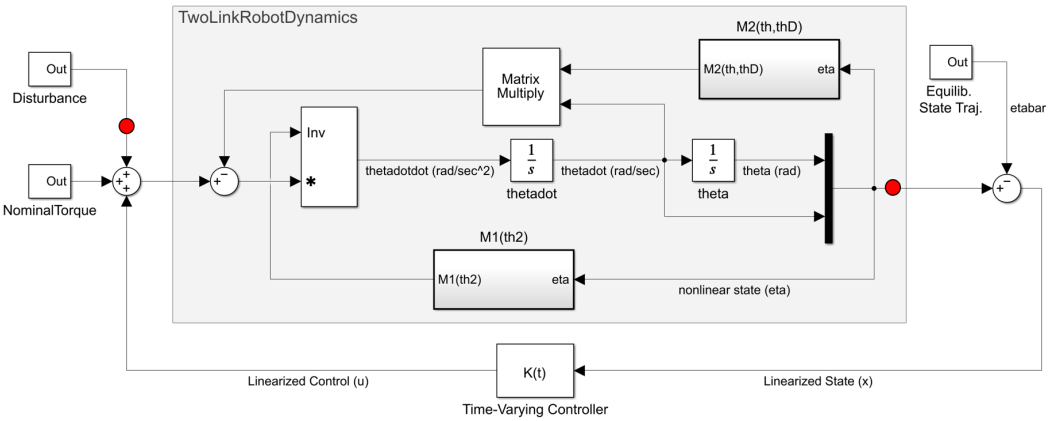

Figure 2.11: Two-Link Robot Simulink Model

```
% Load Simulink model with nominal parameters
mdl = 'TwoLinkRobotCL';
load_system(mdl);

% Perform nominal (open-loop) simulation
timeGrid = sim(mdl);

% Define input/output and perform snapshot linearization
io(1) = linio([mdl '/Disturbance'],1,'input');
io(2) = linio([mdl '/Mux'],1,'output');
sys = linearize(mdl,io,timeGrid);

% LTV system with 2 input and 4 outputs
G = tvss(sys,sys.SamplingGrid.Time);

% Design finite-horizon LQR controller
Q = 100*diag([1 1 0.1 0.1]);
R = 0.1*diag([1,1]);
F = diag([1 1 0.1 0.1]);
Klqr = tvlqr(G,Q,R,[],F,[0,5]);

% Simulate nonlinear system with designed LTV controller
K = Klqr;
sim(mdl);
```

Figure 2.12: MATLAB Code for Finite Horizon LQR Synthesis

2.7 Summary

This chapter provided a high-level overview of the robustness analysis problem and the LTV toolchain using MATLAB and Simulink tools. The MATLAB toolbox LTVTools [9] provides additional commands for analysis and robust synthesis for LTV systems. A complete list of functions, including a summary of the Simulink library for the LTV system, is available in Appendix D.

Chapter 3

Nominal Performance

3.1 Introduction

This chapter presents two useful metrics to assess the performance of a finite horizon Linear Time-Varying (LTV) system (Section 3.3). These include finite horizon H_2 norm and induced gain of an LTV system. The computational approaches of these metrics are discussed in Section 3.4 and 3.5. Two existing approaches for computing LTV induced norms are the power iteration and the bisection method. The power iteration repeatedly integrates the dynamics of the LTV system and a related adjoint system. This amplifies the input in the largest gain direction so that the iterates converge to the worst-case disturbance input. The bisection method relies on a related matrix Riccati Differential Equation (RDE) condition. Upper or lower bounds on the induced gain are obtained based on the existence of a solution to the RDE. The induced gain can be computed within the desired tolerance via bisection. These methods have complementary properties as discussed in Section 3.7. Each step of the power iteration is fast, but overall convergence can be slow. Moreover, the power iteration only computes a lower bound with no guarantee on the gap to the induced gain. In comparison, the RDE bisection method provides guaranteed upper and lower bounds. However, a single RDE integration is slow for higher-order systems. A new and more efficient algorithm is presented in Section 3.8 which utilizes the complementary benefits of the power iteration and RDE bisection method.

3.2 Summary of Related Work

Among the most closely related work is [8], which uses the power iteration to compute induced gains of uncertain nonlinear, time-varying systems. There are no convergence guarantees for such a general class of systems. The restriction to LTV systems allows us to provide a convergence proof for the power iteration (Theorem 3.6.1) and to use the RDE to compute provable upper bounds on the induced gain. The work in [39] also considers computing induced norms for finite-horizon LTV systems. The algorithm in [39] relies on an iteration using a different RDE. Again, no convergence proof is provided. Moreover, the computational cost of the RDE in [39] scales similarly to the RDE in the bisection method. Other related work includes computing induced gain of sampled-data systems [40] as well as LTI systems on infinite horizons [41, 42] and finite horizons [43].

3.3 Linear Time-Varying (LTV) Systems

Consider an LTV System G defined on horizon $[0, T]$:

$$\dot{x}(t) = A(t)\,x(t) + B(t)\,d(t) \tag{3.1}$$

$$e(t) = C(t)\,x(t) + D(t)\,d(t) \tag{3.2}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state, $d(t) \in \mathbb{R}^{n_d}$ is the external disturbance input, and $e(t) \in \mathbb{R}^{n_e}$ is the error output at time t. The state matrices $A : [0,T] \to \mathbb{R}^{n_x \times n_x}$, $B : [0,T] \to \mathbb{R}^{n_x \times n_d}$, $C : [0,T] \to \mathbb{R}^{n_e \times n_x}$, and $D : [0,T] \to \mathbb{R}^{n_e \times n_d}$ are piecewise-continuous (bounded) matrix-valued functions of time. Explicit time dependence of the state matrices will be omitted when it is clear from the context. It is assumed that the horizon is finite i.e., $T < \infty$. The input, output, and state dimensions of G are assumed to be constant throughout the horizon. Many performance metrics can be defined for the system G. The two most useful notions of robustness are discussed in the next subsections. These include finite horizon H_2-norm, which represents a response to the stochastic inputs (e.g., white noise), and induced system gains, representing a response to the deterministic inputs.

3.3.1 Finite Horizon H_2 Norm

Assume that the system G is excited by white noise input signal d with unit covariance i.e. $\mathcal{E}[d(t)] = 0$ and $\mathcal{E}[d(t)d(t)^\top] = I_{n_d}, \forall t \in [0, T]$. Further, assume that initial condition for the state is completely known i.e. $\mathcal{E}[x(0)] = 0$ and $\mathcal{E}[x(0)x(0)^\top] = 0$. Let $D(T) \equiv 0$. The finite horizon H_2 norm of G is then defined as:

$$\|G\|_{H_2,[0,T]} := \sqrt{\mathcal{E}\left[trace(e(T)\, e(T)^\top)\right]} \tag{3.3}$$

This definition is closely related to the output variance at the final time T. For stable LTI systems on a sufficiently long horizon, this quantity converges to the infinite horizon H_2 norm, which has a well-known frequency domain interpretation.

3.3.2 Induced Gain

Assume that the system G is excited by the deterministic input $d \in \mathcal{L}_2^{n_d}[0, T]$. The performance can be assessed by splitting the output in two components:

$$\begin{bmatrix} e_I(t) \\ e_E(t) \end{bmatrix} = \begin{bmatrix} C_I(t) \\ C_E(t) \end{bmatrix} x(t) + \begin{bmatrix} D_I(t) \\ 0 \end{bmatrix} d(t) \tag{3.4}$$

where $e_I(t) \in \mathbb{R}^{n_I}$ and $e_E(t) \in \mathbb{R}^{n_E}$ with $n_e = n_E + n_I$. The generalized induced performance measure of $G : \mathcal{L}_2^{n_d}[0, T] \to \mathbb{R}^{n_E} \oplus \mathcal{L}_2^{n_I}[0, T]$ is then defined as:

$$\|G\|_{[0,T]} := \sup_{\substack{0 \neq d \in \mathcal{L}_2[0,T] \\ x(0)=0}} \left[\frac{\|e_E(T)\|_2^2 + \|e_I\|_{2,[0,T]}^2}{\|d\|_{2,[0,T]}^2} \right]^{\frac{1}{2}} \tag{3.5}$$

This defines an induced gain from the input d to a mixture of an $\mathcal{L}_2$ and terminal Euclidean norm on the output e. Note that if $n_E = 0$, there is no terminal Euclidean norm penalty on the output. This case corresponds to the standard, finite-horizon induced-$\mathcal{L}_2$ gain* of G [23]. Similarly, if $n_I = 0$, then there is no $\mathcal{L}_2$ penalty on the output. This case corresponds to a finite-horizon $\mathcal{L}_2$-to-Euclidean gain. This can be

*For LTI systems on infinite horizon, the induced-$\mathcal{L}_2$ gain is equivalent to the H_∞ system norm, which has a well-known frequency domain interpretation [21, 42, 44]. For SISO systems, this represents a peak on the bode magnitude plot.

used to bound the terminal output $e_E(T)$ reachable by an $\mathcal{L}_2$ disturbance input. Zero feed-through from d to e_E ensures that Euclidean penalty is well-defined at any time $t \in [0, T]$. The system is assumed to be initially at rest, i.e., $x(0) = 0$. Non-zero initial conditions similar to [45] can also be handled with minor modifications.

3.4 Computation of H_2 Norm

Theorem 3.3.1 of [23] considers an alternative definition of the H_2 norm in which computational conditions are provided in terms of the controllability or observability Gramian. With minor modifications, we obtain the computational conditions for H_2 norm (3.3) as follows. Let $X : [0, T] \to \mathbb{S}^{n_x}$ be the differentiable function that satisfies:

$$\dot{X}(t) = A(t)\,X(t) + X(t)\,A(t)^\top + B(t)\,B(t)^\top, \quad X(0) = 0 \tag{3.6}$$

This is a Lyapunov Differential Equation (LDE) associated with the finite horizon state controllability Gramian. It has a unique solution of the form:

$$X(t) = \int_0^t \Phi(t,\tau) B(\tau) B(\tau)^T \Phi(t,\tau)^T \, d\tau \tag{3.7}$$

where $\Phi(t, \tau)$ denotes the unique state transition matrix of the unforced system $\dot{x}(t) = A(t)\,x(t)$ satisfying the following differential equation:

$$\frac{d\Phi(t,\tau)}{dt} = A(t)\,\Phi(t,\tau), \qquad \Phi(\tau,\tau) = I \tag{3.8}$$

Define $Y(t) := C(t)X(t)C(t)^T$ as the output controllability Gramian. The finite horizon H_2 norm can be computed as:

$$\|G\|_{H_2,[0,T]} = \sqrt{trace(Y(T))} \tag{3.9}$$

Note that this computational condition involves only one forward integration of the LDE 3.6 which standard ODE integration solvers can efficiently compute. An alternative but equivalent conditions can be obtained using LDE for the finite horizon observability Gramian.

3.5 Computation of Induced Gain

This section reviews existing methods to compute the induced gain of an LTV system. Note that Equation (3.5) is related to the following optimal control problem [10]:

$$J^* \triangleq \sup_d \frac{1}{2}\left[e_E(T)^\top e_E(T) + \int_0^T e_I(t)^\top e_I(t)\,dt\right]$$
$$s.t.\ \text{Equation (3.1), (3.4) with } x(0) = 0, \|d\|_{2,[0,T]} = 1$$

The optimal cost and induced norm are related as $J^* = \frac{1}{2}\|G\|^2_{[0,T]}$. The standard Euler-Lagrange optimization framework (Section 2.5 of [46]) can be used to solve this problem. The Hamiltonian is defined using the quadratic performance cost and Lagrange multipliers for constraints. A two-point boundary-value problem is obtained from the necessary conditions for optimality. This leads to two numerical algorithms for computing the induced norm of G: (i) the power iteration, and (ii) bisection on a related Riccati Differential Equation. These two algorithms are briefly summarized in the following subsections.

3.5.1 Power Iteration

The power iteration is an iterative method to approximate the maximum eigenvalue and singular value for matrices within numerical tolerance [47, 48]. It was studied in [8] for computing the gain of uncertain, nonlinear time-varying systems. A version of the power iteration can also be used to approximate the induced gain of an LTV system within numerical tolerance. The adjoint (costate) dynamics $G^\sim : \mathbb{R}^{n_E} \oplus \mathcal{L}_2^{n_I}[0,T] \to \mathcal{L}_2^{n_d}[0,T]$ obtained from the two-point boundary-value problem are as follows:

$$\dot{p}(t) = -A(t)^\top p(t) - C_I(t)^\top q(t) \tag{3.10}$$
$$r(t) = B(t)^\top p(t) + D_I(t)^\top q(t) \tag{3.11}$$

with boundary condition:

$$p(T) = C_E(T)^\top e_E(T) \tag{3.12}$$

where $p(t) \in \mathbb{R}^{n_x}, q(t) \in \mathbb{R}^{n_I}, r(t) \in \mathbb{R}^{n_d}$ are adjoint state, input and output respectively. The specific steps for power iteration are outlined in Algorithm 1.

Algorithm 1 Power Iteration for LTV System

1: **Given:** G, $G^{\sim}$
2: **Initialize:** $d^{(1)}$ with $\|d^{(1)}\|_{2,[0,T]} = 1$, N, ϵ_a, $\gamma^{(0)} = -\infty$
3: **for** $i = 1 : N$ **do**
4: **Forward Sim:** Simulate G from $t = 0$ to $t = T$ with $x(0) = 0$ and input $d^{(i)}$ to generate $e_I^{(i)}$, $e_E^{(i)}$, and to compute forward performance $\gamma_f^{(i)}$ using Equation (3.13).
5: **Backward Sim:** Simulate $G^{\sim}$ from $t = T$ to $t = 0$ with $p(T) \equiv C_E(T)^{\top} e_E^{(i)}(T)$ and input $q^{(i)} \equiv e_I^{(i)}$ to generate $r^{(i)}$ using Equation (3.11).
6: **Compute Gain:** $\gamma^{(i)} = \|r^{(i)}\|_{2,[0,T]}$.
7: **Alignment Condition:** $d^{(i+1)} = r^{(i)}/\gamma^{(i)}$.
8: **Stop Condition:** Terminate if $\gamma^{(i)} - \gamma^{(i-1)} < \epsilon_a$.
9: **end for**
10: **Compute Output:** Simulate G using $d_o = d^{(i+1)}$ and compute forward performance γ_o using Equation (3.13).
11: **Output:** γ_o, d_o.

Let system G and its adjoint $G^{\sim}$ be given. The algorithm is initialized with a candidate disturbance $d^{(1)}$, maximum number of iterations N, initial performance $\gamma^{(0)} = -\infty$ and desired absolute tolerance ϵ_a. The initial disturbance can be chosen randomly or by any other means. It is normalized so that $\|d^{(1)}\|_{2,[0,T]} = 1$. The first step in the iteration is to simulate the system G forward in time from $t = 0$ to $t = T$ using chosen disturbance $d^{(i)}$ and zero initial conditions. This step yields the outputs $e_I^{(i)}$ and $e_E^{(i)}$. The forward induced performance $\gamma_f^{(i)}$ can be computed for this specific unit-norm disturbance.

$$\gamma_f^{(i)} := \left[\|e_E^{(i)}(T)\|_2^2 + \|e_I^{(i)}\|_{2,[0,T]}^2 \right]^{\frac{1}{2}} \tag{3.13}$$

The next step involves backward simulation of the adjoint dynamics $G^{\sim}$ from $t = T$ to $t = 0$ using input $q^{(i)} \equiv e_I^{(i)}$ and boundary condition for $p(T) \equiv C_E(T)^{\top} e_E^{(i)}(T)$. This gives the adjoint output $r^{(i)}$. The gain $\gamma^{(i)}$ is given by the norm of the signal $r^{(i)}$. The disturbance for the next iteration $d^{(i+1)}$ is obtained by normalizing $r^{(i)}$. The iterations are terminated if the performance γ fails to increase by more than the specified absolute

tolerance. In addition, the iterations also stop if the number of iterations exceed specified N. The final outputs are induced gain γ_o and the corresponding worst-case disturbance d_o. The repeated evaluations of G and its adjoint $G^{\sim}$ amplify the disturbance. Moreover, the alignment condition ensures that the disturbance $d^{(i)}$ does not blow up and gets aligned along the direction of the largest gain as iteration progresses. It is known that the related matrix power iteration for singular values converges to the induced Euclidean norm of the matrix under mild technical conditions [47, 48]. In Section 3.6, we show that the performance $\gamma^{(i)}$ is monotonically non-decreasing, and power iteration for LTV system G converges to the induced gain. If the iterative loop in Algorithm 1 is terminated early, then γ_o is a lower bound on the actual gain $\|G\|_{[0,T]}$. Numerical integration issues may arise on a significantly long horizon if G is an unstable LTI system.

3.5.2 Bisection on the Riccati Differential Equation (RDE)

Due to linear dynamics, the adjoint solution can be obtained as $p(t) = P(t)\,x(t)$ where $P(t)$ is a time-varying solution to a related RDE. The next theorem states an equivalence between a bound on the performance $\|G\|_{[0,T]}$ and the existence of a solution to a related RDE [45, 23, 17, 10].

Theorem 3.5.1. *Consider an LTV system* (3.1) *with* $\gamma > 0$ *given. Let* $Q : [0,T] \to \mathbb{S}^{n_x}$, $S : [0,T] \to \mathbb{R}^{n_x \times n_d}$, $R : [0,T] \to \mathbb{S}^{n_d}$, *and* $F \in \mathbb{R}^{n_x \times n_x}$ *be defined as follows.*

$$Q := C_I^\top C_I, \quad S := C_I^\top D_I, \quad R := D_I^\top D_I - \gamma^2 I_{n_d}, \quad F := C_E(T)^\top C_E(T)$$

The following statements are equivalent:

1. $\|G\|_{[0,T]} < \gamma$
2. $R(t) < 0$ *for all* $t \in [0,T]$. *Moreover, there exists a differentiable function* $P : [0,T] \to \mathbb{S}^{n_x}$ *such that*

$$\dot{P} + A^\top P + PA + Q - (PB + S)R^{-1}(PB + S)^\top = 0, \qquad P(T) = F$$

This is a Riccati Differential Equation (RDE).

The nominal performance $\|G\|_{[0,T]} < \gamma$ is achieved if the associated RDE solution exists on $[0, T]$ when integrated backward from $P(T) = F$. The assumption $R(t) < 0$ ensures $R(t)$ is invertible and hence the RDE is well-defined $\forall t \in [0, T]$. Thus, the solution of the RDE exists on $[0, T]$ unless it grows unbounded. The smallest bound on γ is computed using bisection as summarized in Algorithm 2.

Algorithm 2 RDE Bisection Method

1: **Given:** G
2: **Initialize:** ϵ_a, γ_{lb}, γ_{ub} with $\gamma_{lb} \leq \|G\|_{[0,T]} \leq \gamma_{ub}$
3: **while** $\gamma_{ub} - \gamma_{lb} > \epsilon_a$ **do**
4: **Bisect:** $\gamma_{try} = 0.5\,(\gamma_{ub} + \gamma_{lb})$
5: **Integrate RDE:** Solve RDE with γ_{try} backwards in time from $P(T) = F$
6: **Update:** If $P(0) < \infty$ then $\gamma_{ub} = \gamma_{try}$ else $\gamma_{lb} = \gamma_{try}$
7: **end while**
8: **Output:** γ_{ub}, γ_{lb}.

This algorithm is initialized with bounds on the gain such that $\gamma_{lb} \leq \|G\|_{[0,T]} \leq \gamma_{ub}$. The simplest choice for the lower bound is $\gamma_{lb} = \max_{t\in[0,T]} \bar{\sigma}(D_I(t))$. This can be computed (approximately) on a dense time grid. An upper bound can be found (if one is not known) by choosing increasing values of γ until the RDE solution exists on $[0, T]$. The gain $\|G\|_{[0,T]}$ is finite, and hence a finite upper bound will exist. Every bisection step involves integrating the RDE with γ_{try} backward from $P(T) = F$. The bisection continues until the bounds are within the specified tolerance ϵ_a. The RDE solution grows unbounded for each lower bound step, i.e., it only exists on $(t^*, T]$ for some $t^* > 0$. The incomplete RDE solution for any γ_{lb} can be used to construct a specific disturbance d_{lb} such that it achieves the induced gain γ_{lb} [49]. This disturbance d_{lb} provides a verification that the induced gain $\|G\|_{[0,T]}$ is at least γ_{lb}. The RDE exhibits numerical integration issues on a significantly long horizon if G is an unstable LTI system.

3.6 Convergence of LTV Power Iteration

The convergence properties of power iterations for finite-dimensional matrices are summarized first. The eigenvalue power iteration for a square matrix $M \in \mathbb{R}^{n\times n}$ involves iterations of the form: $v^{(i+1)} = Mv^{(i)}/\|Mv^{(i)}\|$. Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of M

ordered from largest magnitude to smallest. If $|\lambda_1| > |\lambda_2|$ and $v^{(1)}$ is chosen randomly then $v^{(i)}$ converges (with probability 1) to the eigenvector associated with λ_1 at a convergence rate of $\frac{|\lambda_2|}{|\lambda_1|}$ (Theorem 5.6 of [50]). However, the iteration may fail to converge if $|\lambda_1| = |\lambda_2|$ but $\lambda_1 \neq \lambda_2$ (Theorem 2 of [51], Section 7.3.1 of [47] and related references).

Similarly, the singular value power iteration for a matrix $M \in \mathbb{R}^{n\times m}$ involves iterations of the form: $v^{(i+1)} = M^\top M v^{(i)}/\|M^\top M v^{(i)}\|$. This can be viewed as an eigenvalue power iteration on the matrix $M^\top M$. However, a slightly stronger convergence result is obtained because $M^\top M$ is Hermitian and positive semidefinite. Specifically, if $v^{(1)}$ is chosen randomly then $v^{(i)}$ converges (with probability 1) to the space spanned by the right singular vectors associated with the largest singular values (Theorem 8.2.1 of [47]). The convergence of the finite-dimensional singular value power iteration does not require the largest singular value to have multiplicity 1 (Theorem 3.8, Chapter 4 of [52]).

Next, consider the power iteration for the LTV systems in Algorithm 1. This involves a forward simulation with G followed by a backward simulation with the adjoint $G^\sim$. This combined action can be written as the composition $G^\sim G : \mathcal{L}_2^{n_d}[0,T] \to \mathcal{L}_2^{n_d}[0,T]$. Algorithm 1 is thus analogous to the singular value power iteration but on an infinite-dimensional space. To make this precise, first note that the operator $G^\sim G$ is a non-negative, self-adjoint, and bounded linear operator on the Hilbert space $\mathcal{L}_2^{n_d}[0,T]$. Theorem 3.1 in [15] provides a convergence proof for power iterations on a bounded linear operator in Hilbert space. The corollary stated below follows from this result.

Corollary 3.6.1. *Suppose $G^\sim G$ has a single, simple dominant eigenvalue λ_1 and corresponding eigenvector ψ_1 with $\|\psi_1\|_{2,[0,T]} = 1$. If $\langle d^{(1)}, \psi_1\rangle \neq 0$ then $\|d^{(i)} - \psi_1\|_{2,[0,T]} \to 0$ as $i \to \infty$.*

Next, we present a stronger convergence result for Algorithm 1 which exploits the structure in singular value power iteration on $\mathcal{L}_2^{n_d}[0,T]$. We do not require the assumption of a single, simple dominant eigenvalue. However, we require that G has no feedthrough. This ensures that $G^\sim G$ is a compact operator (Lemma A.1.1 in Appendix A.1). The eigenvalues of $G^\sim G$ are real and eigenvectors associated with distinct eigenvalues are orthogonal (Theorem 8.11 of [53]). In fact, $G^\sim G$ has an orthonormal set of eigenvectors $\{\psi_k\}_{k=1}^\infty \in \mathcal{L}_2^{n_d}[0,T]$ and eigenvalues $\{\lambda_k\}_{k=1}^\infty \in \mathbb{R}^+ \cup \{0\}$. Moreover,

if $d \in \mathcal{L}_2^{n_d}[0,T]$ then the operation $r = G^{\sim}G(d)$ can be written as:

$$r = G^{\sim}G(d) = \sum_{k=1}^{\infty} \lambda_k \langle d, \psi_k \rangle \psi_k \tag{3.14}$$

Assume that eigenvalues are sorted in descending order and use non-negativity of $G^{\sim}G$ to get $\lambda_k \geq \lambda_{k+1} \geq 0$.

Theorem 3.6.1. *Assume $D_I = 0$ so that $G^{\sim}G$ is compact. Further assume that the dominant eigenvalue has multiplicity m and $d^{(1)}$ satisfies $\langle d^{(1)}, \psi_k \rangle \neq 0$ for some $k \in \{1, \ldots, m\}$. Then $\gamma^{(i+1)} \geq \gamma^{(i)}$, $\forall i \geq 1$ and as $i \to \infty$ we have:*

(a) $d^{(i)} \to$ an element in the span of $\{\psi_1, \ldots, \psi_m\}$

(b) $\gamma^{(i)} \to \|G\|^2_{[0,T]}$

Proof. By Equation (3.14), the first iteration $r^{(1)} = G^{\sim}G(d^{(1)})$ yields $r^{(1)} = \sum_{k=1}^{\infty} \lambda_k \alpha_k \psi_k$ where $\alpha_k := \langle d^{(1)}, \psi_k \rangle$. The gain $\gamma^{(1)}$ can be obtained as:

$$\begin{aligned}
\gamma^{(1)} &= \|r^{(1)}\|_{2,[0,T]} \\
&= \sqrt{\langle r^{(1)}, r^{(1)} \rangle} \\
&= \sqrt{\left\langle \sum_{k=1}^{\infty} \lambda_k \alpha_k \psi_k, \sum_{j=1}^{\infty} \lambda_j \alpha_j \psi_j \right\rangle} \\
&= \sqrt{\sum_{k=1}^{\infty} \sum_{j=1}^{\infty} \lambda_k \lambda_j \alpha_k \alpha_j \langle \psi_k, \psi_j \rangle}
\end{aligned} \tag{3.15}$$

Note $\langle \psi_k, \psi_j \rangle = 0$ for $k \neq j$ and $\langle \psi_k, \psi_j \rangle = 1$ for $k = j$, i.e.

$$\gamma^{(1)} = \sqrt{\sum_{k=1}^{\infty} (\lambda_k \alpha_k)^2} \tag{3.16}$$

The next disturbance $d^{(2)}$ can be obtained by:

$$d^{(2)} = \frac{r^{(1)}}{\gamma^{(1)}} = \frac{1}{\gamma^{(1)}} \sum_{k=1}^{\infty} \lambda_k \alpha_k \psi_k \tag{3.17}$$

Another application of $G^{\sim}G$ yields $r^{(2)} = G^{\sim}G(d^{(2)})$, i.e.

$$\begin{aligned} r^{(2)} &= \sum_{k=1}^{\infty} \lambda_k \langle d^{(2)}, \psi_k \rangle \psi_k \\ &= \frac{1}{\gamma^{(1)}} \sum_{k=1}^{\infty} \lambda_k \Big\langle \sum_{j=1}^{\infty} \lambda_j \alpha_j \psi_j, \psi_k \Big\rangle \psi_k \\ &= \frac{1}{\gamma^{(1)}} \sum_{k=1}^{\infty} \sum_{j=1}^{\infty} \lambda_k \lambda_j \alpha_j \langle \psi_j, \psi_k \rangle \psi_k \\ &= \frac{1}{\gamma^{(1)}} \sum_{k=1}^{\infty} \lambda_k^2 \alpha_k \psi_k \end{aligned} \tag{3.18}$$

Define the constant $c^{(i)} := 1/(\gamma^{(i-1)} \dots \gamma^{(2)} \gamma^{(1)})$ and repeat above steps to obtain the following expressions for the i^{th} iteration as:

$$r^{(i)} = c^{(i)} \sum_{k=1}^{\infty} \lambda_k^i \alpha_k \psi_k, \qquad \gamma^{(i)} = c^{(i)} \sqrt{\sum_{k=1}^{\infty} (\lambda_k^i \alpha_k)^2} \tag{3.19}$$

The unit-norm disturbance at the i^{th} iteration is given by:

$$d^{(i)} = \frac{r^{(i-1)}}{\gamma^{(i-1)}} = c^{(i)} \sum_{k=1}^{\infty} \lambda_k^{i-1} \alpha_k \psi_k \tag{3.20}$$

Since the dominant eigenvalue λ_1 has multiplicity m we have:

$$d^{(i)} = c^{(i)} \lambda_1^{i-1} \left[\sum_{k=1}^{m} \alpha_k \psi_k + \sum_{k=m+1}^{\infty} \left(\frac{\lambda_k}{\lambda_1} \right)^{i-1} \alpha_k \psi_k \right] \tag{3.21}$$

As $i \to \infty$ the second sum in the above term $\to 0$ due to $\frac{\lambda_k}{\lambda_1} < 1$. Moreover, the assumption $\alpha_k \neq 0$ for some $k = 1, 2, \dots, m$ ensures that $d^{(i)}$ converges to an element in the subspace spanned by dominant eigenvectors. This assumption holds with probability 1 if $d^{(1)}$ is chosen randomly. Note that the convergence rate depends on ratio $\frac{\lambda_{m+1}}{\lambda_1}$.

Next, use the expression for $\gamma^{(i)}$ to write:

$$\gamma^{(i)} = c^{(i)}\,\lambda_1^i\sqrt{\sum_{k=1}^{m}\alpha_k^2 + \sum_{k=m+1}^{\infty}\left(\frac{\lambda_k}{\lambda_1}\right)^{2i}\alpha_k^2} \tag{3.22}$$

As $i \to \infty$, the second term inside the square root $\to 0$. Moreover, the disturbance is normalized at each iterate, i.e. $\|d^{(i)}\|_{2,[0,T]} = 1$. It follows from Equation (3.21) that:

$$c^{(i)}\,\lambda_1^{i-1}\sqrt{\sum_{k=1}^{m}\alpha_k^2} \to 1 \text{ as } i \to \infty \tag{3.23}$$

Combine (3.23) and (3.22) to conclude $\gamma^{(i)} \to \lambda_1$. Theorem 6.10 of [54] implies that induced norm of G is equal to $\sqrt{\lambda_1(G^{\sim}G)}$. Thus, $\gamma^{(i)} \to \|G\|^2_{[0,T]}$ as $i \to \infty$.
Finally, use Equation (3.19) to write the ratio of gain for two subsequent iterations as:

$$\frac{\gamma^{(i+1)}}{\gamma^{(i)}} = \frac{c^{(i+1)}\sqrt{\sum_{k=1}^{\infty}(\lambda_k^{i+1}\alpha_k)^2}}{c^{(i)}\sqrt{\sum_{k=1}^{\infty}(\lambda_k^{i}\alpha_k)^2}} = \frac{\sqrt{\sum_{k=1}^{\infty}(\lambda_k^{i+1}\alpha_k)^2}}{c^{(i)}\sum_{k=1}^{\infty}(\lambda_k^{i}\alpha_k)^2} \tag{3.24}$$

Define $f_k := \lambda_k^{i+1}\alpha_k$ and $g_k := \lambda_k^{i-1}\alpha_k$ to obtain:

$$c^{(i)}\sum_{k=1}^{\infty}(\lambda_k^i\alpha_k)^2 = c^{(i)}\sum_{k=1}^{\infty} f_k\,g_k \le c^{(i)}\|f\|_2\|g\|_2 \quad \text{(Hölder's Inequality)} \tag{3.25}$$

It follows from Equation (3.20) that $\|d^{(i)}\|_{2,[0,T]} = c^{(i)}\|g\|_2$ and hence $c^{(i)}\|g\|_2 = 1$ as the disturbances have unit norm. It follows from (3.24) and (3.25) that $\gamma^{(i+1)} \ge \gamma^{(i)}$, $\forall i \ge 1$. □

Remark 3.6.1. *For LTI systems, compactness of G (and hence $G^{\sim}G$) can be used to obtain a finite-dimensional eigenvalue problem, as in the periodic sampled-data systems frequency-response literature [40], [55],[42].*

Theorem 3.6.1 added the assumption $D_I = 0$ to ensure compactness of $G^{\sim}G$. This yields a stronger convergence result than obtained in [15] for the eigenvalue power iteration. Specifically, the power iteration in Algorithm 1 converges even if the dominant eigenvalue is repeated. If $D_I \neq 0$ then the power iteration will still converge by result

in [15] if $\lambda_1 > \lambda_2$. However, this condition cannot be verified in practice as the eigenvalues are unknown. Finally, Theorem 3.6.1 also guarantees that the iteration will be non-decreasing. This is a useful diagnostic condition as $\gamma^{(i+1)} < \gamma^{(i)}$ indicates errors in the numerical integration. Algorithm 1 can detect this anomaly, terminate the iteration, and warn the user to reduce the integration step size.

3.7 Computational Issues

The computational time for the bisection method can be approximated by the number of bisections N_{RDE} multiplied by the computational time for one RDE integration T_{RDE}.[†] If the algorithm starts with a gap $\gamma_g := \gamma_{ub} - \gamma_{lb}$ then it takes at least $N_{RDE} = \log_2(\gamma_g/\epsilon_a)$ to achieve an absolute tolerance of ϵ_a. Additional RDE integrations are required if an upper bound is not known at the start of the algorithm. Parallel computing resources can be exploited to integrate RDE on a grid of γ. However, the single RDE integration time depends on many factors, including the time horizon T, system order n_x, and integration solver tolerances. The cost of the RDE grows roughly linearly with T. The RDE is an $n_x \times n_x$ matrix differential equation and hence this integration requires solving $n_x(n_x+1)/2 \approx \mathcal{O}(n_x^2)$ scalar, nonlinear differential equations.[‡] Algorithm 2 guarantees on exit that $\|G\|_{[0,T]}$ has been computed within an absolute tolerance ϵ_a.

The power iteration method has complementary properties. The computational time can be approximated as the number of iterations N_{PI} multiplied by the time required for one iteration T_{PI}. A single power iteration requires the integration of G followed by the integration of $G^{\sim}$. Both G and $G^{\sim}$ are LTV systems of order n_x and hence, T_{PI} scales as $\mathcal{O}(n_x)$. It also scales linearly with the horizon T. The primary advantage of the power iteration is that T_{PI} is typically significantly less than T_{RDE}. We expect the ratio T_{RDE}/T_{PI} to grow with the state dimension as $\mathcal{O}(n_x)$. Moreover, the power iteration is guaranteed to converge by Theorem 3.6.1. However, the primary drawback of the power iteration is that convergence depends on the ratio λ_2/λ_1. This can be arbitrarily slow

[†]The RDE solution may grow unbounded for lower bound step and exist only on $(t^*, T]$ for some $t^* > 0$. It takes longer to integrate over the entire horizon ($t^* = 0$) than if the solution grows unbounded for some $t^* > 0$. A more precise estimate of the computation time would account for the dependence of T_{RDE} on the choice of γ_{try} at each bisection step.

[‡]It is possible to solve the RDE by instead integrating a set of $2n_x \times n_x$ linear differential equations with a related Hamiltonian. This form does not seem to offer computational advantages.

if this ratio is $1-\kappa$ for some $\kappa \ll 1$, i.e., N_{PI} can be arbitrarily large to achieve the desired tolerance ϵ_a. Moreover, the power iteration only provides lower bounds on the induced gain. Algorithm 1 terminates if the iteration fails to make significant progress or if it reaches a maximum number of iterations. Algorithm 1 provides no guarantee on the accuracy between the returned γ_o and the actual value $\|G\|_{[0,T]}$ because the power iteration does not compute this gap. The next two examples illustrate these issues. All examples in this chapter are performed using MATLAB running on a desktop computer with a 3 GHz Intel core i7 processor and 16 GB RAM.

Example 3.7.1. *We randomly sampled five representative SISO LTI systems for each model order $n_x = 1, 10, 20, \ldots, 200$. Computation times were recorded for each model to integrate a single RDE and perform one power iteration step (G followed by $G^\sim$) on horizon $T = 15$ sec. Each RDE integration was performed with a $\gamma > \|G\|_{[0,T]}$ so that the solution existed on $[0, T]$. Let $T_{RDE}(n_x)$ and $T_{PI}(n_x)$ denote the computation times averaged over the five random models of state dimension n_x. Figure 3.1 shows the ratio $T_{RDE}(n_x)/T_{PI}(n_x)$ along with a linear fit. As noted earlier, the costs of the RDE and power iteration scale as $\mathcal{O}(n_x^2)$ and $\mathcal{O}(n_x)$, respectively. Thus, we expect the ratio to grow as $\beta_0 + \beta_1 n_x$ for some constant β_0 and β_1. The offset and slope of the linear fit are $\beta_0 = 7.26$ and $\beta_1 = 0.07$. This indicates that the computational time of a single RDE integration is roughly equal to $\beta_0 + \beta_1 n_x$ power iterations.*

Example 3.7.2. *Let G_1 be an LTI system given by state matrices:*

$$A = \left[\begin{smallmatrix} -0.1 & 0.4 \\ -0.5 & 0 \end{smallmatrix}\right],\ B = \left[\begin{smallmatrix} 2 \\ 0 \end{smallmatrix}\right],\ C_I = \left[\begin{smallmatrix} 0 & 1 \end{smallmatrix}\right],\ D_I = 0$$

The power iteration for induced $\mathcal{L}_2$ gain of G_1 (for $T = 10$ sec) using $\epsilon_a = 5 \times 10^{-3}$ converged to the gain of 7.159 in just four iterations. Next define the MIMO system G_2 such that $G_2 := \left[\begin{smallmatrix} G_1 & 0 \\ 0 & 0.95G_1 \end{smallmatrix}\right]$. We have $\|G_2\|_{[0,T]} = \|G_1\|_{[0,T]}$ by construction. However, the power iteration for G_2 converged to 7.130 in 14 iterations. The power iteration on G_1 and G_2 took 0.07 sec and 0.27 sec, respectively. Figure 3.2 shows the progress of the power iteration on these two systems. The iteration has slow convergence for G_2 because the eigenvalues of $G_2^\sim G_2$ are not well separated. For comparison, the RDE bisection returned the same bounds $[7.157, 7.161]$ for both G_1 and G_2. However, this took 11 bisections and 3.7 seconds of computation time.

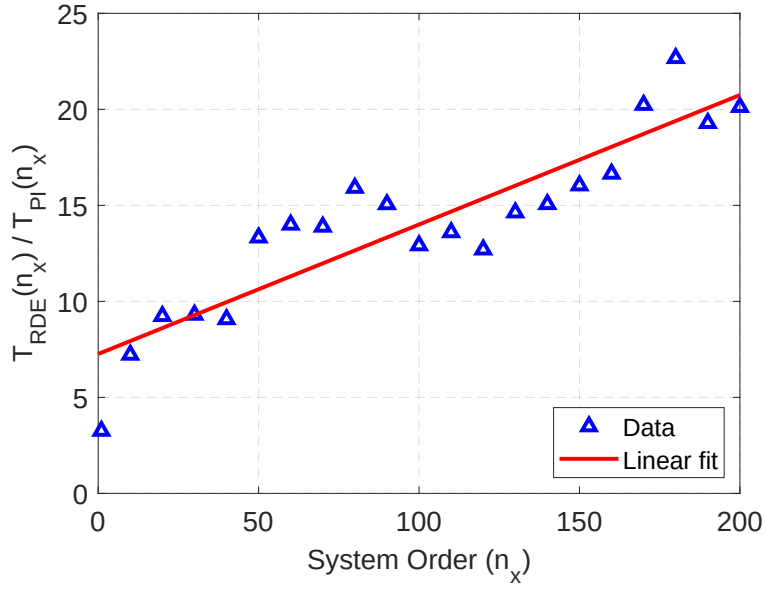

Figure 3.1: Ratio of Average Computational Time vs System Order

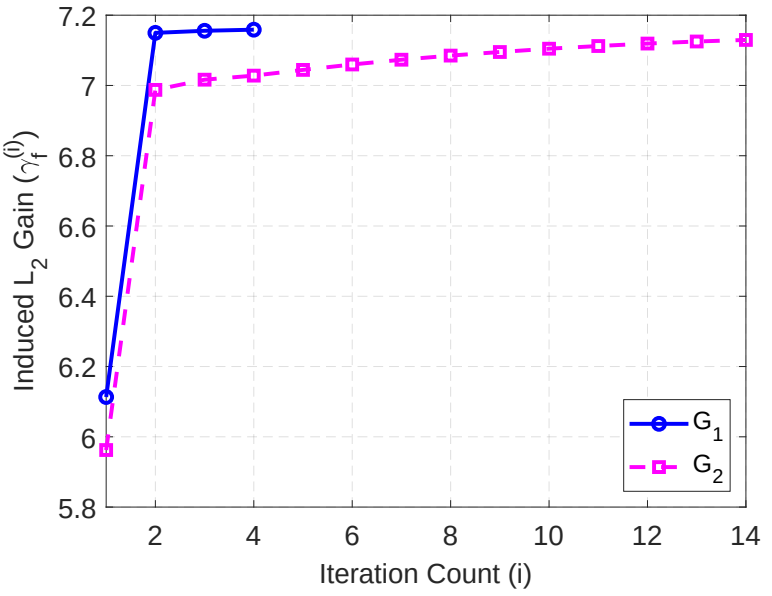

Figure 3.2: Power Iteration Progress for System G_1 and G_2

3.8 Proposed Algorithm

3.8.1 Combined Algorithm

This section presents an algorithm that combines the benefits of the power iteration and RDE bisection. The specific numerical steps are outlined in Algorithm 3. This algorithm is initialized with a candidate disturbance $d^{(1)}$, which is chosen randomly and normalized to have size 1. The maximum number of power iterations N is set to some high value, such as 50. The initial upper bound $\gamma_{ub} = \infty$ and lower bound $\gamma_{lb} = 0$ are fixed. The absolute tolerance ϵ_a is specified. In addition, relative tolerance ϵ_r can also be used.

Algorithm 3 Combined Algorithm

1: **Given:** G, $G^{\sim}$
2: **Initialize:** $d^{(1)}$ with $\|d^{(1)}\|_{2,[0,T]} = 1$, ϵ_a, N, γ_{lb}, γ_{ub}, iteration count $i = 1$.
3: **while** $\gamma_{ub} - \gamma_{lb} > \epsilon_a$ **do**
4: **Run Power Iterations:** $[\gamma_o, d_o]$ = Algorithm 1 (G, $G^{\sim}$, $d^{(i)}$, N, $\epsilon_a/5$)
5: **Integrate RDE:** Solve RDE for $\gamma_{try} = \gamma_o + \epsilon_a$.
6: **Update:** If $P(0) < \infty$ then $\gamma_{ub} = \gamma_{try}$, $\gamma_{lb} = \gamma_0$, $d_{lb} = d_o$ else $\gamma_{lb} = \gamma_{try}$, and construct a unit norm disturbance d_{lb} using method in [49] and set $d^{(i+1)} = d_{lb}$. Increment the count $i = i + 1$.
7: **end while**
8: **Output:** γ_{lb}, γ_{ub}, d_{lb}

The first step in the iterative loop is to perform power iterations with tight numerical tolerance, i.e., $\epsilon_a/5$. This gives the gain γ_o and corresponding disturbance d_o. The next step is to perform RDE integration for $\gamma_{try} = \gamma_o + \epsilon_a$ from the respective boundary condition. Notice that this γ_{try} is just high enough to satisfy the stopping criteria for the loop. Moreover, the tight tolerance used in power iteration increases the likelihood for the existence of the RDE solution for γ_{try}. If the RDE solution exists then γ_{ub}, γ_{lb}, d_{lb} are set to γ_{try}, γ_o, d_o, respectively and we are done as no further iterations required. If RDE solution is incomplete then γ_{try} is a better lower bound γ_{lb} and disturbance d_{lb} can be constructed from non-convergent RDE solution [49]. This disturbance is set to $d^{(i+1)}$ and is used as a starting disturbance in the next iteration. Finally, the iteration count i is incremented. Algorithm 3 can be modified to switch over to the RDE bisection method if it does not make significant progress within some small number of

RDE integration calls. This ensures computational complexity that is no worse than Algorithm 2. In practice, the combined algorithm: (i) is faster than the RDE bisection method and (ii) terminates with guaranteed upper and lower bounds on the induced gain within a specified numerical tolerance. Thus, it merges the benefits of RDE bisection and power iteration.

3.8.2 Special Case: $\mathcal{L}_2$-to-Euclidean Gain

If $n_I = 0$ then the induced norm defined in Equation (3.5) is a finite horizon induced $\mathcal{L}_2$-to-Euclidean gain of G which is denoted by $\|G\|_{E,[0,T]}$. Note that $G : \mathcal{L}_2^{n_d}[0,T] \to \mathbb{R}^{n_E}$ is a finite rank operator. The next theorem presents a simpler condition to compute $\|G\|_{E,[0,\tau]}$ for any intermediate horizon $\tau \in [0,T]$. This simpler condition only requires a forward integration of the LDE 3.6. In comparison, the Algorithm 1, 2 or 3 have to be run for each $\tau \in [0,T]$ to compute $\|G\|_{E,[0,\tau]}$.

Theorem 3.8.1. *Let $Y(t) := C_E(t)X(t)C_E(t)^\top$, $\forall t \in [0,T]$ be the output controllability Gramian with largest eigenvalue denoted as $\lambda_1(Y(t))$. The finite horizon induced $\mathcal{L}_2$-to-Euclidean gain of G for any horizon $\tau \in [0,T]$ is given by $\|G\|_{E,[0,\tau]} = \sqrt{\lambda_1(Y(\tau))}$. Moreover, a unit-norm, worst-case disturbance $d_{wc}(t)$ for $t \in [0,\tau]$ is given by:*

$$d_{wc}(t) = B(t)^\top \Phi(\tau,t)^\top C_E(\tau)^\top \frac{v_1}{\sqrt{\lambda_1(Y(\tau))}} \tag{3.26}$$

where v_1 is a unit-norm eigenvector associated with the maximum eigenvalue $\lambda_1(Y(\tau))$.

Proof. The proof is similar to existing results on Gramian-based minimum energy control (Theorem 1, Section 22 of [16]). The results in [16] provide a condition for transferring the state from a given initial state $x(0) = x_0$ to a final state $x(T) = x_T$ using the least control energy. These results can be used to determine the input of the unit norm that maximizes the Euclidean norm of the final state starting from zero initial conditions. The proof is given below for completeness.

Let $G : \mathcal{L}_2^{n_d}[0,\tau] \to \mathbb{R}^{n_E}$ be a given bounded linear operator with the adjoint $G^\sim : \mathbb{R}^{n_E} \to \mathcal{L}_2^{n_d}[0,\tau]$ for any $\tau \in [0,T]$. The notation $\mathcal{R}(G)$ and $\mathcal{N}(G)$ are used to denote the range and null space of G respectively. We know that G is a finite rank operator because the $\mathcal{R}(G)$ is finite-dimensional. Moreover, every bounded finite rank operator is

compact (Theorem $8.1-4$ of [56]). Assume G is output-controllable, thus $\mathcal{R}(G) \equiv \mathbb{R}^{n_E}$ which is closed. This implies $\mathcal{R}(G^{\sim})$ is also closed (Chapter 4, Theorem 5.13 in [57]). In this case we have:

$$\mathcal{R}(G^{\sim}) = \mathcal{N}(G)^{\perp} \tag{3.27}$$

where $\mathcal{N}(G)^{\perp}$ denote an orthogonal complement of $\mathcal{N}(G)$. Note that $\mathcal{N}(G)$ is a closed linear subspace of the Hilbert space $\mathcal{L}_2^{n_d}[0,\tau]$ (Theorem 1.18 of [58]). Thus, we can decompose the Hilbert space $\mathcal{L}_2^{n_d}[0,\tau]$ as $\mathcal{N}(G)^{\perp} \oplus \mathcal{N}(G)$ (Section 3.4 of [59], Section 5.1 of [57]). This implies that any $d \in \mathcal{L}_2^{n_d}[0,\tau]$ can be decomposed as $d = d_1 + d_2$, where $d_1 \in \mathcal{N}(G)^{\perp}$, $d_2 \in \mathcal{N}(G)$ and the inner product $\langle d_1, d_2 \rangle = 0$. Moreover, since $\mathcal{N}(G)$ is a null space of G, we have $G(d_2) = 0$. Use the the linearity property of an operator i.e. $G(d_1 + d_2) = G(d_1) + G(d_2)$ to rewrite the square of the induced $\mathcal{L}_2$-to-Euclidean cost as follows:

$$J(d) := \frac{\|e(\tau)\|_2^2}{\|d\|_{2,[0,\tau]}^2} = \frac{e(\tau)^{\top} e(\tau)}{\langle d, d \rangle} = \frac{G(d)^{\top} G(d)}{\langle d, d \rangle} = \frac{G(d_1)^{\top} G(d_1)}{\langle d_1, d_1 \rangle + \langle d_2, d_2 \rangle} \leq \frac{G(d_1)^{\top} G(d_1)}{\langle d_1, d_1 \rangle} \tag{3.28}$$

This means $J(d) \leq J(d_1)$. Thus, the optimization over infinite dimensional Hilbert space $\mathcal{L}_2^{n_d}[0,\tau]$ is equivalent to the optimization over $\mathcal{N}(G)^{\perp}$. Use Equation (3.27) to rewrite the problem as:

$$\begin{aligned} \|G\|_{E,[0,\tau]}^2 &= \sup_{d \in \mathcal{L}_2^{n_d}[0,\tau]} J(d) \\ &= \sup_{d \in \mathcal{N}(G)^{\perp}} J(d) = \sup_{d \in \mathcal{R}(G^{\sim})} J(d) \end{aligned} \tag{3.29}$$

Define $w := e(\tau)$ to rewrite the optimization problem over the finite dimensional space $\mathbb{R}^{n_E}$ as:

$$\|G\|_{E,[0,\tau]}^2 = \sup_{w \in \mathbb{R}^{n_E}} \frac{\langle GG^{\sim}(w), GG^{\sim}(w) \rangle}{\langle G^{\sim}(w), G^{\sim}(w) \rangle} \tag{3.30}$$

Using $G^{\sim}(w) = B(t)^{\top}\Phi(\tau,t)^{\top}C_E(\tau)^{\top}w$, the composition $GG^{\sim}(w)$ can be written as:

$$GG^{\sim}(w) = C_E(\tau)X(\tau)C_E(\tau)^{\top}\,w = Y(\tau)\,w \tag{3.31}$$

where $X(\tau) = \int_0^{\tau}\Phi(\tau,s)B(s)\,B(s)^{\top}\Phi(\tau,s)^{\top}ds$ is a solution to LDE (3.6) at time τ. The inner product in the denominator of Equation (3.30) can be written as:

$$\langle G^{\sim}(w), G^{\sim}(w)\rangle = \langle w, GG^{\sim}(w)\rangle = w^{\top}\,Y(\tau)\,w \tag{3.32}$$

Thus, the optimal cost from Equation (3.30) can be written as:

$$\|G\|^2_{E,[0,\tau]} = \sup_{w\in\mathbb{R}^{n_E}} \frac{w^{\top}Y(\tau)^{\top}Y(\tau)w}{w^{\top}Y(\tau)w} \tag{3.33}$$

where $Y(\tau) \in \mathbb{S}^{n_E}$ is diagonalizable and system is assumed to be output controllable i.e. $Y(\tau) > 0$. The eigenvalue decomposition of $Y(\tau)$ can be obtained as $V\Lambda V^{\top}$ where $\Lambda > 0$ is a diagonal matrix of eigenvalues and V is a unitary matrix whose columns are orthonormal eigenvectors v_i, for $i = 1, 2, \ldots, n_E$. Substitute this relation in Equation (3.33) and use $V^{\top}V = I$ to obtain:

$$\|G\|^2_{E,[0,\tau]} = \sup_{w\in\mathbb{R}^{n_E}} \frac{w^{\top}V\Lambda\Lambda V^{\top}w}{w^{\top}V\Lambda V^{\top}w} \tag{3.34}$$

Define $u := \Lambda^{\frac{1}{2}}V^{\top}w$ to rewrite the optimization problem as:

$$\|G\|^2_{E,[0,\tau]} = \sup_{u\in\mathbb{R}^{n_E}} \frac{u^{\top}\Lambda u}{u^{\top}u} \tag{3.35}$$

This is an eigenvalue problem and the cost is maximized with the optimal solution $u^* = \begin{bmatrix}1 & 0 & \ldots & 0\end{bmatrix}^{\top}$ where u^* is an eigenvector of the diagonal matrix Λ corresponding to the maximum eigenvalue λ_1, which is also the maximum eigenvalue of the matrix $Y(\tau)$. Thus $\|G\|^2_{E,[0,\tau]} = \lambda_1(Y(\tau))$. Taking square root implies $\|G\|_{E,[0,\tau]} = \sqrt{\lambda_1(Y(\tau))}$. The worst-case disturbance in $\mathcal{R}(G^{\sim})$ is obtained by simulating the adjoint dynamics with $w^* = V\Lambda^{-\frac{1}{2}}u^* = \frac{v_1}{\sqrt{\lambda_1}}$, i.e. $d_{wc}(t) = B(t)^{\top}\Phi(\tau,t)^{\top}C_E(\tau)^{\top}\frac{v_1}{\sqrt{\lambda_1(Y(\tau))}}$. □

Theorem 3.8.1 shows that related worst-case disturbance $d_{wc}(t)$ for $t \in [0,\tau]$ can

also be computed by simulating the adjoint dynamics in the direction of eigenvector v_1. The similar result related to the square root of the maximum eigenvalue of the output controllability Gramian appeared in [60, 61] for induced $\mathcal{L}_2$-to-$\mathcal{L}_\infty$ gain and in [62, 63, 64] for generalized H_2 norm of LTI systems on infinite horizons. Since *trace* in Equation 3.9 is sum of eigenvalues, we can obtain the following corollary of norm equivalence when system has scalar output.

Corollary 3.8.1. *For system G with scalar outputs, we have $\|G\|_{H_2,[0,T]} = \|G\|_{E,[0,T]}$.*

Similar result appeared in Thereon 2.2 of [65] for LTI systems on infinite horizons. Moreover, it is important to note that for scalar output systems, the fastest way to compute $\|G\|_{E,[0,T]}$ is by means of power iteration, which converges in 3 iterations of Algorithm 1 due to the ratio $\frac{\lambda_2}{\lambda_1} = 0$.

3.9 Numerical Examples

3.9.1 Computational Study: LTI Systems

Consider SISO LTI systems of increasing system order as discussed in Example 3.7.1. Each system is normalized so that the infinite horizon H_∞ norm is one. Algorithms 1, 2 and 3 are run to compute the induced $\mathcal{L}_2$ gain within $\epsilon_a = 0.01$ on a horizon $T = 15$ seconds. The power iteration only computes lower bounds. Figure 3.3 shows the comparison of the bound computed using Algorithm 1 and 2 for different models (sorted in increasing order of gain). It indicates that the RDE bisection method lower bound (blue circles) is always within the desired numerical tolerance ϵ_a (black line) from the upper bound (red squares). However, the power iteration lower bound (green diamonds) may or may not be within the bisection tolerance. Figure 3.4 shows the computation times (averaged over 5 random models) versus state dimension n_x. The solid line denotes the linear fit to the respective data. The proposed method is faster than RDE bisection and provides guaranteed upper/lower bounds (in comparison to the power iteration).

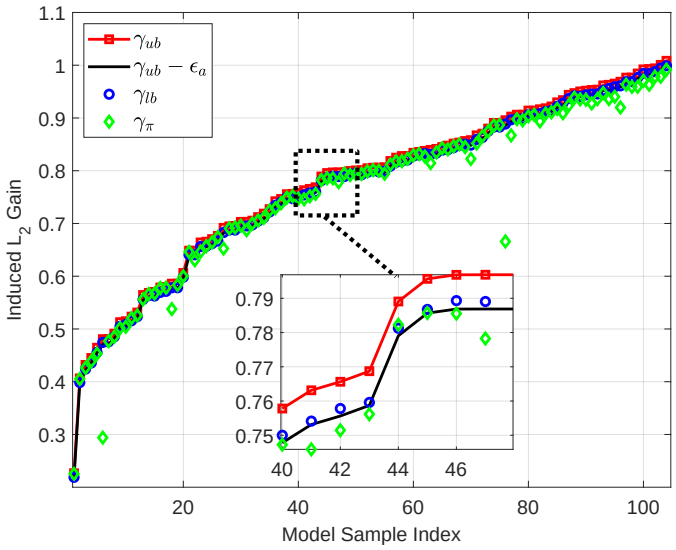

Figure 3.3: Comparison of Gain Computed Using Algorithm 1 and 2

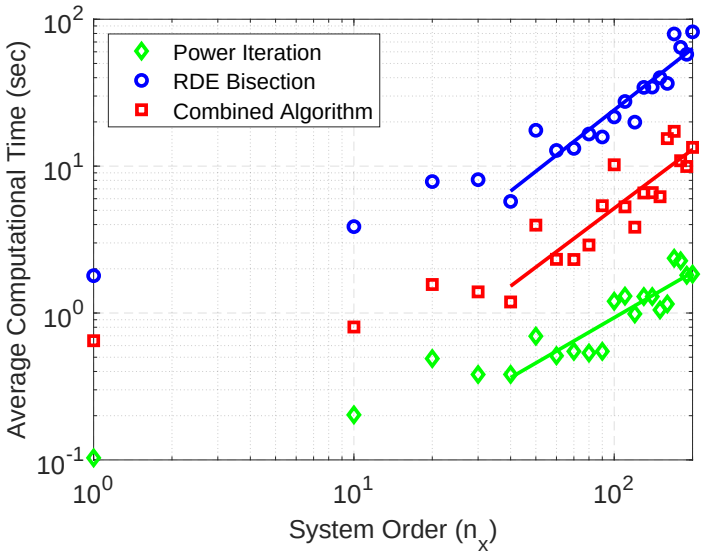

Figure 3.4: Average Computational Time vs System Order

3.9.2 Comparison of H_2 and $\mathcal{L}_2$-to-Euclidean Norm

Consider an LTI system G given by the state-space matrices:

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -0.2 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ C_E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ D_E = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Let initial condition $x(0) = 0$, horizon $T = 1$ second and $\|d\|_{2,[0,T]} = 2$. The $\mathcal{L}_2$-to-Euclidean norm can be computed as 0.878. The worst-case disturbance d_{wc} can also be computed using the adjoint simulation as discussed in Theorem 3.8.1. This disturbance can be scaled so that $\|d_{wc}\|_{2,[0,T]} = 2$. Figure 3.5 shows the phase portrait visualization. The red shaded disk centered at the origin (initial condition) and radius $0.878 \times 2 = 1.754$ represents the Euclidean norm ball at the final time T that can be reached by $\mathcal{L}_2[0,T]$ disturbances of at most size 2. The blue trajectory starting from the origin represents the worst-case disturbance response. This disturbance pushes the final state of the system farthest away from the origin. If d is a white noise input with unit covariance then the H_2 norm can be computed as 0.921. The output covariance matrix at the final time T is obtained as follows:

$$\mathcal{E}[e(T)e(T)^T] = \begin{bmatrix} 0.236 & 0.291 \\ 0.291 & 0.612 \end{bmatrix} \tag{3.36}$$

Let σ denote the standard deviation of the output at the final time. Then the 3σ magenta ellipse representing 99.7% confidence for the final state is shown in Figure 3.5 with origin as a center and covariance reported above. To validate this result, we consider 100 Monte-Carlo simulations with white noise input d on the horizon of 1 second. For each of these simulations, the MATLAB command `randn` is used to sample white noise input from Gaussian distribution at the sample rate of $T_s = 0.01$ second. This input is scaled with the factor of $\frac{1}{\sqrt{T_s}}$ to have the unit variance. The resulting final position of the state is shown as green dots in Figure 3.5. It can be verified that all of these green dots fall within the magenta ellipse.

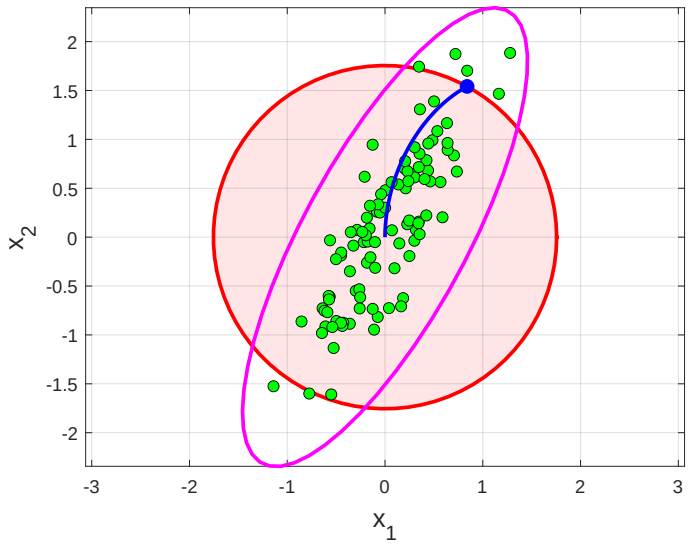

Figure 3.5: Interpretation of H_2 and $\mathcal{L}_2$-to-Euclidean Norm

3.9.3 LTV Example

Consider the Example 4.1 in [39]. Let the LTV system be given by the following state-space matrices:

$$A(t) = \begin{bmatrix} -1 + sin(t) & 1 \\ 0 & -4 \end{bmatrix}, \; B = C_I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; D_I = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The induced $\mathcal{L}_2$ gains are computed on the horizon of $T = 10$ seconds using Algorithm 1, 2 and 3 with $\epsilon_a = 0.01$. Algorithm 1 took 7 power iterations to reach the γ_o of 1.782 in total 0.45 seconds. Algorithm 2 took 10 bisections to compute the bounds of $[1.799, 1.809]$ in 4.75 seconds. Algorithm 3 took only two iterations to compute the same bound $[1.799, 1.809]$ in 1.63 seconds. The unit-norm worst-case disturbances are shown in Figure 3.6 which matches with Figure 1 in [39].

3.9.4 Nonlinear System: Two-Link Robot

Let us revisit an example of a two-link robot as discussed in Section 2.6. We are interested in assessing the performance of the closed-loop LTV system when an LQR

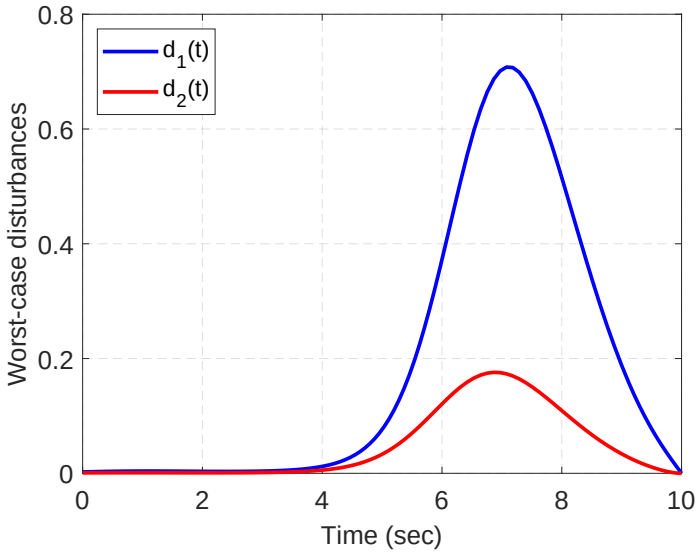

Figure 3.6: Worst-case Disturbances

controller is wrapped around the linearized plant.

First, the induced $\mathcal{L}_2$ gain from disturbance input d to all the linearized states $e_I = x$ is computed using Algorithm 2 and 3 for comparison. The absolute tolerance of $\epsilon_a = 5 \times 10^{-3}$ was selected. Algorithm 2 took 11.8 seconds to compute the lower and upper bounds of 0.098 and 0.102. It took 8 RDE bisections to achieve this accuracy. Algorithm 3 returned the lower and upper bound of 0.096 and 0.101 in 5.4 seconds. It took 1 full iteration of proposed algorithm (i.e. 6 power iterations and 1 RDE integration). Next, consider $\mathcal{L}_2$-to-Euclidean gain analysis using the condition presented in Theorem 3.8.1. This analysis is performed from disturbance input d to the performance output $e_E = \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right]$. It took 2.3 seconds to compute the Euclidean norm bound along the entire trajectory for disturbance with $\|d\|_{2,[0,T]} = 5$. Using the trim values of θ_1, θ_2 and given robot parameters, the robot tip position in the Cartesian coordinate was computed. Figure 3.7 shows results of 50 nonlinear Monte-Carlo simulations using the same sampled disturbances as in [17]. Figure 3.7 shows the worst-case cyan bound at 100 equally spaced points around a nominal trajectory.

Such preliminary analysis performed on an LTV system provides primary guarantees on performance. The worst-case disturbance can also be computed using adjoint

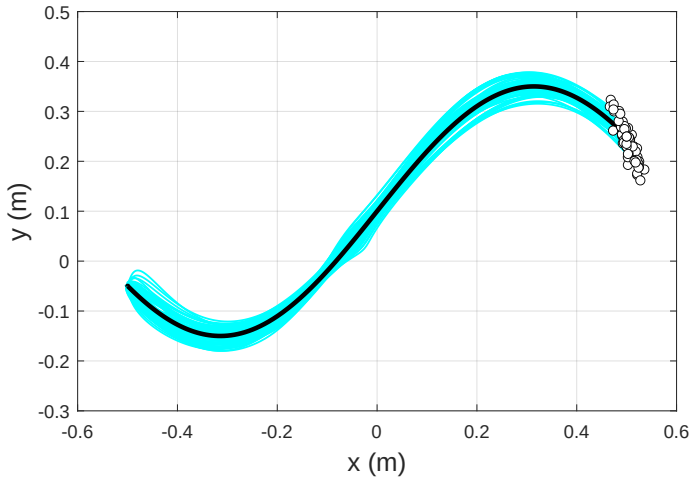

Figure 3.7: Nonlinear Monte-Carlo Simulations with $\|d\|_{2,[0,T]} = 5$

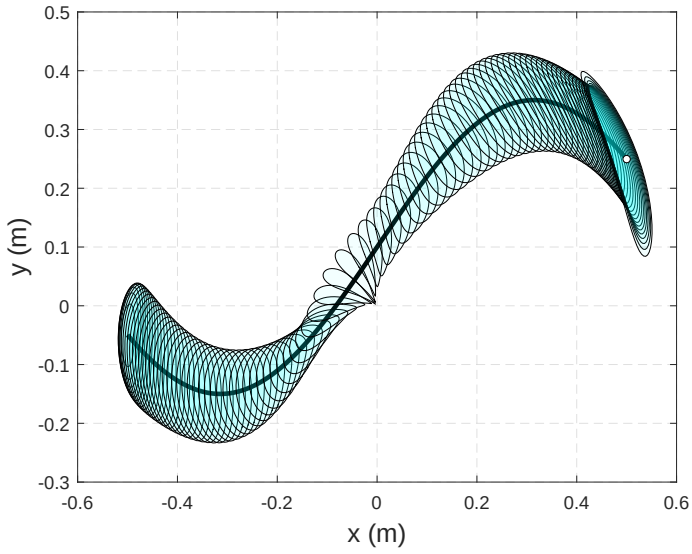

Figure 3.8: Euclidean Norm Bound Along the Trajectory

simulation for any given time horizon. The impact of worst-case disturbance can be further studied in high-fidelity nonlinear simulation. Moreover, this disturbance can be used to initialize the power iterations for the nonlinear model as in [8].

3.10 Summary

This chapter developed an efficient algorithm to compute induced norms of finite-horizon (nominal) LTV systems [13]. Numerical examples were presented to demonstrate the effectiveness of the proposed approach. The next chapter extends this work to the uncertain LTV systems.

Chapter 4

Robust Performance

4.1 Introduction

This chapter focuses on the robust performance of the uncertain LTV system. Worst-case induced gain is defined in Section 4.3 as a metric of interest; however, a similar approach can also be used for the worst-case H_2 norm (see related work in [66, 67] for LTI systems on the infinite horizon). The input-output behavior of uncertainty is described in terms of the Integral Quadratic Constraints (IQCs), and an algorithm presented in [10] is used to compute an upper bound on the worst-case gain (Section 4.4). The worst-case gain lower bound computation approaches are reviewed in Section 4.5 and a new power iteration algorithm is presented. The worst-case uncertainty construction problem is formulated in Section 8.2.2, and key technical challenges are discussed. For uncertain parametric uncertainties, the first-order sensitivity analysis approach is presented in Section 4.6. Such analysis provides a combination of worst-case parameters that can be studied further in high-fidelity nonlinear simulation.

4.2 Related Work

The robustness analysis step uses the IQC framework introduced in [68, 69]. This framework has been extended in [17] and [10] to assess the robustness of the uncertain LTV systems on finite horizons. This work is revisited and used in this chapter. The Gramian based computational conditions (from Section 3.8.2) are extended in related work [31] to

compute worst-case disturbances for sample-based robustness analysis of LTV approximations of uncertain nonlinear systems. Other closely related works are [70, 71, 72, 73]. These include applications of finite horizon LTV analysis to launch vehicle [70, 72] and aircraft autoland systems under crosswinds [73]. The closely related work in power iteration lower bound algorithm are [6, 20, 18, 74, 8]. The work in [6, 20, 18] considers power iterations for structured singular value μ-lower bound computation for the uncertain LTI system and the work in [74, 8] discusses the worst-case induced $\mathcal{L}_2$ gain computation for the uncertain nonlinear systems. Related works in sensitivity analysis are [75, 28, 76].

4.3 Uncertain Linear Time-Varying (LTV) Systems

An uncertain, time-varying system $\mathcal{F}_u(N, \Delta)$ is shown in Figure 4.1. This consists of an interconnection of a known finite horizon LTV system N and a perturbation Δ. This perturbation represents block-structured uncertainties and/or nonlinearities. The term

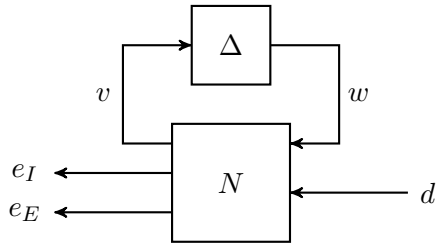

Figure 4.1: Uncertain System Interconnection $\mathcal{F}_u(N, \Delta)$

"uncertainty" is used for simplicity when referring to Δ. It is assumed throughout that the interconnection $\mathcal{F}_u(N, \Delta)$ is well-posed. A formal definition for well-posedness is given in [21, 69]. The LTV system N is described by the following state-space model:

$$\begin{bmatrix} \dot{x}_N(t) \\ \hline v(t) \\ e_I(t) \\ e_E(t) \end{bmatrix} = \left[\begin{array}{c|cc} A_N(t) & B_w(t) & B_d(t) \\ \hline C_v(t) & D_{vw}(t) & D_{vd}(t) \\ C_I(t) & D_{Iw}(t) & D_{Id}(t) \\ C_E(t) & 0 & 0 \end{array}\right] \begin{bmatrix} x_N(t) \\ \hline w(t) \\ d(t) \end{bmatrix} \tag{4.1}$$

In addition to notations defined earlier $v \in \mathbb{R}^{n_v}$ and $w \in \mathbb{R}^{n_w}$ are signals associated with the uncertainty Δ. The state vector is denoted as $x_N \in \mathbb{R}^{n_N}$ to refer to the states of system N.

4.3.1 Worst-Case Gain

The robust performance of the uncertain system $\mathcal{F}_u(N, \Delta)$ is assessed using the worst-case gain as defined below.

Definition 4.3.1. *Let an LTV system N be given by (4.1) and uncertainty $\Delta : \mathcal{L}_2^{n_v}[0,T] \to \mathcal{L}_2^{n_w}[0,T]$ be in some set $\mathbf{\Delta}$. Assume the interconnection $\mathcal{F}_u(N,\Delta)$ is well-posed. The worst-case gain is then defined as:*

$$\gamma_{wc} := \sup_{\Delta \in \mathbf{\Delta}} \|\mathcal{F}_u(N,\Delta)\|_{[0,T]}$$

The worst-case gain is the largest induced gain of the uncertain time-varying system over all uncertainties Δ in set $\mathbf{\Delta}$. This is difficult to compute directly because it involves optimizing over the entire uncertainty set. Instead, we focus on computing an upper bound on the worst-case gain using dissipation inequalities and IQC conditions.

4.4 Computation of Worst-Case Gain Upperbound

4.4.1 Integral Quadratic Constraints (IQCs)

IQCs [68, 69] are used to describe the input-output behavior of Δ. A time-domain formulation is used here to analyze the uncertain time-varying system. This formulation is based on the graphical interpretation as shown in Figure 4.2. Time-domain IQCs, as used in this dissertation, are defined for Δ by specifying a filter Ψ and a finite horizon constraint on the filter output z.

The LTV dynamics of filter Ψ on the horizon $[0,T]$ are given as follows:

$$\left[\begin{array}{c} \dot{x}_\psi(t) \\ \hline z(t) \end{array}\right] = \left[\begin{array}{c|cc} A_\psi(t) & B_{\psi v}(t) & B_{\psi w}(t) \\ \hline C_\psi(t) & D_{\psi v}(t) & D_{\psi w}(t) \end{array}\right] \left[\begin{array}{c} x_\psi(t) \\ \hline v(t) \\ w(t) \end{array}\right] \tag{4.2}$$

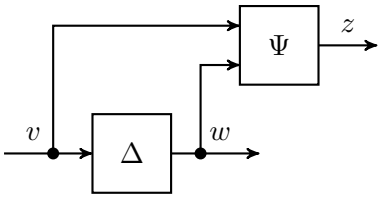

Figure 4.2: Graphical Interpretation for Time Domain IQCs

where $x_\psi \in \mathbb{R}^{n_\psi}$ is the state. The formal definition for a time-domain IQC is given next.

Definition 4.4.1. *Consider an LTV system* $\Psi : \mathcal{L}_2^{(n_v+n_w)}[0,T] \to \mathcal{L}_2^{n_z}[0,T]$ *and* $M : [0,T] \to \mathbb{S}^{n_z}$ *be given with* M *piecewise continuous. A bounded, causal operator* $\Delta : \mathcal{L}_2^{n_v}[0,T] \to \mathcal{L}_2^{n_w}[0,T]$ *satisfies the time domain IQC defined by* (Ψ, M) *if the following inequality holds for all* $v \in \mathcal{L}_2^{n_v}[0,T]$ *and* $w = \Delta(v)$*:*

$$\int_0^T z(t)^\top M(t) z(t)\, dt \geq 0 \tag{4.3}$$

where z *is the output of* Ψ *driven by inputs* (v,w) *with zero initial conditions* $x_\psi(0) = 0$*.*

Note that Definition 4.4.1 allows the IQC filter Ψ to be time-varying. This time-varying generalization provides an additional degree of freedom for finite horizon robustness analysis with IQCs. Similar generalizations for LPV systems are presented in [77] to use parameter-varying IQCs. However, exploring this additional degree of freedom is a subject of future research. Thus, the examples discussed later in this dissertation are for the special case where Ψ is an LTI filter. The notation $\Delta \in \mathcal{I}(\Psi, M)$ is used if Δ satisfies the IQC defined by (Ψ, M). A valid IQC $\mathcal{I}(\Psi, M)$ can be defined for a set $\mathbf{\Delta}$ such that $\mathbf{\Delta} \subseteq \mathcal{I}(\Psi, M)$. A few examples are provided below.

Example 4.4.1. *Let* $\mathbf{\Delta}$ *denote the set of nonlinear, time-varying, uncertainties with a given norm-bound* β*, i.e.* $\Delta \in \mathbf{\Delta}$ *if* $\|\Delta\|_{2\to 2,[0,T]} \leq \beta$*. If* $\Delta \in \mathbf{\Delta}$ *then for any constant* $\lambda \geq 0$ *it satisfies the IQC defined by a constant matrix:*

$$M := \begin{bmatrix} \beta^2 \lambda\, I_{n_v} & 0 \\ 0 & -\lambda\, I_{n_w} \end{bmatrix}$$

and a static filter $\Psi := I_{n_v+n_w}$*.*

Example 4.4.2. *Consider the set* $\mathbf{\Delta}$ *of nonlinear, time-varying, and memoryless uncertainties with given norm-bound* $\beta(t)$ *at each* $t \in [0,T]$*. In other words,* $w(t) = \Delta(v(t),t)$ *with* $\|w(t)\|_2 \leq \beta(t)\|v(t)\|_2$ *for all* $t \in [0,T]$*. In this case* Δ *satisfies the IQC defined by the time-varying matrix as follows and a static filter* $\Psi := I_{n_v+n_w}$*.*

$$M(t) := \begin{bmatrix} \beta(t)^2\lambda(t)\, I_{n_v} & 0 \\ 0 & -\lambda(t)\, I_{n_w} \end{bmatrix}$$

Example 4.4.3. *Let* $\mathbf{\Delta}$ *denote the set of LTI uncertainties* $\Delta \in \mathbb{RH}_\infty$ *with* $\|\Delta\|_\infty \leq 1$*. Let* (Ψ, M) *be defined as follows:*

$$\begin{aligned} \Psi &:= \begin{bmatrix} \Psi_{11} & 0 \\ 0 & \Psi_{11} \end{bmatrix} \text{ with } \Psi \in \mathbb{RH}_\infty^{n_z \times 1} \\ M &:= \begin{bmatrix} M_{11} & 0 \\ 0 & -M_{11} \end{bmatrix} \text{ with } M \in \mathbb{S}^{n_z} \text{ and } M_{11} \succ 0 \end{aligned} \tag{4.4}$$

It is shown in Appendix II of [66] that the pair (Ψ, M) *defines a valid time domain IQC for* Δ *over any* $T < \infty$ *i.e.* $\mathbf{\Delta} \subseteq \mathcal{I}(\Psi, M)$*.*

Example 4.4.4. *Let* $\mathbf{\Delta}$ *be the set of LTV parametric uncertainties* $\delta(t) \in \mathbb{R}$ *with a given norm-bound* $\beta(t)$*, i.e.* $w(t) = \delta(t) \cdot v(t)$*,* $|\delta(t)| \leq \beta(t)$*,* $\forall t \in [0,T]$*. Let* $n_v = n_w = n$ *and* $M_{11} : [0,T] \to \mathbb{S}^n$ *be piecewise continuous with* $M_{11}(t) \succ 0$*,* $\forall t \in [0,T]$*. Then* Δ *satisfies the IQC defined by the time-varying matrix:*

$$M(t) := \begin{bmatrix} \beta(t)^2\, M_{11}(t) & 0 \\ 0 & -M_{11}(t) \end{bmatrix} \tag{4.5}$$

and a static filter $\Psi := I_{2n}$*, i.e.* $\mathbf{\Delta} \subseteq \mathcal{I}(\Psi, M)$*.*

A library of IQCs is provided in [69, 78] for various types of perturbations. Most IQCs are for bounded, causal operators with multipliers Π specified in the frequency domain. Under mild assumptions, a valid time-domain IQC $\mathcal{I}(\Psi, M)$ can be constructed from Π via a J-spectral factorization [79].

4.4.2 Dissipation Inequality Condition

Consider an extended system as shown in Figure 4.3. This interconnection includes the IQC filter Ψ but the uncertainty Δ has been removed. The precise relation $w = \Delta(v)$ is replaced, for the analysis, by the constraint on the filter output z.

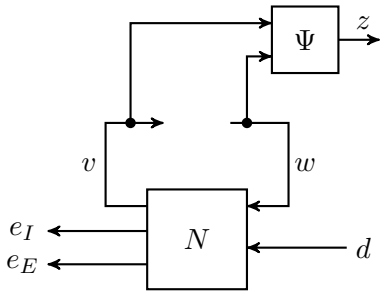

Figure 4.3: Analysis Interconnection

The extended system of N (Equation 4.1) and Ψ (Equation 4.2) is governed by the following state space model:

$$\left[\begin{array}{c} \dot{x}(t) \\ \hline z(t) \\ e_I(t) \\ e_E(t) \end{array}\right] = \left[\begin{array}{c|c} \mathcal{A}(t) & \mathcal{B}(t) \\ \hline \mathcal{C}_z(t) & \mathcal{D}_z(t) \\ \mathcal{C}_I(t) & \mathcal{D}_I(t) \\ \mathcal{C}_E(t) & 0 \end{array}\right] \left[\begin{array}{c} x(t) \\ \hline \begin{bmatrix} w(t) \\ d(t) \end{bmatrix} \end{array}\right] \tag{4.6}$$

The extended state vector is $x := \left[\begin{smallmatrix} x_N \\ x_\psi \end{smallmatrix}\right] \in \mathbb{R}^n$ where $n := n_N + n_\psi$. The state-space matrices are given by:

$$\mathcal{A} := \begin{bmatrix} A_N & 0 \\ B_{\psi v} C_v & A_\psi \end{bmatrix}, \mathcal{B} := \begin{bmatrix} B_w & B_d \\ B_{\psi v} D_{vw} + B_{\psi w} & B_{\psi v} D_{vd} \end{bmatrix}$$
$$\mathcal{C}_z := \begin{bmatrix} D_{\psi v} C_v & C_\psi \end{bmatrix}, \mathcal{C}_I := \begin{bmatrix} C_I & 0 \end{bmatrix}, \mathcal{C}_E := \begin{bmatrix} C_E & 0 \end{bmatrix}$$
$$\mathcal{D}_z := \begin{bmatrix} D_{\psi v} D_{vw} + D_{\psi w} & D_{\psi v} D_{vd} \end{bmatrix}, \mathcal{D}_I = \begin{bmatrix} D_{Iw} & D_{Id} \end{bmatrix}$$

The following differential linear matrix inequality (DLMI) is used to compute an upper

bound on the worst-case gain of $\mathcal{F}_u(N,\Delta)$.

$$\begin{bmatrix} \dot{P}+\mathcal{A}^\top P + P\mathcal{A} & P\mathcal{B} \\ \mathcal{B}^\top P & 0 \end{bmatrix} + \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} + \begin{bmatrix} \mathcal{C}_z^\top \\ \mathcal{D}_z^\top \end{bmatrix} M \begin{bmatrix} \mathcal{C}_z & \mathcal{D}_z \end{bmatrix} \leq -\epsilon I \tag{4.7}$$

This inequality depends on the IQC matrix M and γ. It can be compactly denoted as $DLMI_{Rob}(P,M,\gamma^2,t) \leq -\epsilon I$. This notation emphasizes that the constraint is a DLMI in (P,M,γ^2) for fixed N, Ψ and (Q,S,R,F). The next theorem states a sufficient DLMI condition to bound the generalized (robust) induced performance measure of $\mathcal{F}_u(N,\Delta)$. The proof is similar to Theorem 6 and 7 of [10] and is given below for completeness. It uses IQCs [69] and a standard dissipation argument [80, 81, 82].

Theorem 4.4.1. *Consider an LTV system N given by (4.1) and let $\Delta : \mathcal{L}_2^{n_v}[0,T] \to \mathcal{L}_2^{n_w}[0,T]$ be an operator. Assume $\mathcal{F}_u(N,\Delta)$ is well-posed and $\Delta \in \mathcal{I}(\Psi,M)$. Let $Q : [0,T] \to \mathbb{S}^n$, $S : [0,T] \to \mathbb{R}^{n\times(n_w+n_d)}$, $R : [0,T] \to \mathbb{S}^{(n_w+n_d)}$, and $F \in \mathbb{R}^{n\times n}$ be defined as follows.*

$$Q := \mathcal{C}_I^\top \mathcal{C}_I, \quad S := \mathcal{C}_I^\top \mathcal{D}_I, \quad R := \mathcal{D}_I^\top \mathcal{D}_I - \gamma^2 diag\{0_{n_w}, I_{n_d}\}, \quad F := \mathcal{C}_E(T)^\top \mathcal{C}_E(T) \tag{4.8}$$

If there exists $\epsilon > 0$, $\gamma > 0$ and a differentiable function $P : [0,T] \to \mathbb{S}^n$ such that $P(T) \geq F$ and,

$$DLMI_{Rob}(P,M,\gamma^2,t) \leq -\epsilon I, \quad \forall t \in [0,T]. \tag{4.9}$$

then $\|\mathcal{F}_u(N,\Delta)\|_{[0,T]} < \gamma$.

Proof. Let $d \in \mathcal{L}_2[0,T]$ and $x_N(0) = 0$ be given. By well-posedness, $F_u(N,\Delta)$ has a unique solution (x_N, v, w, e_I, e_E). Define $x := \left[\begin{smallmatrix} x_N \\ x_\psi \end{smallmatrix}\right]$. Then (x, z, e_I, e_E) are a solution of the extended system (4.6) with inputs (w,d) and initial condition $x(0) = 0$. Moreover, z satisfies the the IQC defined by (Ψ, M). Define a storage function by $V(x,t) := x^\top P(t) x$. Left and right multiply the DLMI (4.7) by $[x^\top, w^\top, d^\top]$ and its transpose to show that

V satisfies the following dissipation inequality for all $t \in [0,T]$:

$$\dot{V} + \begin{bmatrix} x \\ [\begin{smallmatrix} w \\ d \end{smallmatrix}] \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} x \\ [\begin{smallmatrix} w \\ d \end{smallmatrix}] \end{bmatrix} + z^\top M z \leq -\epsilon\, d^\top d. \tag{4.10}$$

Use the choices for (Q,S,R) to rewrite the second term as $e_I^\top e_I - \gamma^2 d^\top d$. Integrate over $[0,T]$ to obtain:

$$x(T)^\top P(T) x(T) + \int_0^T z(t)^\top M(t) z(t)\, dt + \|e_I\|_{2,[0,T]}^2 \leq (\gamma^2 - \epsilon)\|d\|_{2,[0,T]}^2.$$

Apply $P(T) \geq F = \mathcal{C}_E(T)^\top \mathcal{C}_E(T)$ and $\Delta \in \mathcal{I}(\Psi, M)$ to conclude:

$$\|e_E(T)\|_2^2 + \|e_I\|_{2,[0,T]}^2 \leq (\gamma^2 - \epsilon)\|d\|_{2,[0,T]}^2. \tag{4.11}$$

This inequality implies $\|F_u(N,\Delta)\|_{[0,T]} < \gamma$. □

4.4.3 Computational Approach

Numerical implementation using IQCs often involves a fixed choice of Ψ and optimization subject to the convex constraints on M. Two examples are provided as follows.

Example 4.4.5. *Consider an LTI uncertainty $\Delta \in \mathbb{RH}_\infty$ with $\|\Delta\|_\infty \leq 1$. By Example 4.4.3, Δ satisfies any IQC (Ψ, M) with $\Psi := \left[\begin{smallmatrix} \Psi_{11} & 0 \\ 0 & \Psi_{11} \end{smallmatrix}\right]$, $M := \left[\begin{smallmatrix} M_{11} & 0 \\ 0 & -M_{11} \end{smallmatrix}\right]$, and $M_{11} \succ 0$. A typical choice for Ψ_{11} is:*

$$\Psi_{11} := \left[1, \frac{1}{(s+p)}, \dots \frac{1}{(s+p)^q}\right]^\top \quad \textit{with } p > 0 \tag{4.12}$$

The analysis is performed by selecting (p,q) to obtain (fixed) Ψ and optimizing over the convex constraint $M_{11} \succ 0$. The results depend on the choice of (p,q). Larger values of q represent a richer class of IQCs and hence yield less conservative results but with increasing computational cost. Note that the IQC filter Ψ is not square in general with $n_z = 2(q+1)$ outputs.

Example 4.4.6. *Conic combinations of multiple IQCs can be incorporated in analysis. Let (Ψ_i, M_i) with $i = 1, 2, \dots, N$ define N valid IQCs for Δ. Hence $\int_0^T z_i^\top M_i z_i\, dt \geq 0$*

where z_i is the output Ψ_i driven by v and $w = \Delta(v)$. The multiple constraints can be multiplied by $\lambda_i \geq 0$ and combined to yield:

$$\int_0^T \sum_{i=1}^{N} \lambda_i z_i^\top M_i z_i \, dt \geq 0 \tag{4.13}$$

Thus a valid time-domain IQC for Δ is given by

$$\Psi := \begin{bmatrix} \Psi_1 \\ \vdots \\ \Psi_N \end{bmatrix} \text{ and } M(\lambda) := \begin{bmatrix} \lambda_1 M_1 & & \\ & \ddots & \\ & & \lambda_N M_N \end{bmatrix} \tag{4.14}$$

The analysis optimizes over λ given selected (Ψ_i, M_i).

An iterative algorithm given in [10] is used to compute the smallest upper bound on the worst-case gain. It combines the DLMI formulation in the Theorem 4.4.1 with a related Riccati Differential Equation (RDE). The algorithm returns the upper bound $\bar{\gamma}_{wc}$ along with the decision variables P and M.

4.5 Computation of Worst-Case Gain Lower Bound

4.5.1 Sample-Based Analysis

One method to compute a lower bound on the worst-case gain is to perform a sample-based analysis. In this approach, candidate uncertainties Δ_i where $i = 1, 2, \ldots, N$ are sampled from the uncertainty set $\mathbf{\Delta}$. Each Δ_i is substituted in the uncertain LFT interconnection (Figure 4.1) to form $\mathcal{F}_u(N, \Delta_i)$. The nominal analysis is then performed on the individual interconnection to compute the corresponding bound on induced gain and worst-case disturbance. Maximizing the gain over the sample space yields the lower bound on gain and corresponding worst-case uncertainty/disturbance. A similar approach is used in [10] for the two-link robot arm discussed in Section 2.6. Moreover, as noted before, the work in [31] extends this sample-based robustness analysis to match the nonlinear performance closely. This is done by accounting for the shift in the nominal trajectory due to the sampled uncertainty. The computational conditions in [31] are similar to the Gramian based result presented in Section 3.8.2. In general, we note that

the computational improvements proposed in Chapter 3 can significantly speed up such sample-based analysis.

4.5.2 Power Iteration

The power iteration for nominal analysis was presented in Section 3.5.1. This section presents a different power iteration algorithm for computing a lower bound on the worst-case gain of an uncertain LTV system. This algorithm can be viewed as a particular case of the work in [8, 74] or finite horizon LTV generalization of the work in [20, 18, 19].

Consider the interconnection $\mathcal{F}_u(N, \Delta)$ shown in Figure 4.1. To simplify notation, we assume $n_E = 0$ and demonstrate the computation of a finite horizon worst-case induced $\mathcal{L}_2$-gain lower bound. The state-space representation of the system N is given in Equation (4.1). The corresponding adjoint (costate) system $N^{\sim} : \mathcal{L}_2^{n_v+n_I}[0,T] \to \mathcal{L}_2^{n_w+n_d}[0,T]$ has the following form:

$$\left[\begin{array}{c} \dot{p}(t) \\ \hline s(t) \\ r(t) \end{array}\right] = \left[\begin{array}{c|cc} -A_N(t)^\top & -C_v(t)^\top & -C_I(t)^\top \\ \hline B_w(t)^\top & D_{vw}(t)^\top & D_{Iw}(t)^\top \\ B_d(t)^\top & D_{vd}(t)^\top & D_{Id}(t)^\top \end{array}\right] \left[\begin{array}{c} p(t) \\ \hline z(t) \\ q(t) \end{array}\right] \tag{4.15}$$

where $p(t) \in \mathbb{R}^{n_x}$ is the adjoint state, $z(t) \in \mathbb{R}^{n_v}$ and $q(t) \in \mathbb{R}^{n_I}$ are the adjoint inputs, and $s(t) \in \mathbb{R}^{n_w}$ and $r(t) \in \mathbb{R}^{n_d}$ are the adjoint outputs. The boundary conditions for the costate is $p(T) = 0$. The uncertainty Δ is assumed to have at-most unit induced $\mathcal{L}_2$-to-$\mathcal{L}_2$ gain i.e. $\|\Delta\|_{2\to 2,[0,T]} \leq 1$. Thus, the input-output of Δ satisfy the following constraint on a finite horizon:

$$\int_0^T w(\tau)^\top w(\tau)\, d\tau \leq \int_0^T v(\tau)^\top v(\tau)\, d\tau \tag{4.16}$$

The power iteration algorithm for an uncertain LTV system is summarized in Algorithm 4. The signal $w^{(1)}$ and $d^{(1)}$ can be initialized randomly or by any other means. These signals are scaled to have unit $\mathcal{L}_2[0,T]$ norm. The algorithm is also initialized with a maximum number of iterations i_{max}, and desired an absolute tolerance ϵ_a. The forward simulation of N from $t = 0$ to $t = T$ is performed by exciting individual input channels, i.e. one forward simulation with input $\left[\begin{smallmatrix} w^{(i)} \\ 0_{n_d} \end{smallmatrix}\right]$ to obtain output $\left[\begin{smallmatrix} v_1 \\ e_{I1} \end{smallmatrix}\right]$ and

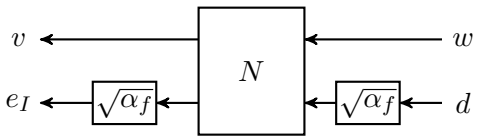

Figure 4.4: Scaled System N_{scl}

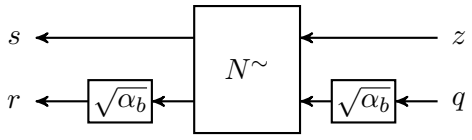

Figure 4.5: Scaled Adjoint $N^\sim_{scl}$

second simulation with the input $\begin{bmatrix} 0_{n_w} \\ d^{(i)} \end{bmatrix}$ to obtain output $\begin{bmatrix} v_2 \\ e_{I2} \end{bmatrix}$. Next, the responses of these two simulations are combined with an inclusion of a variable $\sqrt{\alpha_f}$ as shown in Figure 4.4 to generate output equations for e_I and v as follows:

$$v = v_1 + \sqrt{\alpha_f}\, v_2 \tag{4.17}$$

$$e_I = \sqrt{\alpha_f}(e_{I1} + \sqrt{\alpha_f}\, e_{I2}) \tag{4.18}$$

The $\mathcal{L}_2[0,T]$ norm of the stacked output $\begin{bmatrix} v \\ e_I \end{bmatrix}$ is set as 1 to yield a quartic polynomial in $\sqrt{\alpha_f}$, which can be solved efficiently. The corresponding worst-case gain can be computed as $\gamma_f = 1/\alpha_f$. A similar process is repeated for the backward simulation with $N^\sim$. The alignment conditions in Algorithm 4 ensure that the uncertainty and performance channels are scaled appropriately, and signals do not blow up. Specifically, we enforce the constraint $\|v\|_{2,[0,T]} = \|w\|_{2,[0,T]}$, which is different from the inequality (4.16). A similar constraint is also enforced in [8] for the nonlinear power iterations. The consequences of enforcing equality constraint instead of the inequality are to be better studied in future work. Finally, we note that the power iteration algorithm can be modified with appropriate alignment conditions for different classes of uncertainties (e.g., real parametric uncertainties).

Algorithm 4 terminates if the forward worst-case gain γ_f is within the absolute tolerance ϵ_a of the previously computed value. The algorithm can also stop if the maximum number of iterations is reached. The outputs are the worst-case gain lower bound γ_o, signals w_o, v_o associated with the uncertainty channels and d_o, e_{I_o} associated with the performance channels. These signals are defined on $[0,T]$ and can be used to study the worst-case performance. This algorithm can be modified to include the terminal Euclidean norm penalty on the output ($n_E \neq 0$).

In summary, each power iteration requires total of $4n_x$ scalar ODE integrations (i.e. $2n_x$ scalar ODE integrations for forward simulation of N and $2n_x$ scalar ODE

integrations for backward simulation of $N^{\sim}$), where n_x is the order of a system. Overall, the computational cost of the LTV power iteration is low compared to the IQC upper bound algorithm that combines complementary benefits of DLMI and RDE conditions [10]. However, the convergence properties of the Algorithm 4 should further be investigated as future work.

Algorithm 4 Power Iteration for the Uncertain LTV System

1: **Given:** N, $N^{\sim}$
2: **Initialize:** $w^{(1)}$ and $d^{(1)}$ with $\|w^{(1)}\|_{2,[0,T]} = \|d^{(1)}\|_{2,[0,T]} = 1$, i_{max}, ϵ_a, $\gamma_f^{(0)} = -\infty$, $\gamma_b^{(0)} = -\infty$.
3: **for** $i = 1 : i_{max}$ **do**
4: **Forward Sim:** Simulate N from $t = 0$ to $t = T$ with $x(0) = 0$ and input $\begin{bmatrix} w^{(i)} \\ 0_{n_d} \end{bmatrix}, \begin{bmatrix} 0_{n_w} \\ d^{(i)} \end{bmatrix}$ to generate outputs $v^{(i)}$, $e_I^{(i)}$. Solve for the quartic polynomial in $\sqrt{\alpha_f^{(i)}}$ to have unit $\mathcal{L}_2[0,T]$ norm for the output of a system in Figure 4.4.
5: **Compute Forward Worst-Case Gain:** $\gamma_f^{(i)} = 1/\alpha_f^{(i)}$.
6: **Alignment Condition:** $z^{(i)} = v^{(i)} \frac{\|w^{(i)}\|_{2,[0,T]}}{\|v^{(i)}\|_{2,[0,T]}}$, $q^{(i)} = e_I^{(i)} \frac{\|d^{(i)}\|_{2,[0,T]}}{\|e_I^{(i)}\|_{2,[0,T]}}$.
7: **Backward Sim:** Simulate $N^{\sim}$ from $t = T$ to $t = 0$ with $p(T) \equiv 0$ and input $\begin{bmatrix} z^{(i)} \\ 0_{n_I} \end{bmatrix}, \begin{bmatrix} 0_{n_v} \\ q^{(i)} \end{bmatrix}$ to generate outputs $s^{(i)}$, $r^{(i)}$. Solve for the quartic polynomial in $\sqrt{\alpha_b^{(i)}}$ to have unit $\mathcal{L}_2[0,T]$ norm for the output of a system in Figure 4.5.
8: **Compute Backward Worst-Case Gain:** $\gamma_b^{(i)} = 1/\alpha_b^{(i)}$.
9: **Alignment Condition:** $w^{(i+1)} = s^{(i)} \frac{\|v^{(i)}\|_{2,[0,T]}}{\|s^{(i)}\|_{2,[0,T]}}$, $d^{(i+1)} = r^{(i)} \frac{\|e_I^{(i)}\|_{2,[0,T]}}{\|r^{(i)}\|_{2,[0,T]}}$.
10: **Stop Condition:** Terminate if $\gamma_f^{(i)} - \gamma_f^{(i-1)} < \epsilon_a$.
11: **end for**
12: **Compute Output:** Simulate N using $w_o = w^{(i+1)}$ and $d_o = d^{(i+1)}$ to obtain outputs v_o and e_{I_o} that correspond to the gain of $\gamma_o = \gamma_f^{(i)}$.
13: **Output:** γ_o, w_o, d_o, v_o, e_{I_o}.

Remark 4.5.1. *Algorithm 4 does not enforce causality constraint for the input and output of the uncertainty Δ [74]. As a result, it searches over a larger class of (causal and non-causal) uncertain input-output pairs. Thus, γ_o is not a guaranteed upper or lower bound on the worst-case gain. This issue of causality is further discussed in future work (Section 8.2.2).*

4.6 Trajectory Sensitivity

This section presents a first-order trajectory sensitivity analysis approach for systems with parametric uncertainties. Consider a nonlinear system given by the following differential equation:

$$\dot{\eta}(t) = f(\eta(t), p), \qquad \eta(0) = \eta_0 \tag{4.19}$$

$$\xi(t) = g(\eta(t)) \tag{4.20}$$

where $\eta(t) \in \mathbb{R}^{n_\eta}$ is the nonlinear state, $p \in [p_{min}, p_{max}] \subset \mathbb{R}^{n_p}$ is the uncertain parameter vector, and $\xi(t) \in \mathbb{R}$ is the scalar output of interest. The analysis goal is to compute the deviation in $\xi(t)$ due to parametric uncertainties in the system. The dynamics of a nonlinear system are numerically linearized around the nominal trajectory $\bar{\eta}$ and nominal parameter vector $\bar{p}$ to yield the following LTV system on the horizon $[0, T]$:

$$\delta\dot{\eta}(t) = A(t)\,\delta\eta(t) + B(t)\,\delta p, \qquad \delta\eta(0) = 0 \tag{4.21}$$

$$\delta\xi(t) = C(t)\,\delta\eta(t) \tag{4.22}$$

The following relation is obtained by integrating the above LTV dynamics on $[0, T]$:

$$\delta\xi(t) = \Gamma(t)\,\delta p \tag{4.23}$$

where $\Gamma(t) := C(t)\int_0^t \Phi(t,\tau)B(\tau)\,d\tau$. Define $\Delta p_{min} := p_{min} - \bar{p}$ and $\Delta p_{max} := p_{max} - \bar{p}$. The upper bound ($\xi_{ub}$) and lower bound ($\xi_{lb}$) can be computed by solving the following linear programming problems, respectively.

$$\xi_{ub}(t) = \max_{\Delta p_{min} \le \Delta p \le \Delta p_{max}} [\xi(t, \bar{p}) + \Gamma(t)\Delta p] \tag{4.24}$$

$$\xi_{lb}(t) = \min_{\Delta p_{min} \le \Delta p \le \Delta p_{max}} [\xi(t, \bar{p}) + \Gamma(t)\Delta p] \tag{4.25}$$

The proposed first-order sensitivity analysis at each time t also gives the vector Δp of decision variables which can be used to compute the worst-case parameter combinations.

These worst-case parameter combinations can further be studied in high-fidelity nonlinear simulation. This approach is used in Section 6.4 to study the impact of uncertainties in aerodynamic parameters for the interceptor missile.

4.7 Numerical Examples

4.7.1 LTI Example

Consider the following uncertain LTI system on the horizon $[0, T]$:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & -1.4 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0.7746 \end{bmatrix} w + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} d, \qquad x(0) = 0 \tag{4.26}$$

$$v = 0.07746\, d \tag{4.27}$$

$$e = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x \tag{4.28}$$

where $w = \Delta v$ and Δ is an LTI uncertainty with $\|\Delta\|_\infty \leq 0.6$. The infinite horizon worst-case gain upper and lower bound (computed using MATLAB's `wcgain` function) are 0.1774 and 0.1771, respectively. The transfer function for the worst-case LTI uncertainty is also obtained as:

$$\Delta_{wc}(s) = \frac{-0.555s^3 + 0.727s^2 - 0.238s}{s^4 + 2.234s^3 + 2.068s^2 + 0.957s + 0.183} \tag{4.29}$$

Next, the finite horizon IQC upper bound algorithm from [10] is run, with parameterization similar to Example 4.4.5 using $p = -10$ and $q = 1$. The upper bound on the worst-case induced-$\mathcal{L}_2$ gain is computed for increasing finite horizon $T = 0.1, 1, 2, \ldots, 20, 50$. These upper bounds (red-dashed) are shown in Figure 4.6. For comparison, the power iteration (Algorithm 4) is also run to compute a bound on the worst-case induced-$\mathcal{L}_2$ gain. These bounds (blue-solid) are also shown in Figure 4.6. These results are not directly comparable because linearity and causality of Δ are not explicitly enforced while running the power iteration algorithm. Despite a technical gap in these two analyses, we notice a good agreement between the computed bounds. Future research is required to develop the lower bound algorithm and construct related worst-case uncertainty on

a finite horizon.

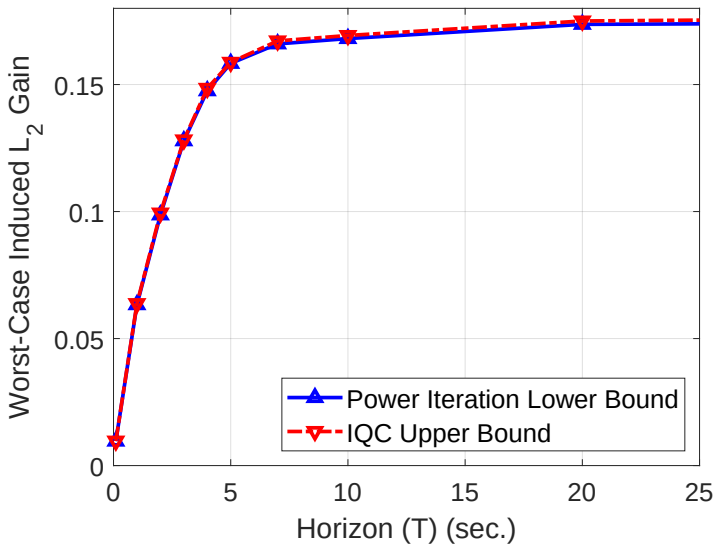

Figure 4.6: Comparison of Power Iteration Lower Bound and IQC Upper Bound

4.7.2 Nonlinear System: Two-Link Robot

We revisit the two-link robot example presented in Section 2.6. The work in [10] considers IQC robustness analysis of the closed-loop with a finite horizon LQR controller. The LTI uncertainty Δ in the second control channel with $\|\Delta\|_\infty \leq \beta$ and $\beta = 0.8$ was considered. This uncertainty may represent unmodeled motor dynamics at the second link joint. This example focuses on the output feedback setting to study closed-loop robustness properties. In particular, we focus on computing worst-case $\mathcal{L}_2$-to-Euclidean gain upper bound for the output feedback configuration discussed next.

Consider the output-feedback interconnection as shown in Figure 4.7. The output $y = \begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \end{bmatrix}$ is filtered through the approximate derivative filter $F(s) := \frac{s}{\tau s+1}$ to compute the rate estimates $\begin{bmatrix} \delta\dot{\theta}_1 \\ \delta\dot{\theta}_2 \end{bmatrix}$. The estimated linearized state $\hat{x}$ is then fed back to the LQR controller to compute the linearized control input $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$. As noted before, the uncertainty Δ is in the second control channel with $\|\Delta\|_\infty \leq \beta$.

The Bode magnitude diagram of the output filter is shown in Figure 4.8. If the filter

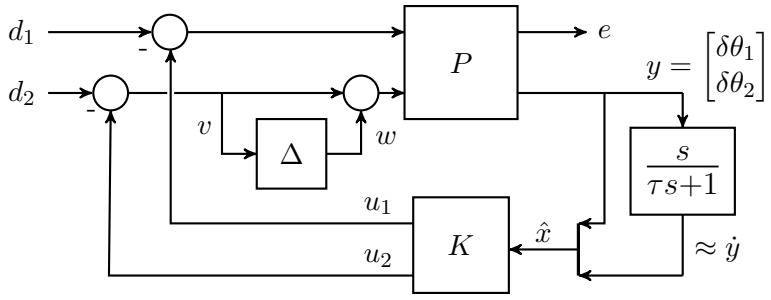

Figure 4.7: Output-Feedback Interconnection with Rate-Estimates

time constant $\tau = 0$, we have a perfect derivative of output y, and the closed-loop performance matches that of a finite horizon LQR controller. However, a perfect derivative of output amplifies the high-frequency noise. In contrast, filter $F(s)$ with $\tau = 0.1$ limits the high-frequency noise but at the expense of degraded low-frequency performance (e.g., disturbance rejection, reference tracking, robustness to unmodeled dynamics). To see this, consider the worst-case induced $\mathcal{L}_2$-to-Euclidean gain bound computed using the IQC upper bound algorithm [10] as the uncertainty level β is increased. The IQC parameterization similar to Example 4.4.5 is used with $p = -10$ and $q = 1$. The upper bounds are shown in Figure 4.9 for different values of τ. We note that the LQR controller is robust to a higher uncertainty level, as the worst-case gain is consistent 0.0592 until $\beta = 0.9$. However, the output feedback controller with $\tau = 0.02, 0.05$ or 0.1 have comparatively higher worst-case gain.

Figure 4.10 shows an impact of the output feedback controller in the joint space at an uncertainty level $\beta = 0.1$. The cyan disk represents the bound at the end of a trajectory when an LQR controller is used in the feedback, whereas the red disk shows the upper bound on the gain when the output feedback controller is used with filter time-constant $\tau = 0.1$. The disk for $\tau = 0.02$ and $\tau = 0.05$ is also shown in green and orange, respectively. These results show that the trajectories can be pushed even further at the final time T in the output feedback setting.

In summary, the finite horizon LQR has good robustness properties, whereas these properties are degraded in the output feedback configuration. This observation is consistent with the LTI counterpart as the LQR design guarantees good robustness margins, whereas there are no guaranteed margins for LQG regulators [83]. A natural question is:

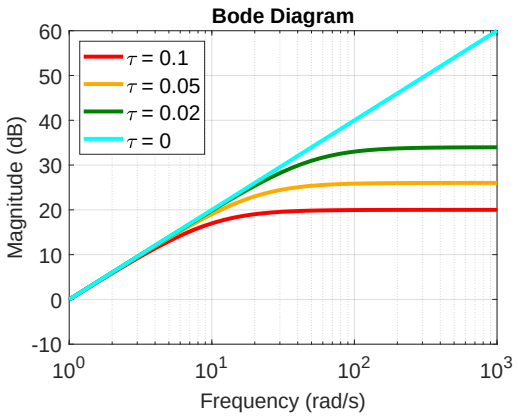

Figure 4.8: Bode Magnitude of Filter $F(s)$

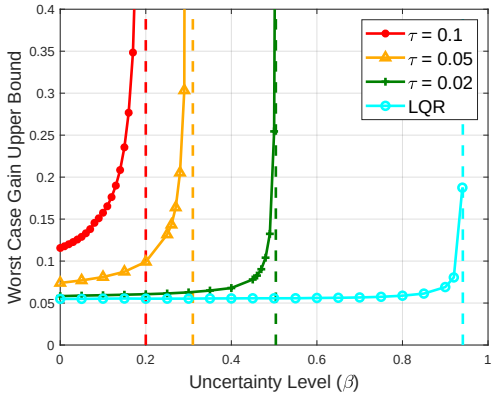

Figure 4.9: Worst-Case Gain

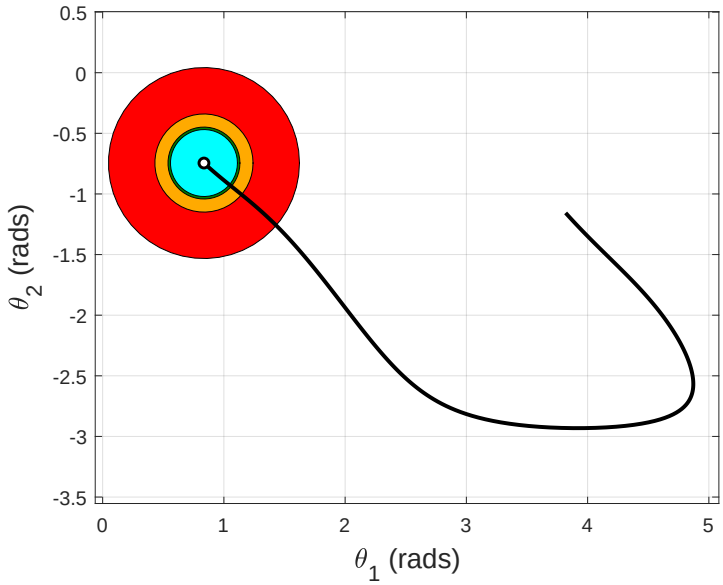

Figure 4.10: Illustration of the Worst-Case Terminal Euclidean Norm Bound for $\beta = 0.1$

can we design a finite horizon LTV controller that explicitly accounts for the uncertainty and external disturbances? The answer is given by robust synthesis work presented in the next chapter.

4.8 Summary

This chapter presented different robustness analysis tools for the uncertain LTV system on a finite horizon. These include IQC upper bound and power iteration lower bound computations for the worst-case LTV performance. We also discussed the first-order trajectory sensitivity analysis approach for parametric uncertainties. The last example demonstrated the degradation of robustness in the output-feedback setting, which motivates the robust synthesis work in the next chapter.

Chapter 5

Robust Synthesis

5.1 Introduction

This chapter considers robust synthesis problem for uncertain linear time-varying (LTV) systems on finite horizons. This problem is motivated by engineering systems that follow a finite-time trajectory and for which model uncertainty is a significant factor. The examples include: aircraft landings [84], missile interceptors [85] and space-launch or reentry systems [86, 70, 87]. Robust synthesis can be used to ensure the stability and robustness of the linearized closed-loop over a range of parametric and dynamic uncertainties. Many existing robust synthesis algorithms, e.g., μ-synthesis [88, 89, 90, 91] have been developed for uncertain linear time-invariant (LTI) system and infinite horizon robustness metrics. This enables the use of frequency-domain techniques. In contrast, this chapter considers uncertain finite horizon LTV systems using time-domain techniques.

The objective is to synthesize a controller to minimize the worst-case performance overall allowable uncertainties. This robust synthesis problem leads, in general, to a non-convex optimization. The proposed algorithm, presented in Section 5.4 iterates between a nominal synthesis step and robustness analysis step. The nominal synthesis step relies on existing finite horizon H_∞ synthesis results which consider a control-theoretic formulation [84, 92, 23, 93]. An alternative game-theoretic formulation is considered in [94] which provides equivalent synthesis conditions. These conditions can be stated in terms of two coupled Riccati Differential Equations (RDEs) [45, 95, 96] or

two coupled Riccati Differential Inequalities (RDIs) [97]. We use the two coupled RDEs to provide a numerical advantage over the RDI conditions. Moreover, in contrast to other work, the results in [45, 95, 96] allow for terminal Euclidean norm penalties on the output. The approach presented in [10] (revisited in Chapter 4) will be used for the robustness analysis. Finally, scaled plant construction is required to link the nominal synthesis and robustness analysis steps.

5.2 Summary of Related Work

The proposed method in this chapter is analogous to the existing DK iteration method for uncertain LTI systems on infinite horizons. The algorithm in this chapter generalizes this method to uncertain LTV systems on finite horizons. Similar extensions have been made in [98, 99] for Linear Parameter-Varying (LPV) systems. Two other closely related works are [100] and [101]. The work in [100] considers an extension of the Glover-McFarlane loop-shaping method to LTV systems on infinite horizons. This leads to a robust stabilization problem with a single full block uncertainty. The work in [101] provides convex synthesis conditions for robust performance of uncertain LTV systems. However, [101] assumes uncertainty to lie in a contractive subset and is block partitioned with $(2, 2)$ block being zero. This special structure is used to convexify synthesis optimization. The algorithm proposed in this chapter considers a more general robust performance formulation than in [100] and [101], which allows us to design output-feedback controllers that robustly bound the reachable set of a finite horizon uncertain LTV system.

5.3 Nominal Synthesis

This subsection provides conditions to synthesize a controller that is optimal with respect to the nominal performance metric introduced in the previous subsection. Consider the feedback interconnection shown in Figure 5.1.

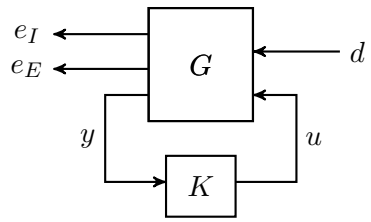

Figure 5.1: Nominal Feedback Interconnection $\mathcal{F}_l(G, K)$

The LTV system G defined on $[0, T]$ is given by:

$$\left[\begin{array}{c} \dot{x}(t) \\ \hline e_I(t) \\ e_E(t) \\ y(t) \end{array}\right] = \left[\begin{array}{c|cc} A(t) & B_d(t) & B_u(t) \\ \hline C_I(t) & 0 & D_{Iu}(t) \\ C_E(t) & 0 & 0 \\ C_y(t) & D_{yd}(t) & 0 \end{array}\right] \left[\begin{array}{c} x(t) \\ \hline d(t) \\ u(t) \end{array}\right] \tag{5.1}$$

where $d(t) \in \mathbb{R}^{n_d}$ is the generalized disturbance, $u(t) \in \mathbb{R}^{n_u}$ is the control input and $y(t) \in \mathbb{R}^{n_y}$ is the measured output. The generalized disturbance is of the form $d(t) = \left[\begin{smallmatrix} d_{in}(t) \\ n(t) \end{smallmatrix}\right]$, where $n(t) \in \mathbb{R}^{n_y}$ is a measurement noise and $d_{in}(t)$ represents all other disturbance inputs. This plant structure also assumes no feedthrough from d to e_E. This is required to ensure that the nominal performance metric is well-posed. In addition, the standard H_∞ synthesis framework imposes additional structure on the matrices relating d to e_I and d to y. This is done to simplify notation and is obtained via standard loop transformations under some minor technical assumptions (Chapter 17 of [21]). This leads to the following additional structure on the plant matrices:

$$C_I := \begin{bmatrix} 0 \\ C_1 \end{bmatrix} \quad D_{Iu} := \begin{bmatrix} I_{n_u} \\ 0 \end{bmatrix} \quad D_{yd} := \begin{bmatrix} 0 & I_{n_y} \end{bmatrix}$$

The nominal synthesis problem is to find a causal linear time-varying controller $K : \mathcal{L}_2^{n_y}[0, T] \to \mathcal{L}_2^{n_u}[0, T]$ that optimizes the closed-loop nominal performance, i.e.:

$$\inf_K \|\mathcal{F}_l(G, K)\|_{[0,T]}$$

As noted previously, if $n_E = 0$ then the nominal performance metric is the (finite

horizon) induced $\mathcal{L}_2$ gain. In this case, the synthesis problem is equivalent to the existing finite horizon H_∞ problem as considered in [23, 93]. The theorem below states the necessary and sufficient conditions for the existence of a γ-suboptimal controller for the nominal performance metric (with n_E not necessarily equal to zero). Theorem 5.3.1 is a special case of results presented in [45, 96].

Theorem 5.3.1. *Consider an LTV system* (5.1) *with $\gamma > 0$ given. Let B, $\hat{C}$, $\bar{R}$ and $\hat{R}$ be defined as follows.*

$$B := \begin{bmatrix} B_d & B_u \end{bmatrix}, \quad \bar{R} := diag\{-\gamma^2 I_{n_d}, I_{n_u}\}, \quad \hat{C} := \begin{bmatrix} C_I^\top & C_y^\top \end{bmatrix}^\top, \quad \hat{R} := diag\{-\gamma^2 I_{n_I}, I_{n_y}\}$$

1. *There exists an admissible output feedback controller K such that $\|\mathcal{F}_l(G,K)\|_{[0,T]} < \gamma$ if and only if the following three conditions hold:*

   (a) *There exists a differentiable function $X : [0,T] \to \mathbb{S}^{n_x}$ such that $X(T) = C_E(T)^\top C_E(T)$,*

   $$\dot{X} + A^\top X + XA - XB\bar{R}^{-1}B^\top X + C_I^\top C_I = 0$$

   (b) *There exists a differentiable function $Y : [0,T] \to \mathbb{S}^{n_x}$ such that $Y(0) = 0$,*

   $$-\dot{Y} + AY + YA^\top - Y\hat{C}^\top \hat{R}^{-1}\hat{C}Y + B_d B_d^\top = 0$$

   (c) *$X(t)$ and $Y(t)$ satisfy the following point-wise in time spectral radius condition,*

   $$\rho(X(t)Y(t)) < \gamma^2, \quad \forall t \in [0,T] \tag{5.2}$$

2. *If the conditions above are satisfied, then the closed loop performance $\|\mathcal{F}_l(G,K)\|_{[0,T]} < \gamma$ is achieved by the following central controller:*

$$\begin{aligned} \dot{\hat{x}}(t) &= A_K(t)\,\hat{x}(t) + B_K(t)\,y(t) \\ u(t) &= C_K(t)\,\hat{x}(t) \end{aligned}$$

where

$$
\begin{aligned}
Z &:= (I - \gamma^{-2} Y X)^{-1} \\
A_K &:= A + \gamma^{-2} B_d B_d^\top X - Z Y C_y^\top C_y - B_u B_u^\top X \\
B_K &:= Z Y C_y^\top \\
C_K &:= -B_u^\top X
\end{aligned}
$$

For a given $\gamma > 0$, the RDEs associated with X and Y are integrated backward and forward in time, respectively. If solution to both RDEs exist then the spectral radius coupling condition (5.2) is checked. If all three conditions are satisfied then the central controller achieves a closed-loop performance of γ. The smallest possible value of γ is obtained using bisection. The results in [45, 96] also consider the effect of uncertain initial conditions.

5.4 Robust Synthesis

5.4.1 Problem Formulation

An uncertain feedback interconnection is shown in Figure 5.2 where G is an LTV system on $[0, T]$ and Δ is assumed to lie in some set $\mathbf{\Delta}$ that is described by valid time domain IQCs.

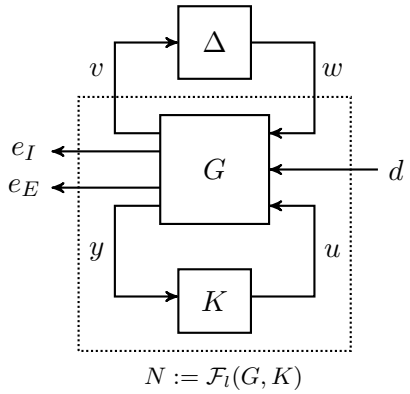

Figure 5.2: Uncertain Feedback Interconnection $\mathcal{F}_u(\mathcal{F}_l(G, K), \Delta)$

The finite horizon robust synthesis problem is to synthesize a controller which minimizes the impact of both worst-case disturbances and worst-case uncertainties, i.e.:

$$\inf_{K} \sup_{\Delta \in \mathbf{\Delta}} \|\mathcal{F}_u(\mathcal{F}_l(G,K),\Delta)\|_{[0,T]} \tag{5.3}$$

Let the LTV system G defined on $[0,T]$ be given as:

$$\left[\begin{array}{c} \dot{x}_G(t) \\ \hline v(t) \\ e_I(t) \\ e_E(t) \\ y(t) \end{array}\right] = \left[\begin{array}{c|ccc} A_G(t) & B_w(t) & B_d(t) & B_u(t) \\ \hline C_v(t) & D_{vw}(t) & D_{vd}(t) & D_{vu}(t) \\ C_I(t) & D_{Iw}(t) & D_{Id}(t) & D_{Iu}(t) \\ C_E(t) & 0 & 0 & 0 \\ C_y(t) & D_{yw}(t) & D_{yd}(t) & D_{yu}(t) \end{array}\right] \left[\begin{array}{c} x_G(t) \\ \hline w(t) \\ d(t) \\ u(t) \end{array}\right] \tag{5.4}$$

where $x_G \in \mathbb{R}^{n_G}$ is the state. This plant structure has no feedthrough from d to e_E for well-posedness. The synthesis problem (5.3) involves the worst-case gain computed over the entire uncertainty set. As noted earlier, instead we focus on minimizing worst-case gain upper bounds. In other words, we define IQCs $\mathcal{I}(\Psi, M)$ such that $\mathbf{\Delta} \subseteq \mathcal{I}(\Psi, M)$ and maximize over $\Delta \in \mathcal{I}(\Psi, M)$ in Equation (5.3). The goal is to design a linear time-varying controller $K : \mathcal{L}_2^{n_y}[0,T] \to \mathcal{L}_2^{n_u}[0,T]$ to minimize the worst-case gain upper bound on $\mathcal{F}_u(\mathcal{F}_l(G,K),\Delta)$. This leads to a non-convex synthesis problem and involves solving for the controller as well as IQC multipliers.

The approach taken here is to decompose the synthesis into two subproblems. First, solve a nominal synthesis problem (on a specially constructed scaled plant) to obtain K. Second, solve an IQC analysis problem to compute the worst-case gain upper bound. These subproblems can be solved iteratively, similar to coordinate descent, to get a reasonable sub-optimal solution. The proposed algorithm utilizes this approach to obtain a finite horizon sub-optimal controller. As with DK synthesis, there are no guarantees that the coordinate-wise iteration will lead to local optima, let alone global optima. However, it is a useful heuristic that enables the robust synthesis to extend naturally from LTI to finite-horizon LTV systems. The following assumption is made for the structure of IQC matrix M and filter Ψ.

Assumption 5.4.1. *The IQC decision variables $M : [0,T] \to \mathbb{S}^{n_z}$ for a specified IQC filter $\Psi : \mathcal{L}_2^{(n_v+n_w)}[0,T] \to \mathcal{L}_2^{n_z}[0,T]$ are assumed to have the following block diagonal*

structure

$$M(t) := \begin{bmatrix} M_v(t) & 0 \\ 0 & -M_w(t) \end{bmatrix}, \quad \Psi := \begin{bmatrix} \Psi_v & 0 \\ 0 & \Psi_w \end{bmatrix}$$

with constraints $M_v(t) \succ 0$ *and* $M_w(t) \succ 0$, $\forall t \in [0, T]$. *Moreover,* Ψ *has a feedthrough matrix* $D_\psi(t) := \begin{bmatrix} D_{\psi v}(t) & D_{\psi w}(t) \end{bmatrix} \in \mathbb{R}^{n_z \times (n_v + n_w)}$ *with full column rank* $\forall t \in [0, T]$.

This block diagonal assumption is made to simplify the notation. More general IQC multipliers are considered for (infinite horizon) synthesis in [91]. As discussed in Example 4.4.5, the IQC filter Ψ, is typically pre-specified by a collection of basis functions. In this case, the worst-case gain condition in Theorem 4.4.1 is a differential LMI in the variables M_v, M_w, P, and γ^2. The filter Ψ is, in general, non-square with $n_z \neq n_v + n_w$. The proposed synthesis method requires a non-unique factorization such that resulting factor is invertible square system i.e. $n_z = n_v + n_w$. The finite horizon factorization (Lemma B.1.1 in Appendix B.1) can be used to construct square invertible systems U_v and U_w such that,

$$\begin{aligned} \Psi_v^\sim M_v \Psi_v &= U_v^\sim U_v \\ \Psi_w^\sim M_w \Psi_v &= U_w^\sim U_w \end{aligned} \tag{5.5}$$

The assumption that feedthrough matrix $D_\psi(t)$ has full column rank is required for the existence of such factorization. This factorization is used in the proposed synthesis algorithm below to construct a scaled plant.

5.4.2 Proposed Algorithm

A high-level overview of the proposed iterative method is given in Algorithm 5. The uncertain finite horizon system is $\mathcal{F}_u(\mathcal{F}_l(G, K), \Delta)$ with G given by Equation (5.4) and Δ specified by uncertainty set $\mathcal{I}(\Psi, M)$. The robust synthesis algorithm is specified to run a given maximum number of iterations N_{syn}. It is initialized with scalings $U_v^{(0)} := I_{n_v}$ and $U_w^{(0)} := I_{n_w}$. There is also an initial performance scaling set to $\gamma_a^{(0)} := 1$.

The beginning of each iteration involves the construction of a scaled plant G_{scl} as shown in Figure 5.3. This step is described further in the next subsection. For now, it is sufficient to note that $G_{scl} = G$ on the first iteration due to the initialization choices.

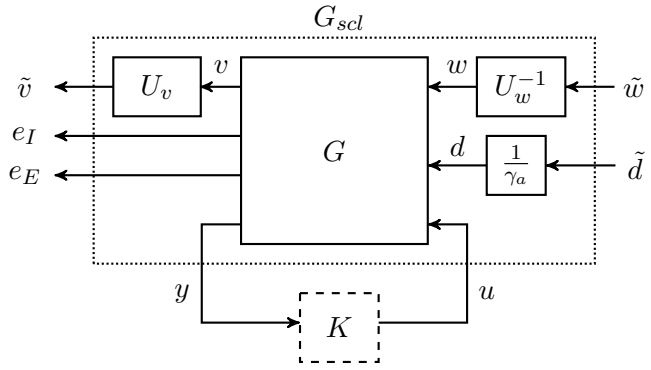

Figure 5.3: LTV Synthesis on Scaled Plant G_{scl}

The next step is to perform finite horizon nominal synthesis on the scaled plant. This step is performed using the synthesis results described previously in Section 5.3. This yields a controller $K^{(i)}$ and the achievable closed-loop performance $\gamma_s^{(i)}$. Each iteration concludes with an IQC analysis on the uncertain closed-loop of $N := \mathcal{F}_l(G, K^{(i)})$ and Δ as shown in Figure 5.2. This closed-loop uses the original (unscaled) plant G and the controller $K^{(i)}$ obtained from the nominal synthesis step. The worst-case gain upper bound $\gamma_a^{(i)}$ is computed using the algorithm in [10] as summarized in Section 4.4. This iterative algorithm requires additional initialization including number of analysis iterations N_{iter}, stopping tolerance tol, DLMI time grid t_{DLMI} and spline basis function time grid t_{sp}, which are not included in Algorithm 5. All subsequent iterations require the construction of a scaled plant using the IQC results. The construction of this scaled plant links together the nominal synthesis and IQC analysis steps. It is described further in Section 5.4.3. Algorithm 5 terminates after N_{syn} iterations. More sophisticated stopping conditions can be employed. For example, the iterations could be terminated if no significant improvement in worst-case gain is achieved. The algorithm returns the controller of order n_K that achieves the best (smallest) bound on the worst-case gain, where $n_K = n_G + n_\psi$.

5.4.3 Construction of a Scaled Plant

The scaled open loop plant $G_{scl}^{(i)}$ is constructed as shown in Figure 5.3 by scaling the performance channels and uncertainty channels of original open loop plant G using

Algorithm 5 Finite Horizon Robust Synthesis

1: **Given:** G
2: **Initialize:** N_{syn}, $U_v^{(0)} := I_{n_v}$, $U_w^{(0)} := I_{n_w}$, $\gamma_a^{(0)} := 1$
3: **for** $i = 1 : N_{syn}$ **do**
4: **Scaled Plant Construction (Section 5.4.3):** Construct a scaled plant $G_{scl}^{(i)}$ using G, $U_v^{(i-1)}$, $U_w^{(i-1)}$, $\gamma_a^{(i-1)}$.
Output: $G_{scl}^{(i)}$
5: **Nominal LTV Synthesis (Section 5.3):** Perform nominal controller synthesis on the scaled plant $G_{scl}^{(i)}$.
Output: $K^{(i)}$, $\gamma_s^{(i)}$
6: **IQC Analysis (Section 4.4):** Choose the basis functions for Ψ and perform worst-case gain iterations on $\mathcal{F}_u(N^{(i)}, \Delta)$ using iterative algorithm presented in [10] where $N^{(i)} := \mathcal{F}_l(G, K^{(i)})$ denotes the closed loop LTV system. Perform finite horizon factorization using the same Ψ and computed decision variables $M^{(i)}$ to compute the uncertainty channel scalings $U_v^{(i)}$ and $U_w^{(i)}$.
Output: $P^{(i)}$, $M^{(i)}$, $\gamma_a^{(i)}$, $U_v^{(i)}$, $U_w^{(i)}$
7: **end for**

$U_v^{(i-1)}$, $U_w^{(i-1)}$ and $\gamma_a^{(i-1)}$ obtained from the previous iteration. This scaling ensures appropriate normalization of the performance and uncertainty channels. This is a key step which integrates the nominal synthesis and worst-case gain problem. To simplify the notation, the superscripts $(i-1)$ will be dropped in the remainder of this subsection.

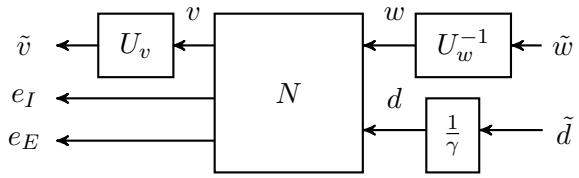

Figure 5.4: Scaled Plant N_{scl}

Let $N := \mathcal{F}_l(G, K)$ be the closed-loop (without uncertainty). For a given IQC filter Ψ an extended system N_{ext} similar to Figure 4.3 can be constructed. The next lemma gives a formal statement connecting robust performance of the extended system N_{ext}

to nominal performance of scaled system N_{scl} as shown in Figure 5.4. A proof of the following lemma uses a time-varying factorization of (Ψ, M) to construct U_v and U_w.

Lemma 5.4.1. *Let $\epsilon > 0$, $\gamma > 0$, $M_v(t) \succ 0$, $M_w(t) \succ 0$ and a differentiable function $P : [0,T] \to \mathbb{S}^n$ such that $P(T) \geq F$ be given with the choice of (Q,S,R,F) as in Equation (4.8). The following statements are equivalent:*

1. $DLMI_{Rob}(P, M, \gamma^2, t) \leq -\epsilon I, \quad \forall t \in [0,T]$.
2. $\|N_{scl}\|_{[0,T]} \leq 1 - \hat{\epsilon}$, *for some* $\hat{\epsilon}$.

Proof. (**1** $\Rightarrow$ **2**) This proof is presented in two parts. In part (i), we first show that $DLMI_{Rob}(P, M, \gamma^2, t) \leq -\epsilon I$, $\forall t \in [0,T]$ can equivalently be written as a dissipation inequality with only single valid IQC. Next, in part (ii), the state-space realization of the extended system N_{ext} and the scaled system N_{scl} are indeed the same, which allow us to rewrite the robust performance DLMI as a nominal performance DLMI for N_{scl}. Integrating the related dissipation inequality completes the proof.

Part (i): Define a storage function $V(x,t) := x^\top P(t) x$. Left and right multiply the DLMI (4.7) by $[x^\top, w^\top, d^\top]$ and its transpose to show that V satisfies the following dissipation inequality for all $t \in [0,T]$:

$$\dot{V} + \begin{bmatrix} x \\ \left[\begin{smallmatrix} w \\ d \end{smallmatrix}\right] \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} x \\ \left[\begin{smallmatrix} w \\ d \end{smallmatrix}\right] \end{bmatrix} + z^\top M z \leq -\epsilon\, d^\top d. \tag{5.6}$$

where $x = \left[\begin{smallmatrix} x_N \\ x_\psi \end{smallmatrix}\right] \in \mathbb{R}^n$ is the state of extended system as shown in Figure 4.3. Consider the outputs of the IQC filter $\Psi = \left[\begin{smallmatrix} \Psi_v & 0 \\ 0 & \Psi_w \end{smallmatrix}\right]$ be partitioned as $z := \left[\begin{smallmatrix} z_v \\ z_w \end{smallmatrix}\right]$. Let Ψ_v have the following state-space representation with state x_v, input v, and output z_v:

$$\begin{aligned} \dot{x}_v(t) &= A_1(t)\, x_v(t) + B_1(t)\, v(t) \\ z_v(t) &= C_1(t)\, x_v(t) + D_1(t)\, v(t) \end{aligned} \tag{5.7}$$

A similar time-varying state-space form also holds for Ψ_w with matrices (A_2, B_2, C_2, D_2),

state x_w, input w, and output z_w. Thus the term $z^\top M z$ in (5.6) can be expressed as:

$$\begin{aligned} z^\top M z &= z_v^\top M_v z_v - z_w^\top M_w z_w \\ &= \begin{bmatrix} x_v \\ v \end{bmatrix}^\top \begin{bmatrix} C_1^\top \\ D_1^\top \end{bmatrix} M_v \begin{bmatrix} C_1 & D_1 \end{bmatrix} \begin{bmatrix} x_v \\ v \end{bmatrix} - \begin{bmatrix} x_w \\ w \end{bmatrix}^\top \begin{bmatrix} C_2^\top \\ D_2^\top \end{bmatrix} M_w \begin{bmatrix} C_2 & D_2 \end{bmatrix} \begin{bmatrix} x_w \\ w \end{bmatrix} \end{aligned} \tag{5.8}$$

First, consider only the terms involving v and define the quadratic storage matrices as:

$$\begin{bmatrix} Q_v & S_v \\ S_v^\top & R_v \end{bmatrix} := \begin{bmatrix} C_1^\top \\ D_1^\top \end{bmatrix} M_v \begin{bmatrix} C_1 & D_1 \end{bmatrix} \tag{5.9}$$

By Lemma B.1.1 in Appendix B.1, the condition $M_v(t) \succ 0$, $\forall t \in [0, T]$ implies that there exists $X_v : [0, T] \to \mathbb{S}^{n_{x_v}}$ such that:

$$\dot{X}_v + A_1^\top X_v + X_v A_1 + Q_v - (X_v B_1 + S_v) R_v^{-1} (X_v B_1 + S_v)^\top = 0, \qquad X_v(T) = 0 \tag{5.10}$$

Moreover, Lemma B.1.1 in Appendix B.1 also implies that there exists a spectral factor U_v with a state-space realization as $(A_1, B_1, \tilde{C}_1, \tilde{D}_1)$ with $\tilde{C}_1 := W_v^{-T}(B_1^\top X_v + S_v^\top)$, $\tilde{D}_1 := W_v$ and $R_v = W_v^\top W_v$. Note that x_v is the state and $\tilde{v}$ is the output of the spectral factor U_v. The RDE (5.10) can be written in terms of the state-space matrices of U_v as:

$$Q_v = -\dot{X}_v - A_1^\top X_v - X_v A_1 + \tilde{C}_1^\top \tilde{C}_1 \tag{5.11}$$

Substitute above Q_v and $S_v^\top = \tilde{D}_1^\top \tilde{C}_1 - B_1^\top X_v$ in (5.9) to obtain the following expression:

$$\begin{aligned} z_v^\top M_v z_v = &- x_v \dot{X}_v x_v - (A_1 x_v + B_1 v)^\top X_v x_v - x_v^\top X_v (A_1 x_v + B_1 v) \dots \\ &+ (\tilde{C}_1 x_v + \tilde{D}_1 v)^\top (\tilde{C}_1 x_v + \tilde{D}_1 v) \end{aligned} \tag{5.12}$$

This can be simplified to the following expression:

$$z_v^\top M_v z_v = -x_v^\top \dot{X}_v x_v - \dot{x}_v^\top X_v x_v - x_v^\top X_v \dot{x}_v + \tilde{v}^\top \tilde{v} = -\frac{d}{dt}\left(x_v^\top X_v x_v \right) + \tilde{v}^\top \tilde{v} \tag{5.13}$$

Similarly, with $M_w(t) \succ 0$, $\forall t \in [0, T]$ the spectral factor U_w can be obtained with

a state-space realization $(A_2, B_2, \tilde{C}_2, \tilde{D}_2)$, states x_w, and outputs $\tilde{w}$. The following expression holds:

$$z_w^\top M_w z_w = -\frac{d}{dt}\left(x_w^\top X_w x_w\right) + \tilde{w}^\top \tilde{w} \tag{5.14}$$

where $X_w : [0,T] \to \mathbb{S}^{n_{x_w}}$ is a solution to a related RDE with respective quadratic storage matrices and the boundary condition $X_w(T) = 0$. Subtract Equation (5.14) from (5.13) to get the left hand side of the Equation (5.8) as follows:

$$z^\top M z = -\frac{d}{dt}\left(x_\psi^\top X x_\psi\right) + \tilde{v}^\top \tilde{v} - \tilde{w}^\top \tilde{w} \tag{5.15}$$

where $x_\psi = \left[\begin{smallmatrix} x_v \\ x_w \end{smallmatrix}\right]$, $X(t) := \begin{bmatrix} X_v(t) & 0 \\ 0 & -X_w(t) \end{bmatrix}$ and $X(T) = 0$. Let the modified matrix $\tilde{P}(t)$ be defined as follows:

$$\tilde{P}(t) := P(t) - \begin{bmatrix} 0 & 0 \\ 0 & X(t) \end{bmatrix} \tag{5.16}$$

This yields a modified storage function $\tilde{V}(x,t) := x^\top \tilde{P}(t) x$. The modified storage function has the form:

$$\tilde{V}(x,t) = V(x,t) - x_v^\top X_v x_v + x_w^\top X_w x_w \tag{5.17}$$

where the second and third term can be interpreted as hidden energy stored in the IQC multiplier. With modified storage function $\tilde{V}$ the dissipation inequality (5.6) can be recast as,

$$\dot{\tilde{V}} + \begin{bmatrix} x \\ \left[\begin{smallmatrix} w \\ d \end{smallmatrix}\right] \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} x \\ \left[\begin{smallmatrix} w \\ d \end{smallmatrix}\right] \end{bmatrix} + \begin{bmatrix} \tilde{v} \\ \tilde{w} \end{bmatrix}^\top J_{n_v, n_w} \begin{bmatrix} \tilde{v} \\ \tilde{w} \end{bmatrix} \leq -\epsilon\, d^\top d \tag{5.18}$$

This dissipation inequality is equivalent to (5.6) with a single IQC $\mathcal{I}(U, J_{n_v,n_w})$ where $U := \left[\begin{smallmatrix} U_v & 0 \\ 0 & U_w \end{smallmatrix}\right]$. Next, We show that $\mathcal{I}(U, J_{n_v,n_w})$ is a valid time-domain IQC. To see this, define $V_\psi(x_\psi(t), t) := x_\psi(t)^\top X(t) x_\psi(t)$, $\tilde{z}(t) := \begin{bmatrix} \tilde{v}(t) \\ \tilde{w}(t) \end{bmatrix}$ and integrate Equation (5.15)

both sides from 0 to T to obtain:

$$\int_0^T z^\top M z \; dt = -V_\psi(x_\psi(T), T) + V_\psi(x_\psi(0), 0) + \int_0^T \tilde{z}^\top J_{n_v, n_w} \tilde{z} \; dt \tag{5.19}$$

Note that $V_\psi(x_\psi(0), 0) = 0$ because $x_\psi(0) = 0$ and $V_\psi(x_\psi(T), T) = x_\psi(T)^\top X(T) x_\psi(T) = 0$ due to the boundary condition $X(T) = 0_{n_\psi}$ of the time-varying factorization RDE. Thus if $\int_0^T z^\top M z \; dt \geq 0$ then we have $\int_0^T \tilde{z}^\top J_{n_v, n_w} \tilde{z} \; dt \geq 0$. Finally, note that $\tilde{P}(t)$ satisfies the same boundary condition as $P(t)$ i.e. $\tilde{P}(T) \geq F$ because of the boundary condition $X(T) = 0$. Thus, $\tilde{V}(x, t)$ is a valid storage function $\forall t \in [0, T]$.

Part (ii): Let the extended system of N with spectral factor U be written in partitioned form as:

$$\left[\begin{array}{c} \dot{x} \\ \hline \tilde{v} \\ \tilde{w} \\ e_I \\ e_E \end{array}\right] = \left[\begin{array}{c|cc} \mathcal{A} & \mathcal{B}_w & \mathcal{B}_d \\ \hline \mathcal{C}_{\tilde{v}} & \mathcal{D}_{\tilde{v}w} & \mathcal{D}_{\tilde{v}d} \\ \mathcal{C}_{\tilde{w}} & \mathcal{D}_{\tilde{w}w} & 0 \\ \mathcal{C}_I & \mathcal{D}_{Iw} & \mathcal{D}_{Id} \\ \mathcal{C}_E & 0 & 0 \end{array}\right] \left[\begin{array}{c} x \\ \hline w \\ d \end{array}\right] \tag{5.20}$$

where $x = \left[\begin{smallmatrix} x_N \\ x_v \\ x_w \end{smallmatrix}\right] \in \mathbb{R}^n$ and state-space matrices:

$$\mathcal{A} := \begin{bmatrix} A_N & 0 & 0 \\ B_1 C_v & A_1 & 0 \\ 0 & 0 & A_2 \end{bmatrix}, \mathcal{B}_w := \begin{bmatrix} B_w \\ B_1 D_{vw} \\ B_2 \end{bmatrix}, \mathcal{B}_d := \begin{bmatrix} B_d \\ B_1 D_{vd} \\ 0 \end{bmatrix}$$
$$\mathcal{C}_{\tilde{v}} := \begin{bmatrix} \tilde{D}_1 C_v & \tilde{C}_1 & 0 \end{bmatrix}, \mathcal{C}_I := \begin{bmatrix} C_I & 0 & 0 \end{bmatrix}, \mathcal{C}_{\tilde{w}} := \begin{bmatrix} 0 & 0 & \tilde{C}_2 \end{bmatrix}$$
$$\mathcal{C}_E := \begin{bmatrix} C_E & 0 & 0 \end{bmatrix}, \mathcal{D}_{\tilde{v}w} := \tilde{D}_1 D_{vw}, \mathcal{D}_{\tilde{v}d} := \tilde{D}_1 D_{vd}$$
$$\mathcal{D}_{\tilde{w}w} := \tilde{D}_2, \mathcal{D}_{Iw} := D_{Iw}, \mathcal{D}_{Id} = D_{Id}$$

Using the choice of (Q, S, R) from Equation (4.8), the following partitioned DLMI is

equivalent to the dissipation inequality (5.18) for the state-space realization of (5.20).

$$\begin{bmatrix} \dot{\tilde{P}} + \mathcal{A}^\top \tilde{P} + \tilde{P}\mathcal{A} & \tilde{P}\mathcal{B}_w & \tilde{P}\mathcal{B}_d \\ \mathcal{B}_w^\top \tilde{P} & 0_{n_w} & 0 \\ \mathcal{B}_d^\top \tilde{P} & 0 & -\gamma^2 I_{n_d} \end{bmatrix} + \begin{bmatrix} \mathcal{C}_I^\top \\ \mathcal{D}_{Iw}^\top \\ \mathcal{D}_{Id}^\top \end{bmatrix} \begin{bmatrix} \mathcal{C}_I & \mathcal{D}_{Iw} & \mathcal{D}_{Id} \end{bmatrix} \cdots$$
$$+ \begin{bmatrix} \mathcal{C}_{\tilde{v}}^\top & \mathcal{C}_{\tilde{w}}^\top \\ \mathcal{D}_{\tilde{v}w}^\top & \mathcal{D}_{\tilde{w}w}^\top \\ \mathcal{D}_{\tilde{v}d} & 0 \end{bmatrix} J_{n_v, n_w} \begin{bmatrix} \mathcal{C}_{\tilde{v}} & \mathcal{D}_{\tilde{v}w} & \mathcal{D}_{\tilde{v}d} \\ \mathcal{C}_{\tilde{w}} & \mathcal{D}_{\tilde{w}w} & 0 \end{bmatrix} \leq -\epsilon I \tag{5.21}$$

The condition $M_w(t) \succ 0, \forall t \in [0, T]$ is sufficient to ensure that $\mathcal{D}_{\tilde{w}w} := \tilde{D}_2$ is nonsingular. The output equation for w can be written as: $w = \mathcal{D}_{\tilde{w}w}^{-1}(\tilde{w} - \mathcal{C}_{\tilde{w}} x)$. Use this relation to substitute for w in Equation (5.20). This gives the following scaled system:

$$\left[\begin{array}{c} \dot{x} \\ \hline \tilde{v} \\ e_I \\ e_E \end{array}\right] = \left[\begin{array}{c|cc} \mathcal{A} & \mathcal{B}_w & \mathcal{B}_d \\ \hline \mathcal{C}_{\tilde{v}} & \mathcal{D}_{\tilde{v}w} & \mathcal{D}_{\tilde{v}d} \\ \mathcal{C}_I & \mathcal{D}_{Iw} & \mathcal{D}_{Id} \\ \mathcal{C}_E & 0 & 0 \end{array}\right] L \left[\begin{array}{c} x \\ \hline \tilde{w} \\ d \end{array}\right] \tag{5.22}$$

where the nonsingular time-varying matrix L is defined as:

$$L := \begin{bmatrix} I_n & 0 & 0 \\ -\mathcal{D}_{\tilde{w}w}^{-1}\mathcal{C}_{\tilde{w}} & \mathcal{D}_{\tilde{w}w}^{-1} & 0 \\ 0 & 0 & I_{n_d} \end{bmatrix} \tag{5.23}$$

Equation (5.22) can be rewritten as follows:

$$\left[\begin{array}{c} \dot{x} \\ \hline \tilde{v} \\ e_I \\ e_E \end{array}\right] = \left[\begin{array}{c|cc} \tilde{\mathcal{A}} & \mathcal{B}_{\tilde{w}} & \mathcal{B}_d \\ \hline \tilde{\mathcal{C}}_{\tilde{v}} & \mathcal{D}_{\tilde{v}\tilde{w}} & \mathcal{D}_{\tilde{v}d} \\ \tilde{\mathcal{C}}_I & \mathcal{D}_{I\tilde{w}} & \mathcal{D}_{Id} \\ \mathcal{C}_E & 0 & 0 \end{array}\right] \left[\begin{array}{c} x \\ \hline \tilde{w} \\ d \end{array}\right] \tag{5.24}$$

where the updated state-space matrices are defined as:

$$\begin{aligned}
\tilde{\mathcal{A}} &:= \begin{bmatrix} A_N & 0 & -B_w\tilde{D}_2^{-1}\tilde{C}_2 \\ B_1C_v & A_1 & -B_1D_{vw}\tilde{D}_2^{-1}\tilde{C}_2 \\ 0 & 0 & A_2 - B_2\tilde{D}_2^{-1}\tilde{C}_2 \end{bmatrix}, \mathcal{B}_{\tilde{w}} := \begin{bmatrix} B_w\tilde{D}_2^{-1} \\ B_1D_{vw}\tilde{D}_2^{-1} \\ B_2\tilde{D}_2^{-1} \end{bmatrix} \\
\tilde{\mathcal{C}}_{\tilde{v}} &= \begin{bmatrix} \tilde{D}_1C_v & \tilde{C}_1 & -\tilde{D}_1D_{vw}\tilde{D}_2^{-1}\tilde{C}_2 \end{bmatrix}, \tilde{\mathcal{C}}_I := \begin{bmatrix} C_I & 0 & -D_{Iw}\tilde{D}_2^{-1}\tilde{C}_2 \end{bmatrix} \\
\mathcal{D}_{\tilde{v}\tilde{w}} &:= \tilde{D}_1D_{vw}\tilde{D}_2^{-1}, \mathcal{D}_{I\tilde{w}} := D_{Iw}\tilde{D}_2^{-1}
\end{aligned}$$

Note that the following state-space matrices of the inverse system of U_w shows up in the above representation.

$$U_w^{-1} := \left[\begin{array}{c|c} A_2 - B_2\tilde{D}_2^{-1}\tilde{C}_2 & B_2\tilde{D}_2^{-1} \\ \hline -\tilde{D}_2^{-1}\tilde{C}_2 & \tilde{D}_2^{-1} \end{array}\right] \tag{5.25}$$

Let scaled signal $\tilde{d} := \gamma\, d$ and state-space matrices ($\mathcal{A}_{scl}$, $\mathcal{B}_{scl}$, $\mathcal{C}_{scl}$, $\mathcal{D}_{scl}$) be defined as follows:

$$\begin{aligned}
&\mathcal{A}_{scl} := \tilde{\mathcal{A}}, \quad \mathcal{B}_{scl} := \begin{bmatrix} \mathcal{B}_{\tilde{w}} & \gamma^{-1}\mathcal{B}_d \end{bmatrix}, \quad \mathcal{C}_{scl} := \begin{bmatrix} \tilde{\mathcal{C}}_{\tilde{v}} \\ \tilde{\mathcal{C}}_I \end{bmatrix}, \\
&\mathcal{D}_{scl} := \begin{bmatrix} \tilde{D}_1D_{vw}\tilde{D}_2^{-1} & \gamma^{-1}\tilde{D}_1D_{vd} \\ D_{Iw}\tilde{D}_2^{-1} & \gamma^{-1}D_{Id} \end{bmatrix}
\end{aligned} \tag{5.26}$$

It is readily verified that, with above definition, the scaled plant N_{scl} has a state-space realization as follows:

$$\left[\begin{array}{c} \dot{x} \\ \hline \begin{bmatrix} \tilde{v} \\ e_I \end{bmatrix} \\ \hline e_E \end{array}\right] = \left[\begin{array}{c|c} \mathcal{A}_{scl} & \mathcal{B}_{scl} \\ \hline \mathcal{C}_{scl} & \mathcal{D}_{scl} \\ \mathcal{C}_E & 0 \end{array}\right] \left[\begin{array}{c} x \\ \hline \begin{bmatrix} \tilde{w} \\ \tilde{d} \end{bmatrix} \end{array}\right] \tag{5.27}$$

Perform the congruence transformation by multiplying the DLMI (5.21) on the left/right

by $L^\top/L$ to get:

$$\begin{bmatrix} \dot{\tilde{P}} + \tilde{\mathcal{A}}^\top \tilde{P} + \tilde{P}\tilde{\mathcal{A}} & \tilde{P}\mathcal{B}_{\tilde{w}} & \tilde{P}\mathcal{B}_d \\ \mathcal{B}_{\tilde{w}}^\top \tilde{P} & 0_{n_{\tilde{w}}} & 0 \\ \mathcal{B}_d^\top \tilde{P} & 0 & -\gamma^2 I_{n_d} \end{bmatrix} + \begin{bmatrix} \tilde{\mathcal{C}}_I^\top \\ \mathcal{D}_{I\tilde{w}}^\top \\ \mathcal{D}_{Id}^\top \end{bmatrix} \begin{bmatrix} \tilde{\mathcal{C}}_I & \mathcal{D}_{I\tilde{w}} & \mathcal{D}_{Id} \end{bmatrix} \cdots \\ + \begin{bmatrix} \tilde{\mathcal{C}}_{\tilde{v}}^\top & 0 \\ \mathcal{D}_{\tilde{v}\tilde{w}}^\top & I_{n_{\tilde{w}}} \\ \mathcal{D}_{\tilde{v}d}^\top & 0 \end{bmatrix} J_{n_v, n_w} \begin{bmatrix} \tilde{\mathcal{C}}_{\tilde{v}} & \mathcal{D}_{\tilde{v}\tilde{w}} & \mathcal{D}_{\tilde{v}d} \\ 0 & I_{n_{\tilde{w}}} & 0 \end{bmatrix} \leq -\epsilon I \tag{5.28}$$

This DLMI can also be written in more compact notation using the state matrices of N_{scl}. Multiply inequality (5.28) left and right by $[x^\top, \tilde{w}^\top, d^\top]$ and its transpose to show that $\tilde{V}(x(t), t) = x(t)^\top \tilde{P}(t) x(t)$ satisfies the following dissipation inequality:

$$\dot{\tilde{V}} + e_I^\top e_I - \gamma^2 d^\top d + \tilde{v}^\top \tilde{v} - \tilde{w}^\top \tilde{w} \leq -\epsilon\, d^\top d \tag{5.29}$$

Define $\tilde{d} := \gamma\, d$, $\tilde{\epsilon} := \epsilon\, \gamma^{-2}$ and combine the inputs $\tilde{w}$, $\tilde{d}$ together to rewrite the inequality (5.29) as follows:

$$\dot{\tilde{V}} + \begin{bmatrix} \tilde{v} \\ e_I \end{bmatrix}^\top \begin{bmatrix} \tilde{v} \\ e_I \end{bmatrix} - \begin{bmatrix} \tilde{w} \\ \tilde{d} \end{bmatrix}^\top \begin{bmatrix} \tilde{w} \\ \tilde{d} \end{bmatrix} \leq -\tilde{\epsilon}\, \tilde{d}^\top \tilde{d} \tag{5.30}$$

Integrate over $[0, T]$ to obtain the following dissipation inequality:

$$\tilde{V}(x(T), T) - \tilde{V}(x(0), 0) + \| \left[\begin{smallmatrix} \tilde{v} \\ e_I \end{smallmatrix}\right] \|_{2,[0,T]}^2 - \| \left[\begin{smallmatrix} \tilde{w} \\ \tilde{d} \end{smallmatrix}\right] \|_{2,[0,T]}^2 \leq -\tilde{\epsilon} \|\tilde{d}\|_{2,[0,T]}^2 \tag{5.31}$$

Note that $\tilde{V}(x(0), 0) = 0$ as $x(0) = 0$ and $\tilde{V}(x(T), T) = x(T)^\top \tilde{P}(T) x(T)$ with boundary condition $\tilde{P}(T) \geq F$ as shown earlier. Apply this boundary condition in inequality (5.31) with the definition of F from Equation (4.8) to conclude:

$$\|e_E(T)\|_2^2 + \| \left[\begin{smallmatrix} \tilde{v} \\ e_I \end{smallmatrix}\right] \|_{2,[0,T]}^2 \leq \| \left[\begin{smallmatrix} \tilde{w} \\ \tilde{d} \end{smallmatrix}\right] \|_{2,[0,T]}^2 - \tilde{\epsilon} \|\tilde{d}\|_{2,[0,T]}^2 \tag{5.32}$$

Divide both sides by $\| \left[\begin{smallmatrix} \tilde{w} \\ \tilde{d} \end{smallmatrix}\right] \|_{2,[0,T]}^2 < \infty$ and define $\hat{\epsilon} := \tilde{\epsilon} \|\tilde{d}\|_{2,[0,T]}^2 / \| \left[\begin{smallmatrix} \tilde{w} \\ \tilde{d} \end{smallmatrix}\right] \|_{2,[0,T]}^2$ to show that $\|N_{scl}\|_{[0,T]} \leq 1 - \hat{\epsilon}$. This proof can be worked backwards to prove ($\mathbf{2} \Rightarrow \mathbf{1}$). $\square$

Remark 5.4.1. *If the IQC decision variables M and the state-space matrices of Ψ are constant on the given time horizon, then for sufficiently large horizon T, the RDE solution for the finite horizon factorization converges to that of the steady-state Algebraic Riccati Equation (ARE). As a result, the state-space realization of the finite horizon factorization converges to that of an infinite horizon LTI spectral factorization. However, the ARE solution X for infinite horizon spectral factorization is sign indefinite, and it fails to satisfy the terminal boundary condition on $\tilde{P}$. Thus, it is essential to note in the above proof that to meet the boundary condition on storage function, one must use finite horizon factorization.*

The above lemma states that the extended system given by Equation (4.6) satisfies the robust performance condition (4.9) if and only if the scaled system has nominal performance less than 1. Note that Figure 5.4 considered the scaling of $1/\gamma_a$ on the input side of the performance channel. However, for better numerical conditioning, the scaling of $1/\sqrt{\gamma_a}$ can be considered on both input and output performance channels.

5.4.4 Main Theorem

The plant $G_{scl}^{(1)} = G$ for the robust synthesis may include the uncertainty and performance channel design weights as in standard robust control workflow [21, 22]. Depending on the requirement, these weights can be static, dynamic, and/or time-varying. Typically, multiple design iterations are performed to tune these weights and yield an acceptable trade-off between robustness and performance. Note that the first nominal LTV synthesis step in Algorithm 5 may not yield a finite performance $\gamma_s^{(1)}$. For example, if the uncertainty level is too high then the RDEs for nominal synthesis of $G_{scl}^{(1)} = G$ may not have a solution on $[0, T]$ for any finite $\gamma_s^{(1)}$. However, in this case, finite performance can be achieved by reducing the uncertainty level and restarting the iteration. Next, the main theorem is presented with a technical assumption that the first nominal synthesis yields a finite performance.

Theorem 5.4.1. *If the first nominal synthesis step yields a finite performance* $\gamma_s^{(1)}$ *then all the subsequent iterations are well-posed at each step and worst-case gain is non-increasing, i.e.*

$$\gamma_a^{(i+1)} \leq \gamma_a^{(i)} \quad \forall i \geq 1$$

Proof. The first iteration ($i = 1$) is different from the subsequent one. Due to initialization choices $G_{scl}^{(1)} = G$. The synthesis step is performed with no modifications and yields a controller $K^{(1)}$ that guarantees the closed loop performance of $\gamma_s^{(1)}$. By assumption, we have $\gamma_s^{(1)} < \infty$. The IQC analysis step performed on the closed loop $N^{(1)} := \mathcal{F}_l(G, K^{(1)})$ uncertain plant then achieves a finite horizon worst-case gain upper bound of $\gamma_a^{(1)} < \infty$. Thus, the first iteration is well-posed.

All subsequent iterations ($i > 1$) begin with the iteration count update in the for loop. The IQC analysis step from previous iteration shows that there exists ($P^{(i-1)}$, $M^{(i-1)}$, $\gamma_a^{(i-1)}$) for a chosen Ψ that satisfies DLMI (4.7). This implies that the finite horizon factorization exists and multipliers $U_v^{(i-1)}$ and $U_w^{(i-1)}$ can be obtained using Lemma B.1.1 in Appendix B.1. Using these multipliers and worst-case gain $\gamma_a^{(i-1)}$, scaled plant similar to Figure 5.4 can be constructed. By Lemma 5.4.1, this scaled plant satisfies nominal performance < 1. Removing the controller yields the scaled open-loop plant $G_{scl}^{(i)}$. Thus, the construction of a scaled open-loop plant as shown in Figure 5.3 is well-defined. The synthesis step performed on $G_{scl}^{(i)}$ optimizes over all time-varying finite horizon controllers to yield a new controller $K^{(i)}$ that guarantees performance $\gamma_s^{(i)} < 1$. This new controller $K^{(i)}$ yields better nominal performance than the previous controller $K^{(i-1)}$ when used with the unscaled plant G. Thus, the closed loop $N^{(i)} := \mathcal{F}_u(G, K^{(i)})$ must satisfy the nominal performance < 1 when using $\gamma_a^{(i-1)}$. Lemma 5.4.1 can be used backwards in the next analysis step of $N^{(i)}$. Specifically, the closed loop with unscaled plant G and $K^{(i)}$ satisfies the DLMI analysis condition with ($P^{(i-1)}$, $M^{(i-1)}$, $\gamma_a^{(i-1)}$). Further, analysis step on $N^{(i)} := \mathcal{F}_l(G, K^{(i)})$ optimizes over all feasible P and M. This yields a worst-case gain $\gamma_a^{(i)}$ no greater than the previous step $\gamma_a^{(i-1)}$. Thus $\forall i \geq 1$, we have $\gamma_a^{(i+1)} \leq \gamma_a^{(i)}$. □

5.5 Numerical Examples

5.5.1 Comparison of Finite and Infinite Horizon Robust Synthesis

Consider a first order linear time-invariant (LTI) system G with the following dynamics:

$$\dot{x}(t) = 0.5\,x(t) + u(t) + w(t) + d_{in}(t) \tag{5.33}$$

$$v(t) = u(t) + d_{in}(t) \tag{5.34}$$

$$e_I(t) = \begin{bmatrix} x(t) \\ 0.2\,u(t) \end{bmatrix} \tag{5.35}$$

$$y(t) = x(t) + 0.01\,n(t) \tag{5.36}$$

where the performance output is $e_I(t) \in \mathbb{R}^2$ and measurement output is $y(t) \in \mathbb{R}$. The generalized disturbance input is $d(t) := \left[\begin{smallmatrix} d_{in}(t) \\ n(t) \end{smallmatrix}\right]$ where $d_{in}(t) \in \mathbb{R}$ is an external disturbance input and $n(t) \in \mathbb{R}$ is a measurement noise. Δ is a norm bounded, time-varying, nonlinear uncertainty with norm bound $\beta = 0.6$. The goal is to design a measurement feedback controller K that minimizes the worst-case induced $\mathcal{L}_2$-gain from disturbance d to output e_I.

First, an infinite horizon robust synthesis is performed to minimize the worst-case gain of the closed-loop system. This is achieved by bisection on the gain γ for a scaled plant until the robust performance is equal to 1. At each bisection step, MATLAB's `musyn` function is called to optimize the structured singular value upper bound $\bar{\mu}$. This function uses an iterative control design process (DK-iteration) to optimize the $\bar{\mu}$ of the closed-loop system. The infinite horizon worst-case induced $\mathcal{L}_2$-gain for the designed robust controller is 0.0563. Next, finite horizon robust synthesis is performed on a relatively long horizon (i.e., $T = 300$ seconds) using the method proposed in this chapter. The closed-loop worst-case gain achieved by a finite horizon time-varying controller is computed as 0.0560.

These simple comparison results show that the worst-case gain achieved using the finite horizon controller (on a relatively long horizon) approaches a value attained by an infinite horizon controller. Note that the proposed method uses a purely time-domain approach, whereas the μ-synthesis method uses the frequency gridding approach to approximate $\bar{\mu}$ and the associated D-scales. Thus, the close agreement between the two

worst-case gains on this example may not generally hold.

5.5.2 LTI Example: Mass-Spring-Damper System

This section uses Algorithm 5 to design a finite horizon LTV robust controller. The formulation for the control design is shown in Figure 5.5. The nominal plant H is a mass-spring-damper system on a horizon of $[0, T]$ where $T = 3$ second. The dynamics of H are given by:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} F(t) \tag{5.37}$$

where $F \in \mathbb{R}$ is the input force (N) and the state $x \in \mathbb{R}^2$ is given by the position x_1 (m) and the velocity x_2 (m/s). The outputs of H are the states $e_E := x \in \mathbb{R}^2$ and the position x_1. The position measurement is corrupted by noise n and fed to the controller as $y := x_1 + n$. The controller generates a commanded force u (N) which is corrupted by an input disturbance d_{in} and input model uncertainty Δ. The model parameters used for this example are given by $m = 1\ kg$, $k = 1\ N/m$, and $b = 0.8\ Ns/m$.

The plant input perturbation Δ is a SISO norm-bounded operator with $\|\Delta\|_{2\to 2,[0,T]} \leq 0.6$. This corresponds to the uncertainty set $\mathbf{\Delta}$ with $\beta = 0.6$ as discussed in Example 4.4.1. The synthesis objective is to minimize the closed-loop, worst-case gain from the generalized disturbance $\tilde{d} := \begin{bmatrix} \tilde{d}_{in} \\ \tilde{n} \end{bmatrix}$ to the generalized error $\tilde{e} := \begin{bmatrix} \tilde{u} \\ \tilde{e}_E \end{bmatrix}$. The weighted

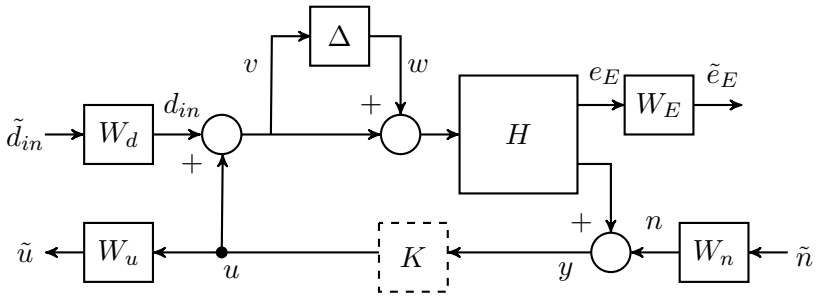

Figure 5.5: Uncertain Output-Feedback Interconnection

control effort $\tilde{u}$ is penalized in an $\mathcal{L}_2$ sense while $\tilde{e}_E$ is penalized with a terminal Euclidean norm at $T = 3$ second. The following design weights are chosen for the synthesis:

$$W_d = 0.1, \quad W_n = 0.01, \quad W_u = 1, \quad W_E = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right]$$

Let $\tilde{G}$ denote this weighted design interconnection for robust synthesis. It can be expressed in state-space form as in Equation (5.4). Algorithm 5 is run with $N_{syn} = 20$ iterations on a standard desktop computer with intel core i7 processor, 3 GHz CPU and 16 GB RAM. The IQC analysis step is performed using the approach in [10] with $tol = 5 \times 10^{-3}$, $N_{iter} = 10$, t_{DLMI} as 25 and τ_{sp} as 10 evenly spaced grid points on the horizon $[0, T]$.

Let $K_{0.6}$ denote the controller obtained at the end of the robust synthesis algorithm. This controller achieves the closed-loop worst-case performance of $\gamma_{wc} = 0.174$. It took 48.2 minutes to complete the 20 iterations. A nominal synthesis with $\Delta = 0$ was performed using the approach in Section 5.3. This controller, denoted K_0, achieves a closed-loop nominal performance of $\gamma_0 = 0.113$. It took 7.26 seconds to perform this nominal synthesis. The corresponding uncertain closed-loops with the nominal and robust controllers are denoted by $\tilde{T}_0 := \mathcal{F}_u(\mathcal{F}_l(\tilde{G}, K_0), \Delta)$ and $\tilde{T}_{0.6} := \mathcal{F}_u(\mathcal{F}_l(\tilde{G}, K_{0.6}), \Delta)$.

Figure 5.6 shows the worst-case performance versus the uncertainty level β for the uncertain closed-loops with these two controllers. The curve for $\tilde{T}_0$ (blue circles) has $\gamma = 0.113$ at $\beta = 0$ as reported above. The curve for $\tilde{T}_{0.6}$ (red squares) has $\gamma = 0.174$ at $\beta = 0.6$ as also reported above. This figure reveals the typical trade-off between performance and robustness. The nominal controller K_0 achieves better nominal performance ($\beta = 0$) than $K_{0.6}$. However, $K_{0.6}$ is more robust to higher levels of uncertainties. The primary design goal was to tightly bound the states at the final time $T = 3$ seconds. To study this further consider the impact of disturbance d_{in} on the Euclidean outputs e_E. Let G denote the unweighted plant which has the same inputs/outputs as the weighted plant $\tilde{G}$ but with all weights set to identity. Further, let the respective uncertain interconnection using G be denoted as $T_0 := \mathcal{F}_u(\mathcal{F}_l(G, K_0), \Delta)$ and $T_{0.6} := \mathcal{F}_u(\mathcal{F}_l(G, K_{0.6}), \Delta)$. A worst-case terminal Euclidean norm bound is computed for $T_{0(d_{in} \to e_E)}$ and $T_{0.6(d_{in} \to e_E)}$ at the uncertainty level $\beta = 0.6$. The corresponding upper bound using the algorithm in [10] was obtained as 0.799 and 0.723, respectively. This shows approximately a 9.6 %

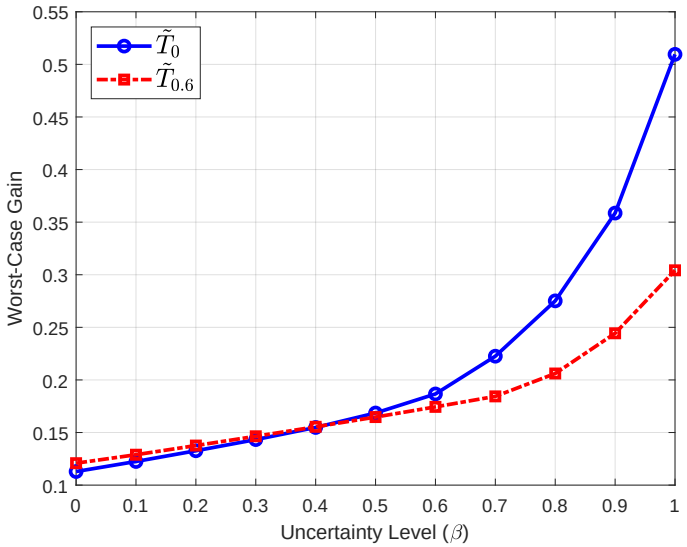

Figure 5.6: Worst-Case Gain Comparison

reduction in terminal Euclidean gain bound. As graphical illustration, these bounds are shown in Figure 5.7 as a disk in phase plane centered at the origin (initial condition). The bound accounts for all the disturbances d_{in} that satisfy $\|d_{in}\|_{2,[0,T]} \leq 1$ and all the uncertainties $\Delta \in \mathbf{\Delta}$ with $\beta = 0.6$. The corresponding dot-dashed line in the phase plane represents a combined response to a worst-case disturbance d_{1wc} and a candidate worst-case uncertainty Δ_{wc}. The worst-case disturbances for both interconnections are computed by solving the two-point boundary value problem as presented in [30]. The uncertainties are sampled randomly as first order linear time-invariant (LTI) systems.

The worst-case gain lower bounds corresponding to these perturbations are obtained as 0.786 and 0.715, respectively. A combination of worst-case disturbance and uncertainty pushes the states of the closed-loop system as far as the lower bound of the worst-case gain. Overall, these simple comparison results show that closed-loop worst-case Euclidean gain obtained through $K_{0.6}$ is smaller than K_0 at modeled uncertainty level. The differences are more significant at a higher uncertainty level.

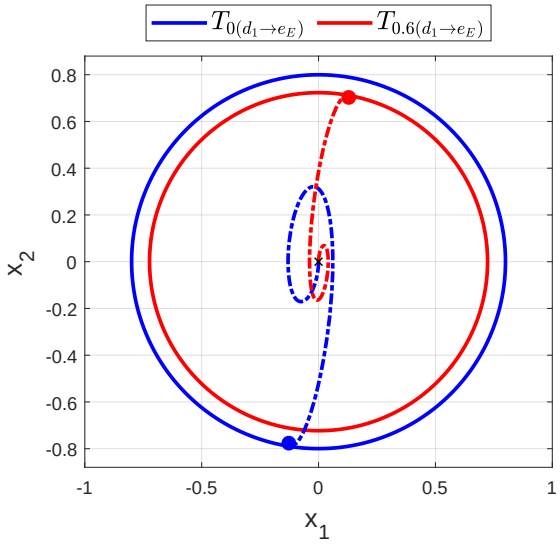

Figure 5.7: Robust Analysis ($\beta = 0.6$)

5.5.3 Nonlinear System: Two-Link Robot

We revisit an example of a two-link robot arm from Section 2.6. We denote the input disturbance as d_{in} and the linearized LTV system as H. An uncertain output feedback weighted interconnection of H is shown in the Figure 5.8. Let $\delta\theta := \begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \end{bmatrix}$ represent first-order perturbations in angular positions, which is the output of interest e_E. The measurement is also $\delta\theta$ but corrupted by noise $n = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$ and is fed to the controller as $y = \delta\theta + n$. The controller generates a commanded torque $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ is corrupted by input disturbance $d_{in} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$. The second control channel gets further corrupted by LTI input uncertainty Δ. The plant input perturbation Δ is a SISO LTI system with $\|\Delta\|_\infty \leq \beta$ where uncertainty level $\beta = 0.8$. This corresponds to the uncertainty set as discussed in Example 4.4.3. The synthesis objective is to minimize the closed-loop, worst-case gain from the generalized disturbance $\tilde{d} := \begin{bmatrix} \tilde{d}_{in} \\ \tilde{n} \end{bmatrix} = \begin{bmatrix} \tilde{d}_1 & \tilde{d}_2 & \tilde{n}_1 & \tilde{n}_2 \end{bmatrix}^\top$ to the generalized error $\tilde{e} := \begin{bmatrix} \tilde{u} \\ \tilde{e}_E \end{bmatrix}$. The weighted control effort $\tilde{u}$ is penalized in an $\mathcal{L}_2[0, T]$ sense while $\tilde{e}_E$ is penalized with a terminal Euclidean norm at $T = 5$ second. Let $I_2 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The

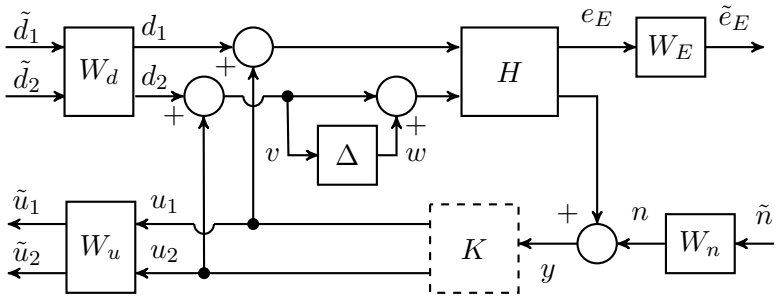

Figure 5.8: Uncertain Output Feedback Weighted Interconnection

following (constant) design weights are chosen for the performance channels:

$$W_d = 0.1\, I_2, \quad W_n = 0.01\, I_2, \quad W_u = 0.5\, I_2, \quad W_E = I_2$$

The design weight associated with the uncertainty channels are not considered in this example, however, in general, the weights W_v and W_w can also be used for the respective uncertainty channels. As noted earlier, these design weights can be dynamic and/or time-varying. Let $\tilde{G}$ denote this weighted design interconnection for robust synthesis. It can be expressed in state-space form as in Equation (5.4). Algorithm 5 is run with $N_{syn} = 7$ iterations. No significant improvement is obtained after 7^{th} iteration. The IQC analysis step is performed based on the approach in [10] and using parameterization similar to Example 4.4.5 with $p = 10$, $q = 1$, $tol = 5 \times 10^{-3}$, $N_{iter} = 10$, t_{DLMI} as 20 and τ_{sp} as 10 evenly spaced grid points on the horizon $[0, 5]$ seconds.

Let $K_{0.8}$ denote the controller obtained at the end of the robust synthesis algorithm. This controller achieves the closed-loop worst-case performance of $\gamma_{0.8} = 0.126$. It took 11.7 hours to complete the 7 iterations on a standard desktop computer with a 3 GHz Core i7 processor. In addition, a nominal synthesis with $\Delta = 0$ was performed using the approach in Section 5.3. This controller, denoted as K_0, achieves a closed-loop nominal performance of $\gamma_0 = 0.089$. It took 49.8 seconds to perform this nominal synthesis. The corresponding uncertain closed-loops with the nominal and robust controllers are denoted by $\tilde{T}_0 := \mathcal{F}_u(\mathcal{F}_l(\tilde{G}, K_0), \Delta)$ and $\tilde{T}_{0.8} := \mathcal{F}_u(\mathcal{F}_l(\tilde{G}, K_{0.8}), \Delta)$. Figure 5.9 shows the worst-case performance versus the uncertainty level β for the uncertain closed-loops with these two controllers. The curve for $\tilde{T}_0$ (blue circles) has $\gamma = 0.089$ at $\beta = 0$

as reported above. The curve for $\tilde{T}_{0.8}$ (red squares) has $\gamma = 0.126$ at $\beta = 0.8$ as also reported above. This figure reveals the typical trade-off between performance and robustness. The nominal controller K_0 achieves better nominal performance ($\beta = 0$) than $K_{0.8}$. However, $K_{0.8}$ is more robust to higher levels of uncertainties.

Note that each data point in Figure 5.9 represents a worst-case gain induced from the generalized disturbance input $\tilde{d}$ to the generalized error output $\tilde{e}$. Both $\tilde{d}$ and $\tilde{e}$ have two components which can further be analyzed using induced gain for individual input-output pairs. Table 5.1 shows an analysis with no uncertainty (level $\beta = 0$) performed for the closed-loops with nominal control design $\tilde{T}_0$ (blue) and robust control design $\tilde{T}_{0.8}$ (red). It is evident that the induced gain from noise $\tilde{n}$ to control effort $\tilde{u}$ dominates the overall performance for both the interconnections. Moreover, the induced $\mathcal{L}_2$-gain from $\tilde{n}$ to $\tilde{u}$ for $\tilde{T}_{0.8}$ is 0.124 which is approximately 41% higher than the corresponding value for $\tilde{T}_0$ ($= 0.088$). Likewise, the induced $\mathcal{L}_2$-to-Euclidean gain from disturbance $\tilde{d}_{in}$ to $\tilde{e}_E(T)$ is approximately 16.6% higher for $\tilde{T}_{0.8}$ ($= 0.077$) as compared to $\tilde{T}_0$ ($= 0.066$). The combined effect of the disturbance and noise is responsible for performance degradation of the robust controller at $\beta = 0$. Table 5.2 shows the worst-case gain upper bounds for robust analysis performed at an uncertainty level $\beta = 0.8$ for both interconnections. Note that the closed-loop with robust controller has the same worst-case induced $\mathcal{L}_2$-gain of 0.124 from $\tilde{n}$ to $\tilde{u}$ as in Table 5.1. Since, the robust controller explicitly accounts for model uncertainty, it has approximately 37.8% lower worst-case induced $\mathcal{L}_2$-gain from $\tilde{d}_{in}$ to $\tilde{u}$ compared to the nominal controller. Similarly, the robust controller performs better in terms of bounding the Euclidean outputs as compared to the nominal controller at $\beta = 0.8$. Overall, the disturbance rejection property for the nominal controller is degraded more from Table 5.1 to Table 5.2 as compared to the robust controller. This observation is consistent with known frequency domain insights for infinite horizon LTI systems, as the high frequency noise rejection properties are typically less impacted by model uncertainties than the low frequency disturbance rejection properties.

As noted earlier, the primary design goal was to tightly bound the states at the final time $T = 5$ seconds. To study this further consider the impact of link joint disturbance d_{in} on the Euclidean output e_E. Let G denote the unweighted plant which has the same inputs/outputs as the weighted plant $\tilde{G}$ but with all weights set to identity. Further, let

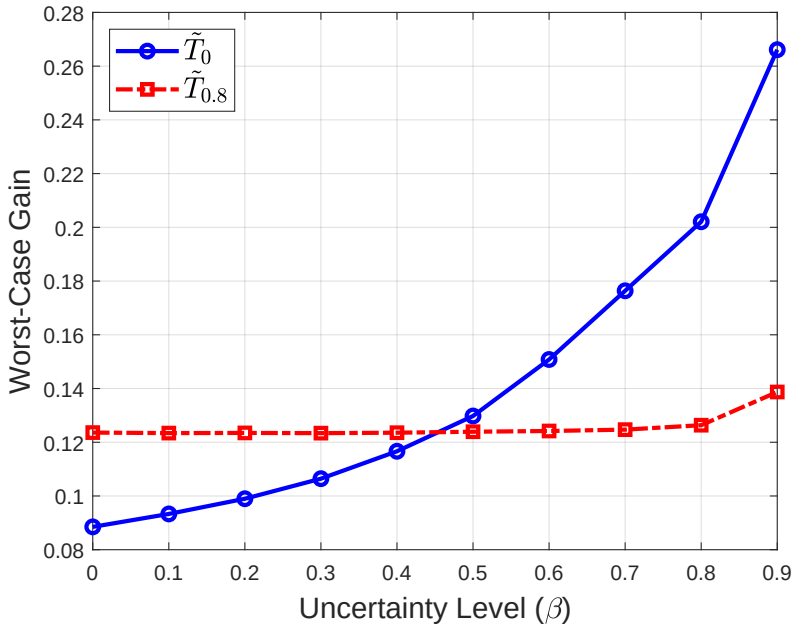

Figure 5.9: Worst-Case Gain Comparison

$\tilde{T}_0$ / $\tilde{T}_{0.8}$	**Disturbance** $\tilde{d}_{in}$	**Noise** $\tilde{n}$	**Generalized** $\tilde{d}$
Euclidean output $\tilde{e}_E(T)$	0.066 / 0.077	0.050 / 0.047	0.082 / 0.081
Control effort $\tilde{u}$	0.085 / 0.087	0.088 / 0.124	0.089 / 0.124
Generalized error $\tilde{e}$	0.086 / 0.097	0.088 / 0.124	0.089 / 0.124

Table 5.1: Induced Gain Upper Bounds (Nominal Analysis, $\beta = 0$)

the respective uncertain interconnection using G be denoted as $T_0 := \mathcal{F}_u(\mathcal{F}_l(G, K_0), \Delta)$ and $T_{0.8} := \mathcal{F}_u(\mathcal{F}_l(G, K_{0.8}), \Delta)$. Nominal analysis performed for both the $T_{0\,(d_{in}\to e_E)}$ and $T_{0.8\,(d_{in}\to e_E)}$ interconnections gives both upper and lower bounds on the nominal performance. The upper bounds are obtained as 0.656 and 0.766 respectively, which are shown as blue and red disk in Figure 5.10 at the final time. The corresponding lower bounds are obtained as 0.648 and 0.763. The worst-case disturbance $\|d_{in}\|_{2,[0,T]} \leq 0.5$ for both interconnections are computed by solving the two point boundary value problem as presented in [30]. These specific bad disturbances (Figure 5.11) pushes the state trajectory (dashed line) as far as the computed lower bound in the LTV simulation.

$\tilde{T}_0$ / $\tilde{T}_{0.8}$	**Disturbance** $\tilde{d}_{in}$	**Noise** $\tilde{n}$	**Generalized** $\tilde{d}$
Euclidean output $\tilde{e}_E(T)$	0.102 / 0.093	0.068 / 0.061	0.120 / 0.120
Control effort $\tilde{u}$	0.190 / 0.118	0.098 / 0.124	0.202 / 0.126
Generalized error $\tilde{e}$	0.190 / 0.118	0.106 / 0.125	0.202 / 0.126

Table 5.2: Worst-Case Gain Upper Bounds (Robust Analysis, $\beta = 0.8$)

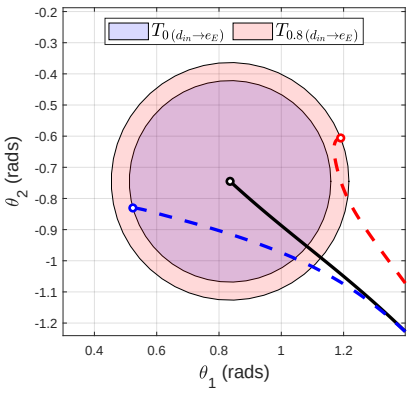

Figure 5.10: Nominal Analysis ($\beta = 0$)

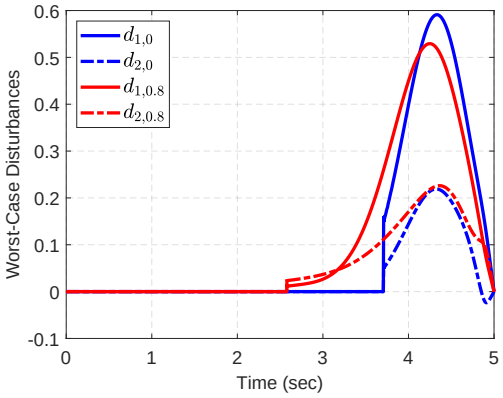

Figure 5.11: Worst-Case Disturbances

A worst-case terminal Euclidean norm bound is computed for both the interconnections at the uncertainty level $\beta = 0.8$. The corresponding upper bound using the algorithm in [10] was obtained as 1.02 and 0.93, respectively. This shows approximately an 8.82% reduction in the Euclidean norm bound. As a graphical illustration, these bounds are depicted in Figure 5.12 as a disk at the final time $T = 5$ seconds. The bound accounts for all the disturbances d_{in} that satisfy $\|d_{in}\|_{2,[0,T]} \leq 0.5$ and all the LTI uncertainties Δ with norm bound $\beta = 0.8$.

To obtain a reasonable lower bound on the worst-case gain, first 100 uncertainties are sampled randomly as first-order LTI systems with at most size 0.8. Then, uncertainty block Δ was replaced with each of sampled uncertainties, and nominal LTV analysis was performed from d_{in} to e_E for both the interconnections. Worst-case uncertainties are then obtained after maximizing performance over the sample space. Let, the specific bad perturbation that yields to the poor performance for both $T_{0\,(d_{in}\to e_E)}$ and $T_{0.8\,(d_{in}\to e_E)}$

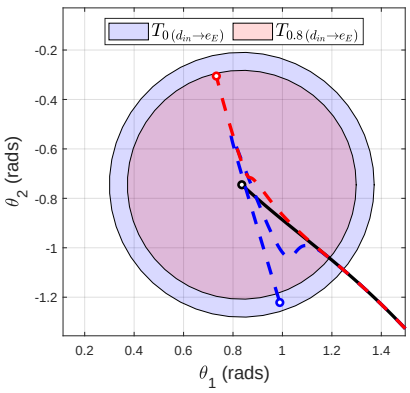

Figure 5.12: Robust Analysis ($\beta = 0.8$)

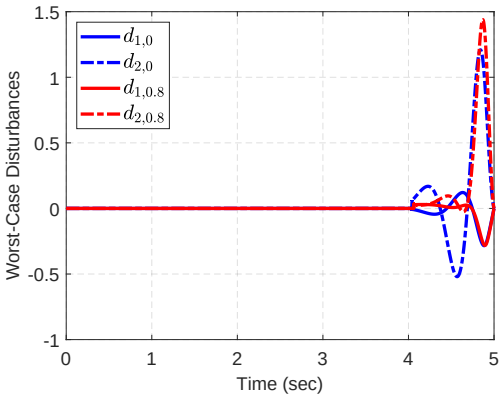

Figure 5.13: Worst-Case Disturbances

be denoted as Δ_{wc_1} and Δ_{wc_2} respectively.

$$\Delta_{wc_1} = \frac{-0.8\,s + 12.18}{s + 15.23}, \qquad \Delta_{wc_2} = \frac{-0.8\,s + 25.89}{s + 32.36}$$

The worst-case gain lower bounds corresponding to these perturbations are 1.001 and 0.903, respectively. A combination of worst-case disturbance (scaled to have size 0.5) and uncertainty (of size 0.8) pushes the states of the closed-loop system (dashed line) as far as the lower bound of the worst-case gain. Overall, these simple comparison results show a typical robustness and performance trade-off. The nominal controller performs best at no uncertainty, whereas the robust controller performs better at modeled uncertainty level.

5.6 Summary

This chapter proposed an iterative algorithm to design an output-feedback controller that bounds the worst-case gain of an uncertain LTV system on a finite horizon. A similar design can also be performed in a state-feedback formulation. Time-domain dynamic IQCs were used to describe the input-output behavior of the uncertainty. The effectiveness of the proposed approach was demonstrated using the LTI system and the nonlinear two-link robot arm example.

Chapter 6

Analysis of Missile Engagements

6.1 Introduction

This chapter applies the finite horizon robustness analysis tools to missile engagements. Section 6.3.1 develops a robustness metric that can be used in the framework of induced gain. The nominal analysis tools from Chapter 3 are used in Section 6.3.2 to analyze the missile engagement example from Section 2.5. The Monte-Carlo studies complement the nonlinear simulations with worst-case disturbances. Since LTV is just an approximation, the size of disturbance is further studied in a nonlinear simulation setting (Section 6.3.3). Finally, a first-order trajectory sensitivity analysis is discussed in Section 6.4 to analyze the impact of parametric uncertainties in the interceptor missile.

6.2 Summary of Related Work

There has been a lot of work that focuses on providing performance guarantees of missile engagements. The most relevant work is summarized here, including capture region conditions and LTV analysis using the adjoint method.

Capture Region: The set of initial conditions for the interceptor and threat missile which will result in a successful intercept is known as the capture region. If the engagement starts in a capture region, then intercept is guaranteed. Most earlier work to provide performance guarantees on missile engagement focused on characterizing the capture region in the relative velocity space [102]. This work assumes a non-maneuvering

target and provides analytical conditions to guarantee intercept if engagement starts in the capture circle. It focuses on the True PN (TPN) guidance law for which the analysis is analytically tractable. This work has been extended to the maneuvering targets in [103]. Moreover, other practical guidance laws such as Pure PN [104], generalized TPN [105], realistic TPN [106] are investigated because TPN is difficult to mechanize in practice [107]. A related work in [108] considers numerically classifying the capture region. These works assume perfect dynamics and ignore the effect of model uncertainties/unmodeled dynamics and external disturbances.

Adjoint Method: The performance of a missile engagement can also be characterized in terms of the miss-distance. One of the most popular methods to compute the miss-distance is through adjoint simulations [109, 110, 111, 35, 112, 113]. The work in [114] proposes SLAM (Statistical Linearization Adjoint Method) to provide an error budget for the RMS miss distance for the missile engagement using a single simulation. The computational conditions in [114] are similar to that for the finite horizon H_2 norm (Section 3.4). In contrast, this chapter considers the worst-case miss-distance (perpendicular to the line-of-sight) as a performance metric. A similar metric is considered in [115, 116] and also used in [117] to develop an evasive guidance law.

Robustness to Model Uncertainties: The most closely related work in [29] considers performance analysis of the missile guidance loop using DLMI conditions. A linear double integrator model describes the relative motion between the threat and the interceptor. The worst-case robust H_2 analysis is performed for norm-bounded LTV uncertainties. In that regard, the theoretical tools presented in Chapter 4 handle a more general class of uncertainties described by Integral Quadratic Constraints. However, the computational algorithm in Section 4.4.3 is not efficient for higher-order engagement models such as the one considered in Section 2.5. Thus, future work is needed to make the computational tools more efficient.

6.3 Nominal Analysis

6.3.1 LTV Performance Metric

This section develops a performance metric for the LTV analysis of missile engagements. To make the discussion more concrete, consider the head-on engagement scenario between a single interceptor and a threat discussed in Section 2.5. In this section, we consider the effect of disturbances such as wind gust on the interceptor, which is modeled as a fin-deflection disturbance (d) to the plant input (see Simulink model in Figure 2.6). The nominal simulation leads to an intercept at time $T = 5.208$ seconds. The nominal end-game is shown in the Figure 6.1, where a shaded red region around the threat missile represented 20 feet of the lethal radius when the simulation stopped. Numerical linearization from input fin-deflection disturbance to the relative position yields the following LTV system on the horizon $[0, 5.208]$ seconds:

$$\dot{x}(t) = A(t)\,x(t) + B(t)\,d(t), \qquad x(0) = 0 \tag{6.1}$$

$$e(t) = C(t)\,x(t) \tag{6.2}$$

where $x(t) \in \mathbb{R}^{34}$ is the linearized state along the trajectory, $d(t) \in \mathbb{R}$ is the disturbance input and $e(t) \equiv \delta R(t) \in \mathbb{R}^2$ is a first-order perturbation to the Line-of-Sight (LOS) vector R at time t. Figure 6.2 shows a LOS vector R (green), a perpendicular vector

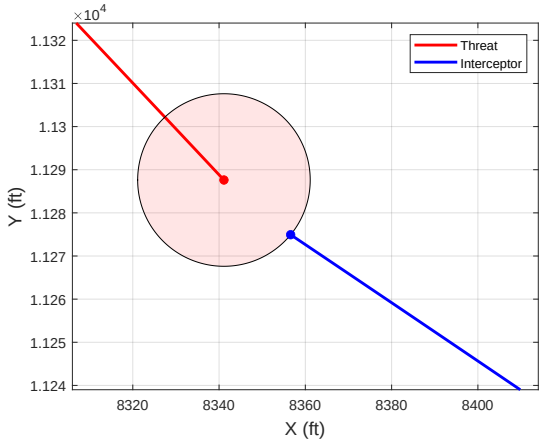

Figure 6.1: Nominal Simulation (Zoomed)

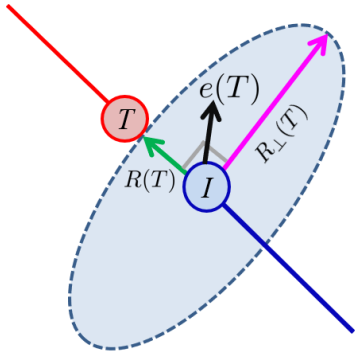

Figure 6.2: Endgame Geometry

$R_\perp$ (magenta) and the first-order perturbation vector e (black) at the final time T. A known limitation of the PN guidance law is that it cannot eliminate the induced miss-distance perpendicular to the LOS at the final time [118]. With limited maneuverability, there is almost no time available for the outer loop guidance law to correct this induced error, which will lead the interceptor to miss its target. Note that any induced error in the direction of LOS will typically change the time of intercept, however does not play a major role in inducing miss-distance [118]. To perform a meaningful LTV analysis, we restrict the first-order perturbation of the LTV model (6.2) to the perpendicular subspace of LOS. First, define a time-varying vector $n(t)$ as follows:

$$n(t) := \frac{R(t)}{\|R(t)\|_2} \tag{6.3}$$

Note that $\|n(t)\|_2 = 1$ at any time $t \in [0, T]$. Recall that $e(t)$ is the first-order perturbation to the LOS vector at time t. The vector $e_{\text{LOS}}(t) := \left[n(t)^\top e(t)\right] n(t)$ is the projection of $e(t)$ onto the LOS vector R at time t. The remainder $\tilde{e}(t) = e(t) - e_{LOS}(t)$ corresponds to the first-order perturbation (perpendicular to the LOS) in the relative position space at time t. This can be expressed as follows:

$$\tilde{e}(t) := \left[I - n(t)n(t)^\top\right] e(t) \tag{6.4}$$

The modified output $\tilde{e}(T)$ provides the perturbation perpendicular to the LOS at the nominal intercept time T. The induced $\mathcal{L}_2$-to-Euclidean norm from d to the modified output $\tilde{e}(T)$ serves as a metric of robustness to characterize the intercept. The specific worst-case disturbance can be scaled to induce a given miss-distance in LTV simulation*. Due to the symmetry of linear analysis, the disturbance with a negative sign also induces the same miss-distance in the LTV simulation.

Remark 6.3.1. *As noted above, the modified output $\tilde{e}$ captures the perpendicular deviation in the relative position space. A similar metric is also used in [117] to design an evasive guidance law. However, an alternative metric such as relative velocity [116] or acceleration (perpendicular to the LOS) can be considered with minor modification to the Simulink model.*

*The proposed metric for LTV analysis is also applicable to the 3D engagement setting. In this case, the LOS vector is in $\mathbb{R}^3$, and the perpendicular subspace is a plane in 2D.

6.3.2 Worst-Case Interceptor Disturbance

This section further studies the impact of the worst-case fin-deflection disturbances on the interceptor. The output of the LTV system G obtained in Figure 2.7 is modified as described in the previous section using Equation (6.4). The computed induced $\mathcal{L}_2$-to-Euclidean gain from d to the output $\tilde{e}$ is 7.082 in the units of feet/(degrees$\sqrt{\text{seconds}}$). The worst-case disturbance (in degrees) of unit $\mathcal{L}_2[0,T]$ norm will induce 7.082 feet of miss-distance for the LTV model. However, this disturbance is not large enough to cause a miss for the stopping criteria of 20 feet used in the model. Thus, the worst-case disturbance is scaled to have $\|d_{wc}\|_{2,[0,5.208]} = 20/7.082 = 2.8241$, so as to induce a miss-distance of 20 feet in the LTV simulation. This scaled disturbance is shown in Figure 6.3.

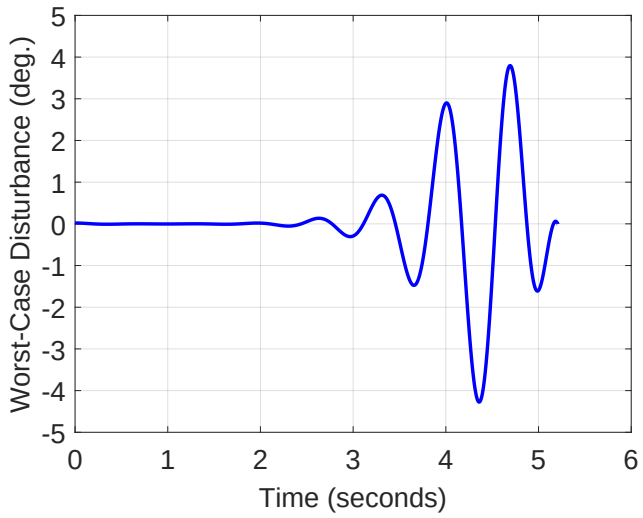

Figure 6.3: Worst-Case Disturbance with $\|d_{wc}\|_{2,[0,5.208]} = 2.8241$

Next, the effect of the fin-deflection disturbance is evaluated in the nonlinear simulation model. We consider a set of 100 candidate disturbances on the horizon of $[0, 5.208]$ seconds for Monte-Carlo simulations. These are pulse-shaped disturbances where the pulse start-time and stop-time are randomly chosen (from a uniform distribution) 2.5 to 4.5 seconds and 4.6 to 5.208 seconds, respectively. The pulse is scaled to have $\mathcal{L}_2[0,T]$ norm of 2.8241 degrees$\sqrt{\text{seconds}}$. The worst-case disturbance has a peak value of 4.276

degrees, whereas the largest peak value of sampled disturbance is 6.02 degrees. Other known disturbance profiles of the same $\mathcal{L}_2[0,T]$ norm can also be included. The nonlinear model is simulated using these disturbances to study their impact on the endgame. Such Monte-Carlo simulations are shown as cyan interceptor trajectories in Figure 6.4 overlaid on the nominal trajectories. The threat missile (red) is headed towards a fixed asset and is assumed to be guided by the PN guidance law for all simulations. The cyan ball around the interceptor at the nominal intercept time shows 20 feet of miss-distance that the worst-case disturbance can induce in the LTV simulation. The nonlinear engagement can also be simulated using $+d_{wc}$ and -d_{wc}, which yields the (green and magenta) worst-case trajectories for the interceptor missing its target (i.e., miss-distance in nonlinear simulation is $>$ 20 feet at the time of intercept).

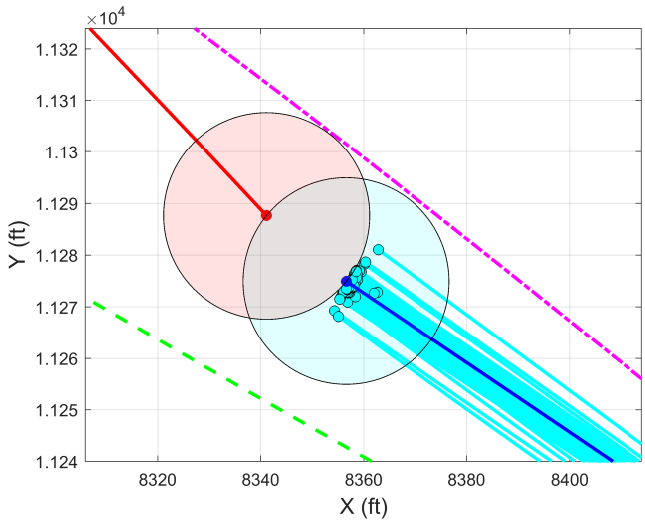

Figure 6.4: 100 Nonlinear Monte-Carlo simulations (cyan) overlaid on a nominal interceptor simulation (blue) along with worst-case interceptor trajectories with disturbances $+d_{wc}$ (green dashed) and -d_{wc} (magenta dot-dashed).

This example demonstrates the effectiveness of LTV analysis applied to study interceptor disturbances. We observe that the interceptor successfully captures the threat for the same size of sampled disturbances used in the Monte-Carlo simulations. However, interceptor trajectories with the worst-case disturbance miss the target. Moreover, it

took approximately 0.85 seconds to compute induced gain and the worst-case disturbance (in Figure 6.3) using the power iteration algorithm (Algorithm 1). In comparison, a single nonlinear simulation with sampled disturbance took an average of 2.25 seconds[†] of computational time. This corresponds to approximately 225 seconds of computational time to run 100 Monte-Carlo simulations. As noted earlier in Chapter 3, the power iteration algorithm on a finite-dimensional space is guaranteed to converge and is the faster method. However, the Gramian based method in Section 3.8.2 can also be used at some additional computational expense to compute a tube around nominal interceptor trajectory. Overall, LTV analysis quickly finds such edge-case behavior early in the design iterations. Thus, the worst-case disturbances computed using LTV analysis should be included in high-fidelity Monte-Carlo simulations. This disturbance can be used to initialize the nonlinear power iteration algorithm in [8].

This example modeled wind disturbance as an effective fin-deflection to the interceptor and included only pulse-shaped disturbances in the Monte-Carlo simulations. However, a more realistic approach considers characterizing allowable disturbances using altitude-dependent models (e.g., Dryden wind-gust model) and performing Monte-Carlo simulations against a more realistic set of disturbances. Moreover, the LTV analysis presented in this section guarantees desired miss-distance only in the linearized engagement. The nonlinear simulation results in Figure 6.4 show an asymmetrical behavior of the worst-case trajectories around the threat, which is further discussed in the next section.

6.3.3 Gap in LTV Analysis and Nonlinear Performance

This section studies the gap in linear and nonlinear analysis for the example considered in the previous section. Specifically, the miss-distance obtained by scaling the worst-case disturbance is studied for linear (time-varying) and nonlinear models.

Consider Figure 6.5 where the horizontal axis is a size of worst-case disturbance d_{wc} and the vertical axis is the terminal Euclidean norm of the projected output $\tilde{e}$. Based on the previous section, we expect the linear analysis cost (solid blue line) to grow linearly

[†]Simulations are performed using MATLAB/Simulink running on a laptop computer with Intel Core i7 processor, 2.8 GHz CPU, and 16 GB of RAM. Parallel computing resources can significantly speed up these simulations; however, they are not used in this example.

at the slope of 7.082. The nonlinear cost can be computed by simulating the nonlinear model on a horizon of $[0, 5.208]$ seconds using the scaled worst-case disturbance, where the scaling is gradually increased from 0.1 to 3. For the smaller size of disturbance, the linear cost matches well with the related cost in a nonlinear sim with $+d_{wc}$ (green dashed) and $-d_{wc}$ (magenta dot-dashed). However, there are deviations between linear cost and nonlinear simulation for larger disturbances. Due to model nonlinearities, this gap is not unexpected. It may be different for other high-fidelity models or another engagement setup.

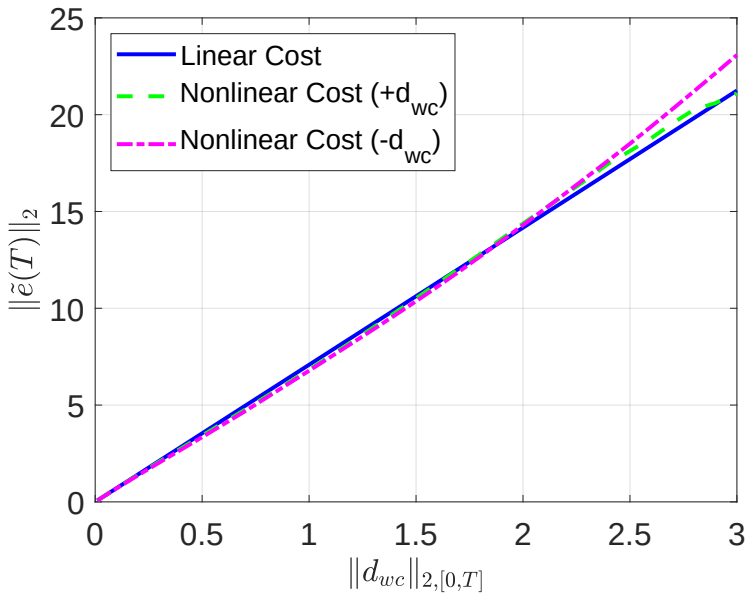

Figure 6.5: Comparison of a Linear and Nonlinear Cost

The example considered here shows that the nonlinear miss-distance (when simulated with $+d_{wc}$ or $-d_{wc}$) is higher than the 20 feet obtained through LTV analysis for the size of $\|d_{wc}\|_{2,[0,5.208]} = 2.8241$. However, this conclusion may not hold in general and nonlinear cost might be lower than the one estimated by the linear analysis. A bisection algorithm can be used to find the exact size of the worst-case disturbance that causes an interceptor to miss its target in a nonlinear high-fidelity simulation. The next remark provides a comment on the H_2 analysis of the LTV system and the gap to nonlinear Monte-Carlo analysis.

Remark 6.3.2. *The close agreement between the linear Monte-Carlo simulations and the 3σ bound computed in Section 3.9.2 is due to the linearity of the original system. The agreement between nonlinear Monte-Carlo simulations and 3σ bound obtained through LTV analysis may not generally hold due to system nonlinearities.*

6.4 Trajectory Sensitivity Analysis

The previous section used induced gain to analyze the impact of exogenous disturbances. This section uses trajectory sensitivities to analyze the impact of uncertain parameters in the interceptor model. Consider an engagement setup in which threat dynamics are unknown, but the threat is assumed to be following some trajectory to hit a static asset somewhere. A PN guided interceptor missile is launched from the origin to engage the threat. The interceptor model is similar to the one considered in Section 2.5 with 3-loop autopilot, fin actuator, and nonlinear dynamics. The nominal aerodynamic coefficients of the model are given from Equation (2.9) to (2.13). The nominal simulation is shown in Figure 6.6, where the interceptor hits the threat in 4.866 seconds. Figure 6.6 shows the (zoomed) nominal range (solid black) as a function of time. The simulation stopped when the range (distance) between the threat and an interceptor reached 20 feet in magnitude.

The first order trajectory sensitivity analysis presented in Section 4.6 is used to demonstrate the impact due to uncertainties in aerodynamic parameters. There are a total of 9 constant coefficients in Equation (2.9) to (2.13), which are only assumed to be known to within $\pm 10\%$ of their nominal values. The nonlinear missile model is linearized on the horizon $[0, 4.866]$ seconds from 9 aerodynamic parameter input to the range magnitude. This yields an LTV system similar to Equation (4.21).

The two linear programs presented in the Section 4.6 are used to efficiently compute upper and lower bound on the range at 20 linearly spaced grid points on horizon $[0.01, 4.866]$ seconds. It took approximately 1.24 seconds to compute these bounds using MATLAB's `linprog` function. These bounds are shown in Figure 6.7 as red and blue lines, respectively. A total of 500 different parameter vectors are uniformly sampled from the uncertainty set ($\pm 10\%$ of nominal values) to perform nonlinear Monte-Carlo

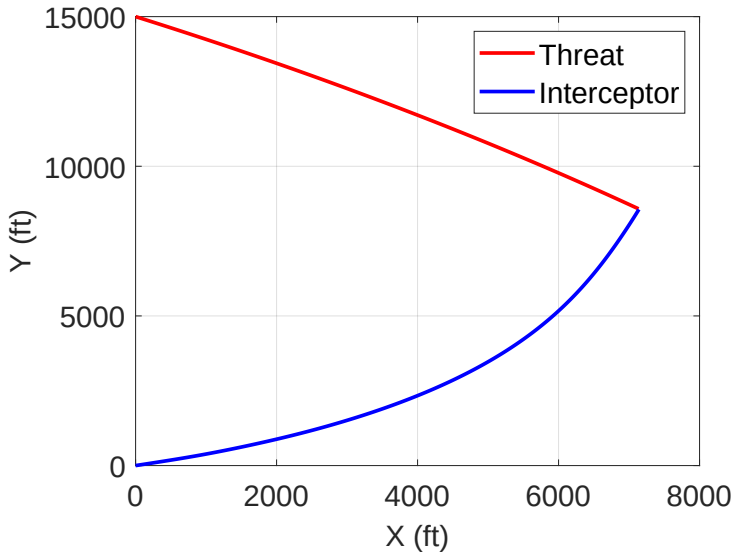

Figure 6.6: Nominal Simulation

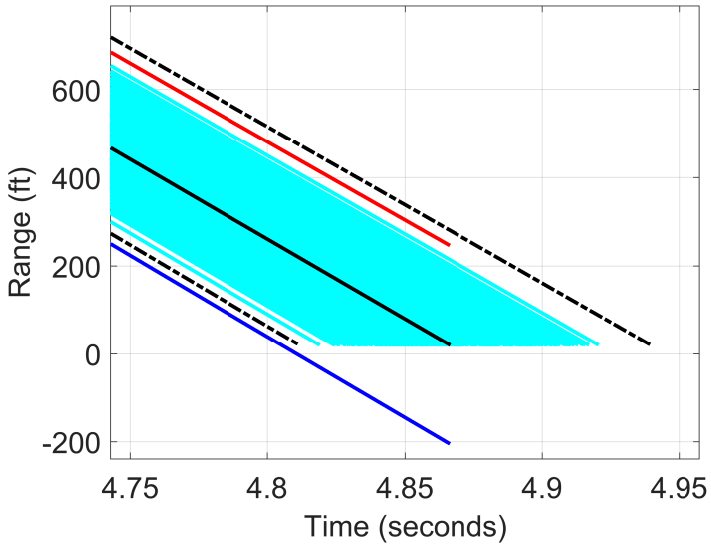

Figure 6.7: Nominal Range (black solid), Upper (red) and Lower (blue) bounds on the Range from first order sensitivity analysis, 500 Monte-Carlo Simulations (cyan), Nonlinear simulations with worst-case parameter combinations (black dot-dashed).

simulations. The range obtained in all of these simulations is shown as cyan trajectories in Figure 6.7. Moreover, the worst-case combinations obtained from sensitivity analysis are further studied in the nonlinear simulation to get the black dot-dashed line in Figure 6.7, which agrees well with the computed bounds. In general, such sensitivity analysis suggests that intercept is guaranteed (i.e., range will drop to 20 feet) even if we have 10% uncertainty in aerodynamic parameters for the interceptor missile.

6.5 Summary

This chapter presented applications to missile engagements using the finite horizon LTV robustness analysis tools developed in previous chapters. We looked at disturbances and parametric uncertainties individually to assess their behavior on the engagement endgame. The combined worst-case behavior should further be studied in the nonlinear simulation setting. The approach presented in this chapter for single interceptors and threats can be extended to higher fidelity 6-DOF missile models and different engagement setups, including multiple interceptors and threats.

A similar finite horizon analysis can also compute evasive trajectories for the threat missile to evade from an incoming interceptor. However, such an approach requires a full engagement model, typically unavailable. An approximate model can be used to design evasive trajectories; however, the design must guarantee the safety of a threat (from the interceptor) in the presence of model uncertainties and nonlinearities. The theoretical tools developed in the next chapter aim to solve an onboard optimization problem for the threat to escape from being intercepted successfully.

Chapter 7

Robust Safety-Critical Control

7.1 Introduction

This chapter presents safety-critical control for a plant with sector-bounded input uncertainties. There are many applications, including autonomous driving, medical or industrial robotics, and aerospace vehicles, that require prioritizing safety over performance objectives [119]. One of the popular methods to encode safety is using Control Barrier Functions (CBF), which can be used as a constraint in a quadratic program to modify control actions to adhere to safety specifications [120, 121]. CBFs are used for collision avoidance of a fixed-wing aircraft in [122] and for UAVs in [123, 124]. Overall, CBFs can be designed using a representative or a surrogate model of the system [121] or can be learned online [125, 126]. For simplicity, actuator nonlinearities are often ignored, which raises robustness concerns.

The approach used in this chapter is to characterize the input-output behavior of nonlinearities at the plant input using point-wise in time quadratic constraints. We then use the robust control barrier functions (RCBFs) presented in Section 7.4.2 to provide safety guarantees for the entire uncertainty set. This is done by ensuring that a safe action exists for all nonlinearities at the modeled uncertainty level. This combines a traditional robust control approach with the CBF methods for safety-critical control. As a result, we obtain more cautious trajectories close to the unsafe region. The proposed method can also be integrated with the robust Model Predictive Control (MPC) formulation to incorporate future behaviors proactively.

7.2 Summary of Related Work

There is a large body of literature on CBFs with a good overview provided in [121]. The closely related works are input-to-state CBF [127] and robust CBFs [128, 129, 130] for ensuring safety in the presence of $\mathcal{L}_\infty$ bounded disturbances, where the bound on the disturbance is fixed (constant) and does not depend on the choice of control input. The work in [131] considers stochastic disturbances instead. Ref. [132] also considers robust CBFs to account for the changes in the dynamics as a perturbation to the vector field. A key distinction is that the input nonlinearities in our work lead to uncertainties that depend on the control decisions, which are not captured by a disturbance model with a (constant) $\mathcal{L}_\infty$ bound. Our approach provides safety guarantees for static nonlinearities and/or time-varying memoryless input uncertainties. The recent work in [133] considers a more general class of unmodeled dynamics (e.g., time-delays, actuation lag, etc.) using α-IQCs and CBFs. The price for this generality is that the trajectories tend to be even more conservative than those obtained with the method presented here. Other related work on robust CBFs are [134, 135, 136].

7.3 Preliminaries

7.3.1 Problem Formulation

Consider the design interconnection as shown in Figure 7.1. The uncertain plant P is described as a series interconnection of known part G and an unknown perturbation ϕ at the plant input.

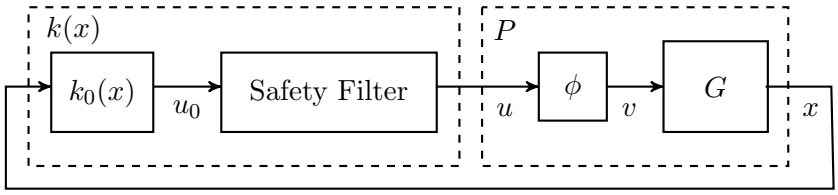

Figure 7.1: Uncertain State-Feedback Design Interconnection

This perturbation represents nonlinearities and/or time-varying, memoryless uncertainties. We will refer to ϕ as a "nonlinearity" for simplicity. Let P be given with the

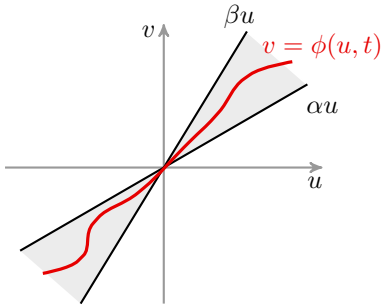

Figure 7.2: Illustration of SISO Nonlinearity $\phi \in [\alpha, \beta]$

following input-affine dynamics:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + g(x(t))\, v(t), \qquad x(0) = x_0 \\ v(t) &= \phi(u(t), t) \end{aligned} \tag{7.1}$$

where $x(t) \in D \subset \mathbb{R}^n$ is the state, $u(t) \in \mathcal{U} \subset \mathbb{R}^m$ is the admissible control input and $v(t) \in \mathcal{V}(u(t)) \subset \mathbb{R}^m$ is the uncertain input. Moreover, $f : D \subset \mathbb{R}^n \to \mathbb{R}^n$ and $g : D \subset \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are locally Lipschitz continuous functions of the state x. It is assumed that the dynamics given by (7.1) are defined on open set $D \subset \mathbb{R}^n$ and are forward complete, i.e. for every initial condition $x(0) \in D$, there exists a unique solution $x(t)$ for all $t \geq 0$. The nonlinearity ϕ is assumed to lie in a sector $[\alpha, \beta]$ with $0 < \alpha \leq 1 \leq \beta$ so that the sector-bound* contains the nominal case $v = u$. This sector-bound can be written as the following point-wise in time quadratic constraint (Section 6.1 of [82]):

$$[v(t) - \alpha u(t)]^\top [\beta u(t) - v(t)] \geq 0, \ \forall t \geq 0. \tag{7.2}$$

If $m = 1$ then the single control channel sector-bound can be illustrated as shown in Figure 7.2. The uncertainty ϕ lies in a sector $[\alpha, \beta]$ when two lines can bound it with slopes of α and β, respectively. The shaded gray region represents the allowable uncertainty set. In more general setting when $m \neq 1$, the constraint (7.2) allows cross-coupling between the input channels.

We assume that a locally Lipschitz continuous function $k_0 : D \subset \mathbb{R}^n \to \mathcal{U} \subset \mathbb{R}^m$

*The sector-bound itself can be functions of time, i.e. $\alpha(t)$ and $\beta(t)$.

is given such that the baseline control law is $u_0 = k_0(x)$, which satisfies stability and acceptable performance but is not necessarily safe. The notion of safety is formalized by defining a safe set $\mathcal{C} \subset D \subset \mathbb{R}^n$ in the state space that the system must remain within [120]. In particular, consider the set $\mathcal{C}$ as the zero-superlevel set of a continuously differentiable function $h : D \subset \mathbb{R}^n \to \mathbb{R}$:

$$\mathcal{C} \triangleq \{x \in D \subset \mathbb{R}^n : h(x) \geq 0\} \tag{7.3}$$

The boundary and interior of the safe set are denoted as $\partial\mathcal{C}$ and $\text{Int}(\mathcal{C})$, respectively. It is assumed that zero is a regular value of h and $\mathcal{C}$ is non-empty with no isolated points. These assumptions mean that $h(x) = 0$ implies $\frac{\partial h}{\partial x}(x) \neq 0$, $\text{Int}(\mathcal{C}) \neq \emptyset$, and $\overline{\text{Int}(\mathcal{C})} = \mathcal{C}$. Explicit time dependence of variables are omitted when it is clear from the context. It is assumed that the initial condition is in the safe set i.e. $x_0 \in \mathcal{C}$.

Our primary goal is to design a safety filter in Figure 7.1 that minimally alters the baseline control command u_0 to prioritize safety. This ensures that the state of the closed-loop system remains safe even in the presence of nonlinearity. The proposed solution in Section 7.4.3 is an optimization problem that can be solved online to compute safe control action $u \in \mathcal{U}$.

7.3.2 Background

This section provides a brief summary on set-invariance using control barrier functions [120]. First, consider the nominal case i.e. without nonlinearities at the plant input. In this case, $v = u$. If we let $u = k(x)$ then the closed-loop dynamics are given by:

$$\dot{x} = f_{cl}(x) = f(x) + g(x)\,k(x), \qquad x(0) = x_0 \tag{7.4}$$

In the context of the autonomous system above, safety is synonymous with the forward invariance of $\mathcal{C}$ [120]:

Definition 7.3.1 (Forward Invariance and Safety)**.** *A set $\mathcal{C} \subset D \subset \mathbb{R}^n$ is forward invariant if for every initial condition $x_0 \in \mathcal{C}$, the solution to the closed-loop system* (7.4) *satisfies $x(t) \in \mathcal{C}$ for all $t \geq 0$. The system* (7.4) *is safe with respect to $\mathcal{C}$ if $\mathcal{C}$ is forward*

invariant.

As mentioned in the previous section, the baseline control law $u_0 = k_0(x)$ is not necessarily safe. In this case, we would like to ensure that a safe action u exists, which can steer the system to remain within the safe set $\mathcal{C}$. This is formalized by defining the notion of control invariance.

Definition 7.3.2 (Control Invariance)**.** *A set $\mathcal{C}$ is control invariant if there exists a controller $k : D \subset \mathbb{R}^n \to \mathcal{U} \subset \mathbb{R}^m$ such that $\mathcal{C}$ is forward invariant with respect to the system* (7.4).

Control barrier functions are used to ensure control invariance for the nominal plant. Let $L_f h := \frac{\partial h}{\partial x} f$ and $L_g h := \frac{\partial h}{\partial x} g$ denote the Lie derivatives of h with respect to f and g. Moreover, a continuous function $\eta : \mathbb{R} \to \mathbb{R}$ is called extended class-$\mathcal{K}_\infty$ (denoted as $\mathcal{K}_{\infty,e}$) if it is strictly monotonically increasing and satisfies $\eta(0) = 0, \lim_{r\to-\infty} \eta(r) = -\infty$, and $\lim_{r\to\infty} \eta(r) = \infty$. Moreover, assume uniform relative degree one i.e. $L_g h(x) \neq 0$, $\forall x \in D$. This implies that the control input u shows up after differentiating the function h once.

Definition 7.3.3 (Control Barrier Function)**.** *Let $\mathcal{C} \subset D \subset \mathbb{R}^n$ be the zero-superlevel set of a continuously differentiable function $h : D \subset \mathbb{R}^n \to \mathbb{R}$. Then h is a Control Barrier Function if there exists $\eta \in \mathcal{K}_{\infty,e}$ such that for all $x \in D$:*

$$\sup_{u\in\mathcal{U}} [L_f h(x) + L_g h(x)\, u] \geq -\eta(h(x)) \tag{7.5}$$

The existence of a control barrier function h satisfying constraint (7.5) implies that if the state reaches the boundary of $\mathcal{C}$, then the control input u can be used to prevent the state from entering the unsafe region. The main result in [120, 121] can be used to design a Lipschitz continuous safety filter that yields safety for the nominal closed-loop. The specific implementation involves solving the following quadratic program online with CBF constraint to minimally alter the baseline action u_0.

$$\begin{aligned} u^*(x) = \arg\min_{u\in\mathcal{U}} \frac{1}{2}\|u - u_0\|_2^2 \qquad & \text{(CBF-QP)} \\ \text{s.t. } L_f h(x) + L_g h(x)\, u \geq -\eta(h(x)) & \end{aligned}$$

In general, constructing a CBF h is not straightforward and often requires careful consideration [121].

Next, consider the uncertain case with the sector-bounded nonlinearity ϕ at the plant input. This nonlinearity can alter the control command resulting in a safety violation. The uncertain closed-loop dynamics with the nonlinearity can be written as follows:

$$\dot{x} = f_{ucl}(x) = f(x) + g(x)\,\phi(k(x), t), \quad x(0) = x_0 \tag{7.6}$$

The notion of robust control invariance (Definition 4.4 in [137]) is helpful to guarantee safety in the presence of uncertainties. A specific definition for sector-bounded nonlinearity is given next.

Definition 7.3.4 (Robust Control Invariance and Robust Safety)**.** *A set $\mathcal{C}$ is robust control invariant if there exists a controller $k : D \subset \mathbb{R}^n \to \mathcal{U} \subset \mathbb{R}^m$, such that $\mathcal{C}$ is forward invariant with respect to the uncertain closed-loop* (7.6) *for all nonlinearities ϕ in a sector $[\alpha, \beta]$. The system* (7.6) *is robustly safe with respect to $\mathcal{C}$ if $\mathcal{C}$ is robust control invariant.*

Robust control invariance can be ensured by showing the existence of a robust control barrier function, as presented in the following section.

7.4 Main Results

7.4.1 Uncertainty Mapping

The first step is to use a loop-shifting transformation (Section 6.5 of [82]) to map the nonlinearity $\phi \in [\alpha, \beta]$ into a normalized, input additive form, as in standard robust control workflow. To make this precise, define $\Delta : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$ such that $v(t) = \phi(u(t), t)$ is mapped to:

$$v(t) = \frac{1}{2}\,(\alpha + \beta)(u(t) + \Delta(u(t), t)) \tag{7.7}$$

The mapped nonlinearity satisfies $\Delta \in [-\theta, +\theta]$ where $\theta := (\beta - \alpha)/(\beta + \alpha)$. This re-centers the sector-bound to 0 and separates the nominal control action $u(t)$ and the uncertain control command $\Delta(u(t), t)$. The factor $\frac{1}{2}\,(\alpha + \beta)$ scales the input function

$g(x)$ to be $\tilde{g}(x) := \frac{1}{2}(\alpha + \beta)\, g(x)$. This yields the following input-affine system with mapped nonlinearity:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + \tilde{g}(x(t))\,(u(t) + w(t)), \quad x(0) = x_0 \\ w(t) &= \Delta(u(t), t) \end{aligned} \tag{7.8}$$

Assume the uncertainty level satisfies $0 \leq \theta < 1$. The symmetric sector constraint on the mapped nonlinearity Δ corresponds to a norm bound:

$$\|w(t)\|_2 \leq \theta \|u(t)\|_2, \ \forall t \geq 0. \tag{7.9}$$

Note that as $\theta \to 0$, i.e. as α and β both tend to 1, then $w(t) \to 0$ and we recover the nominal plant G. For simplicity, the remainder of this chapter considers the system (7.8) with mapped nonlinearity Δ instead of the original system (7.1) with ϕ. Let $\mathcal{W}(u(t))$ denote the set of uncertain inputs $w(t)$ satisfying the norm bound constraint (7.9). The set $\mathcal{W}(u(t))$ depends on the control input u at time t. Thus, the uncertain input $w(t)$ can not be simply treated as an exogenous disturbance input, as it depends on $u(t)$ through the nonlinearity Δ.

7.4.2 Robust Control Barrier Functions (RCBF)

Robust control barrier functions defined in this section can be used to synthesize controllers ensuring the safety of the uncertain closed-loop system with respect to a given set $\mathcal{C}$.

Definition 7.4.1. *Let $\mathcal{C} \subset D \subset \mathbb{R}^n$ be a safe set given by* (7.3)*. Assume $L_{\tilde{g}} h(x) \neq 0, \forall x \in D$. The function h is a Robust Control Barrier Function for* (7.8) *if there exists $\eta \in \mathcal{K}_{\infty,e}$ such that for all $x \in D$:*

$$\sup_{u \in \mathcal{U}} \inf_{w \in \mathcal{W}} \left[L_f h(x) + L_{\tilde{g}} h(x)(u + w) \right] \geq -\eta(h(x)) \tag{7.10}$$

The nonlinearity Δ can, in the worst case, yield an uncertain input w that minimizes the left side of inequality (7.10). We aim to choose a single control input $u \in \mathcal{U}$ for all uncertain inputs $w \in \mathcal{W}$ satisfying the constraint (7.9). The worst-case uncertain input

$w^*(u)$ after solving the inner optimization problem is given by:

$$w^*(u) = -\theta \|u\|_2 \frac{L_{\tilde{g}} h(x)^\top}{\|L_{\tilde{g}} h(x)\|_2} \tag{7.11}$$

This follows from the linear cost in (7.10), but a more formal argument using Lagrange relaxation is given in Appendix C.1. It can be verified that as $\theta \to 0$, we have $w^*(u) \to 0$. Plugging in for $w^*(u)$ in condition (7.10) yields:

$$\sup_{u \in \mathcal{U}} \left[L_f h(x) + L_{\tilde{g}} h(x)(u + w^*(u)) \right] \geq -\eta(h(x)) \tag{7.12}$$

Define $p(x) := L_f h(x) + \eta(h(x))$ and the set of all control actions that render the set $\mathcal{C}$ robustly safe as follows:

$$\mathcal{U}_{RCBF}(x) := \{u \in \mathcal{U} : p(x) + L_{\tilde{g}} h(x)(u + w^*(u)) \geq 0\}$$

The feasibility of RCBF constraint (7.10) ensures that the above set is nonempty. This implies that if the state reaches the boundary of $\mathcal{C}$, then there exists a control input to prevent the state of the uncertain closed-loop from crossing out of the safe set. Thus, the existence of a robust control barrier function implies that the system is robustly safe. This statement is formalized in the following theorem, which can be viewed as a robust version of the main result in [120].

Theorem 7.4.1. *Let $\mathcal{C} \subset D$ be a safe set defined using (7.3) as the superlevel set of a continuously differentiable function $h : D \subset \mathbb{R}^n \to \mathbb{R}$. If h is a robust control barrier function on D and $\frac{\partial h}{\partial x}(x) \neq 0$ for all $x \in \partial\mathcal{C}$, then any Lipschitz continuous controller $u(x) \in \mathcal{U}_{RCBF}(x)$ for the system (7.8) renders the set $\mathcal{C}$ robust control invariant.*

Proof. If h is a RCBF on open set D then for any $x \in \partial\mathcal{C}$ and for all $w \in \mathcal{W}$, at each time t there exists a control input $u(t) \in \mathcal{U}$ such that $\frac{d}{dt} h(x(t), w(t), u(t)) \geq -\eta(h(x)) = 0$. For a Lipschitz continuous control law $u(x) \in \mathcal{U}_{RCBF}(x)$, according to generalizations of Nagumo's theorem (Theorem 4.10 in [137]) the closed set $\mathcal{C}$ is robust control invariant. □

7.4.3 Optimization-Based Control

We can solve the following optimization online to compute a control input u that minimally alters the baseline input u_0 to ensure safety:

$$u^*(x) = \arg\min_{u \in \mathcal{U}} \frac{1}{2}\|u - u_0\|_2^2 \quad \text{s.t. } p(x) + L_{\tilde{g}}h(x)(u + w^*(u)) \geq 0 \tag{7.13}$$

Plugging in for $w^*(u)$ and expanding the cost function yields the following problem with an equivalent optimizer.

$$u^*(x) = \arg\min_{u \in \mathcal{U}} \left[\frac{1}{2}u^\top u - u_0^\top u\right] \quad \text{s.t. } p(x) + L_{\tilde{g}}h(x)u - \theta\|u\|_2\|L_{\tilde{g}}h(x)\|_2 \geq 0 \tag{7.14}$$

Again, the last term drops out as the uncertainty level $\theta \to 0$, yielding an affine constraint in u. In this case, the above optimization problem is simply a (nominal) CBF-QP. However, for $\theta > 0$, the decision variable u in the constraint of (7.14) appears as the Euclidean norm $\|u\|_2$. Reformulate the problem (7.14) using a slack variable q as follows:

$$\begin{bmatrix} u^*(x) \\ q^*(x) \end{bmatrix} = \arg\min_{u \in \mathcal{U}, q} \left[q - u_0^\top u\right] \quad \begin{aligned} &\text{s.t. } p(x) + L_{\tilde{g}}h(x)u \geq \theta\|L_{\tilde{g}}h(x)\|_2\|u\|_2 \\ &2q \geq \|u\|_2^2 \end{aligned} \tag{7.15}$$

This yields a minimization problem over $u \in \mathcal{U}$ and $q > 0$. The optimal solutions are related by $2q^* = u^{*\top}u^*$. The second constraint can be rewritten as a rotated second-order cone (SOC) condition as in Section 10.1 of [138], which yields the optimization (7.15)

as follows:

$$\begin{bmatrix} u^*(x) \\ q^*(x) \end{bmatrix} = \arg\min_{u\in\mathcal{U},q} \begin{bmatrix} q - u_0^\top u \end{bmatrix} \qquad \text{(RCBF-SOCP)}$$
$$\text{s.t. } \theta\|L_{\tilde{g}}h(x)\|_2\|u\|_2 \leq p(x) + L_{\tilde{g}}h(x)u$$
$$\left\| \begin{bmatrix} \sqrt{2}\,u \\ q-1 \end{bmatrix} \right\|_2 \leq q+1$$

This problem falls under a special class of convex optimization problems known as second-order cone programs (SOCP) [139], which can be solved online using existing numerical solvers. At higher uncertainty levels, the RCBF-SOCP problem (if feasible) yields more cautious (conservative and safe) control actions to prevent the states from entering the unsafe region. For the point-wise feasibility of the RCBF-SOCP, it is assumed that the set of control inputs $\mathcal{U}$ is not overly restrictive, thus allowing us to have sufficient control authority to maintain safety in the presence of modeled uncertainty. However, a similar approach to [140] can also be used to relax this assumption. In scalar input case ($m = 1$), an efficient implementation can be obtained by appropriately scaling the solution u^* of the CBF-QP problem (Appendix C.2.2), however this is not true in general with $m > 1$.

7.4.4 Lipschitz Continuity

This section discusses a key Lipschitz continuity property of the RCBF-SOCP problem. In the nominal case ($\theta = 0$), if $\mathcal{U} \equiv \mathbb{R}^m$, then the CBF-QP has only a single linear constraint in u, and in this special case, there is a straightforward solution. This solution, along with assumption $L_{\tilde{g}}h(x) \neq 0$, can be used to show that the resulting safety filter is a locally Lipschitz continuous function of the state $x \in D$ (Theorem 8 of [128]). For the robust case ($\theta \neq 0$), the optimization problem (7.13) can compactly be written as:

$$u^*(x) = \arg\min_{u\in\mathcal{U}_{RCBF}(x)} \frac{1}{2}\|u - u_0\|_2^2 \tag{7.16}$$

To the best of our knowledge, there is no explicit solution to this general problem. However, it is a standard projection problem over the parameterized non-empty closed convex set, i.e. the optimizer $u^*(x)$ is a projection of $u_0 = k_0(x)$ onto the set $\mathcal{U}_{RCBF}(x)$. The main results in Section 6 of [141] show that $u^*(x)$ is a locally Lipschitz continuous

function of x, if the set $\mathcal{U}_{RCBF}(x)$ is described by polyhedral constraints parameterized by the state x. We conjecture that $u^*(x)$ remains locally Lipschitz when $\mathcal{U}_{RCBF}(x)$ is described by the two SOC constraints as in the RCBF-SOCP problem. Future work will focus on investigating this further.

The remainder of this section presents a Lipschitz continuity result for the special case of scalar control input $u \in \mathbb{R}$. For this case $m = 1$ and $L_{\tilde{g}}h(x) \in \mathbb{R}$. The optimization (7.13) can be written as follows:

$$\begin{aligned} u^*(x) = \arg\min_{u} \frac{1}{2}(u - u_0)^2 \\ \text{s.t. } p(x) + L_{\tilde{g}}h(x)\, u - \theta|u||L_{\tilde{g}}h(x)| \geq 0 \end{aligned} \tag{7.17}$$

The following theorem makes a formal statement about the Lipschitz continuity of the function u^*. A direct and more concise (independent of [141]) proof is provided in this case.

Theorem 7.4.2. *Assume $f : D \subset \mathbb{R}^n \to \mathbb{R}^n$, $g : D \subset \mathbb{R}^n \to \mathbb{R}^{n\times 1}$, $\eta \in \mathcal{K}_{\infty,e}$ and u_0 are given locally Lipschitz continuous functions. Let $u \in \mathbb{R}$ and $h : D \subset \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous robust control barrier function. Moreover, let $L_{\tilde{g}}h(x) \neq 0$. Then the solution, $u^*(x)$ of (7.17) is a locally Lipschitz continuous function of $x \in D$.*

Proof. Assume $L_{\tilde{g}}h(x) > 0$ and $0 \leq \theta < 1$. The constraint from (7.17) can be written as:

$$u - \theta|u| \geq \frac{-p(x)}{L_{\tilde{g}}h(x)} \tag{7.18}$$

If $\frac{-p(x)}{L_{\tilde{g}}h(x)} \geq 0$ then $u \geq 0$. Hence the constraint (7.18) is:

$$u \geq \frac{-p(x)}{(1-\theta)L_{\tilde{g}}h(x)} \tag{7.19}$$

If $\frac{-p(x)}{L_{\tilde{g}}h(x)} < 0$ then (7.18) is satisfied for any $u \geq 0$. Moreover it is satisfied for negative values of u that satisfy:

$$u \geq \frac{-p(x)}{(1+\theta)L_{\tilde{g}}h(x)} \tag{7.20}$$

Thus the constraint in (7.17) is a parameterized interval of the form $u(x) \in [u_l(x), +\infty)$ where $u_l(x)$ is given by:

$$u_l(x) = \max\left\{ \frac{-p(x)}{(1-\theta)L_{\tilde{g}}h(x)},\ \frac{-p(x)}{(1+\theta)L_{\tilde{g}}h(x)} \right\} \tag{7.21}$$

The functions p and $L_{\tilde{g}}h$ are locally Lipschitz continuous, since f, g, η and h are locally Lipschitz continuous by assumption. Note that $L_{\tilde{g}}h(x) \neq 0$, because $L_{\tilde{g}}h(x)$ is Lipschitz continuous, so there exists an $\epsilon > 0$ and a ball around x such that $|L_{\tilde{g}}h(x)| \geq \epsilon > 0$. Thus, by Proposition 1.30 and Corollary 1.31 of [142], the ratio $\frac{-p(x)}{L_{\tilde{g}}h(x)}$ is also locally Lipschitz continuous. The boundary function u_l is locally Lipschitz continuous, because the point-wise maximum/minimum of two Lipschitz functions is also Lipschitz by Proposition 1.32 of [142]. Finally, the optimal solution of (7.17) can be written as $u^*(x) = \max\{u_l(x), u_0\}$. The baseline controller u_0 is assumed to be locally Lipschitz continuous. Thus, again using the Proposition 1.32 of [142], we have that u^* is a locally Lipschitz continuous function of $x \in D$. The case $L_{\tilde{g}}h(x) < 0$ can be handled similarly. □

7.4.5 Extensions

Sector-Bound for Individual Control Channels:

Consider the case where the control input $u(t) \in \mathbb{R}^m$ is vector-valued ($m > 1$). The sector bound corresponding to the constraint (7.2) allows for uncertainty that is coupled across input channels. An alternative model is to treat each input as having its own sector-bounded nonlinearity with no cross-coupling. In this case, the dynamics (7.8) can be written as:

$$\dot{x}(t) = f(x(t)) + \sum_{i=1}^{m} \tilde{g}_i(x(t))\,(u_i(t) + w_i(t)), \quad x(0) = x_0$$

where $w_i(t) = \Delta_i(u_i(t), t)$, and each Δ_i satisfies the constraint $|w_i(t)| \leq \theta_i |u_i(t)|$, $\forall t \geq 0$. The worst-case uncertain input for an individual channel can be obtained as $w_i^* = -\theta_i |u_i| \text{sgn}(L_{\tilde{g}_i}h(x))$. The linear programming trick (presented in Appendix C.2.1 for a scalar input) can be used to separate positive and negative parts of the individual

control input u_i. This yields a quadratic program for the safety filter.

Robust Exponential CBF:

The previous section relied on the uniform relative degree one assumption used in Definition 7.4.1. However, the proposed approach can be generalized to cases where h has higher relative degree. This requires the uncertain input w in Equation (7.8) to be matched with the control input u. This matching condition allows us to naturally extend the RCBF framework to higher relative degree robust CBFs (referred to as robust exponential CBFs or RECBF). The theory is similar to that of the (nominal) exponential CBF combined with the robustness argument already presented in this chapter before. More details on (nominal) ECBF are provided in [143] with an overview in [121]. The example in Section 7.5 demonstrates the RECBF design approach for the relative degree of two.

Unifying with RCLF:

Consider the case where the baseline controller's objective is to regulate the system to the desired equilibrium point (in the absence of safety considerations). Suppose that a baseline controller is designed using a robust control Lyapunov function (RCLF) (Chapter 3 of [144]). This ensures that the closed-loop is robustly stable in the absence of safety concerns. The proposed RCBF-SOCP does not necessarily guarantee asymptotic stability to the desired equilibrium point when safety constraints are included. It is helpful to combine the stability and safety objectives in a single multi-objective optimization problem to design the controller $k(x)$ in Fig. 1. For the nominal case ($\theta = 0$), this problem is referred to as CLF-CBF QP [121]. An analogous formulation for the robust counterpart ($\theta > 0$) yields an RCLF-RCBF SOCP problem as follows:

$$
\begin{aligned}
u^*(x) = \arg \min_{u \in \mathcal{U},\, \delta \in \mathbb{R}} \left[\frac{1}{2} u^\top H(x) u + p\delta^2 \right] \qquad & \text{(RCLF-RCBF SOCP)} \\
\text{s.t. } L_f h(x) + L_{\tilde{g}} h(x) u - \theta \|u\|_2 \|L_{\tilde{g}} h(x)\|_2 \geq -\eta(h(x)) & \\
L_f V(x) + L_{\tilde{g}} V(x) u + \theta \|u\|_2 \|L_{\tilde{g}} V(x)\|_2 \leq -\gamma(V(x)) + \delta &
\end{aligned}
$$

where $V : D \subset \mathbb{R}^n \to \mathbb{R}$ is the continuously differentiable RCLF and γ is a class $\mathcal{K}$ function. This specific implementation uses a hard constraint with an RCBF to enforce robust safety and a soft constraint with an RCLF (using a slack variable δ). This formulation strictly enforces the robust safety constraint with minimal deviation (as measured by the slack variable) to the RCLF stability condition. Note that safety has a higher priority in this formulation. As a result, asymptotic stability to the desired equilibrium point is not necessarily assured.

Parametric Uncertainties:

The robust design approach presented in this chapter can also be extended to systems with uncertain parameters in the input function $g(x)$. Let the system dynamics be given by:

$$\dot{x} = f_0(x) + \left[g_0(x) + \sum_{i=1}^{n_p} g_i(x)\delta_i \right] u, \quad x(0) = x_0$$

where the functions f_0 and g_0 capture the nominal dynamics. The remaining terms capture the effects of uncertain, real parameters $\{\delta_i\}_{i=1}^{n_p}$. Each parameter variation is assumed to be normalized such that $|\delta_i| \leq \theta_i$. Define $w_i := \delta_i u$ and rewrite the constraint on δ_i as an individual norm-bound constraint $\|w_i\| \leq \theta_i \|u\|_2$. Definition 7.4.1 can be modified to consider inner optimization over each w_i. The worst-case w_i^* for the inner optimization is then given by an expression similar to that of (7.11) using each θ_i and $L_{\tilde{g}_i} h(x)$.

Remark 7.4.1. *If $n_p = 1$ and $g_1(x) = g_0(x)$, then $\dot{x} = f_0(x) + g_0(x)\,(u + w)$, with $w = \delta_1 u$ and $|\delta_1| \leq \theta_1$. These dynamics are similar to those in* (7.8)*. This special case corresponds to a gain variation at the plant input as appears in the classical gain margin calculation.*

7.5 Numerical Example: Vehicle Lateral Control

Consider a vehicle being driven on a straight road that must avoid a stationary obstacle. The lateral dynamics of vehicle are linearized at a constant longitudinal speed to obtain

the following linear time-invariant (LTI) model [145].

$$\dot{x}_{lat}(t) = \mathbf{A}x_{lat}(t) + \mathbf{B}(u(t) + w(t)), \;\; x_{lat}(0) = x_{lat,0} \qquad (7.22)$$
$$|w(t)| \leq \theta \, |u(t)|$$

where $x_{lat}(t) = \left[\, e(t) \;\; \dot{e}(t) \;\; \psi(t) \;\; \dot{\psi}(t) \, \right]^\top \in \mathbb{R}^4$ is the linearized lateral state and the control $u(t) \in \mathbb{R}$ is the front wheel steering angle input. The model from [145] is slightly modified to include the uncertain input $w(t) \in \mathbb{R}$, which represents nonlinearities (e.g. allowable saturation) and/or time-varying uncertainties. Here, e is the lateral distance to the lane center and ψ is the vehicle heading relative to the path. The longitudinal dynamics $x_{long} = \left[\begin{smallmatrix} s \\ \dot{s} \end{smallmatrix} \right]$ are not controlled. The system states consists of both lateral and longitudinal dynamics, i.e. $x = \left[\begin{smallmatrix} x_{lat} \\ x_{long} \end{smallmatrix} \right]$. The state-space matrices [145] for the lateral dynamics are given by:

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

The vehicle parameters from [145] are given as follows:

m	Vehicle mass	1.67×10^3 kg
I_z	Vehicle moment of inertia	2.1×10^3 kg-m^2
a	Distance from vehicle CG to front axle	0.99 m
b	Distance from vehicle CG to rear axle	1.7 m
U	Longitudinal velocity	28 m/s
$C_{\alpha f}$	Front cornering stiffness	-1.23×10^5 N/rad
$C_{\alpha r}$	Rear cornering stiffness	-1.042×10^5 N/rad

Table 7.1: Vehicle Parameters

A baseline state-feedback controller was designed using linear quadratic regulator with cost matrices $Q = diag(10, 1, \frac{1}{30}, 1)$ and $R = 5$. This was implemented to track the reference command $r(t) \in \mathbb{R}^4$ as:

$$u_0 = K \cdot (r - x_{lat}), \;\; \text{where } K = \left[\, 1.41 \;\; 0.41 \;\; 3.30 \;\; 0.24 \, \right].$$

This differs slightly from the feedback diagram in Figure 7.1 due to the inclusion of the reference command, i.e. the baseline controller is of the form $u_0 = k_0(x_{lat}, r)$.

A stationary obstacle of radius d is assumed at the origin. The safe set $\mathcal{C}$ is defined by Equation (7.3) with $h(x) = e^2 + s^2 - d^2 \geq 0$ where d is chosen based on geometries of the vehicle and the obstacle. The first and second time-derivatives of h along a trajectory of state x are given by:

$$\frac{d}{dt}h(x(t), w(t), u(t)) = 2e\dot{e} + 2s\dot{s}$$
$$\frac{d^2}{dt^2}h(x(t), w(t), u(t)) = 2e\ddot{e}(u, w) + 2\dot{e}^2 + 2\dot{s}^2$$

Since the control input u and uncertain input w both appear in $\ddot{e}$, the system has relative degree two. This requires the robust exponential CBF as discussed in Section 7.4.5 to ensure safety. The set $\mathcal{U} \equiv \mathbb{R}$ is considered for simplicity.

The initial state of the vehicle is $x_{lat,0} = \begin{bmatrix} 2 & 0 & 0 & 0 \end{bmatrix}^\top$ with $x_{long,0} = \begin{bmatrix} -20 \\ 28 \end{bmatrix}$. . The safe distance d is chosen as 3 meters. The reference trajectory is selected to track the center of the lane, i.e. $r(t) = 0 \in \mathbb{R}^4$. The safety filter is designed using the RECBF-SOCP approach and is implemented on MATLAB using the `coneprog` function.

Let the uncertainty level be $\theta = 0.5$ for the simulation study. Figure 7.3 shows the simulation results for the uncertain plant with the worst-case uncertainty as in Equation (7.11). The shaded red circle represents the unsafe region. The ECBF and RECBF design poles are chosen to be two repeated poles at -30. Note that the nominal LQR controller runs into the obstacle due to its lack of safety consideration. The (nominal) ECBF-QP safety filter does not explicitly account for the uncertainty. Thus, its trajectory slightly violates the safety requirement around $s = 0$ or $t \approx 0.7$ seconds. The RECBF-SOCP trajectory avoids the obstacle successfully by choosing cautious control input to account for the uncertainty.

Next, the simulations are performed on the nominal plant. In other words, the RECBF-SOCP controller is designed assuming $\theta > 0$, but the simulations are performed on a nominal plant without the nonlinearity. Figure 7.4 shows the RECBF-SOCP simulations with the same initial condition, but with the safety filter designed at different assumed values for the uncertainty bound θ. Only the zoomed region around the obstacle is shown. It is observed that, as the model uncertainty level in design increases from

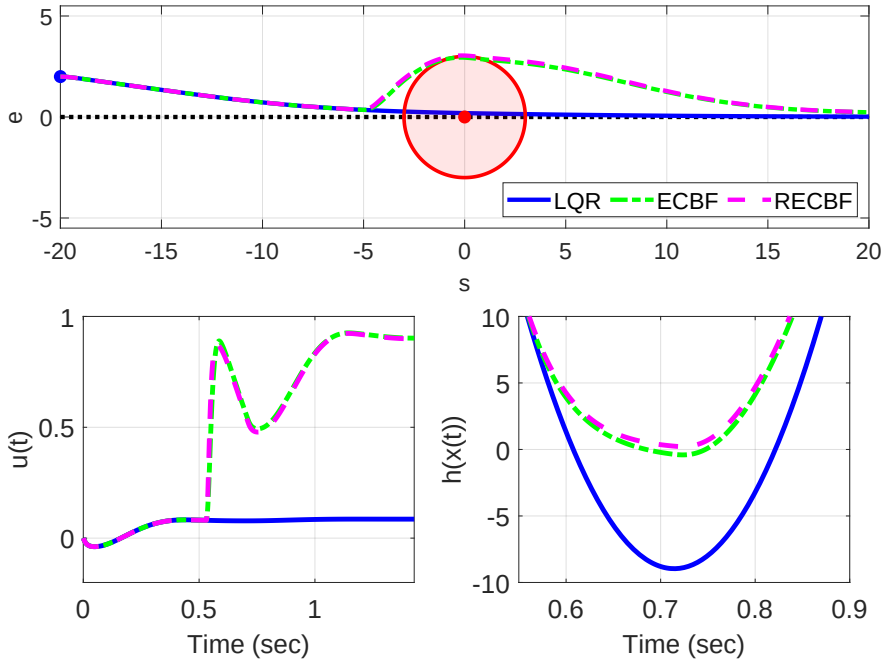

Figure 7.3: LQR, ECBF, and RECBF simulations with worst-case plant.

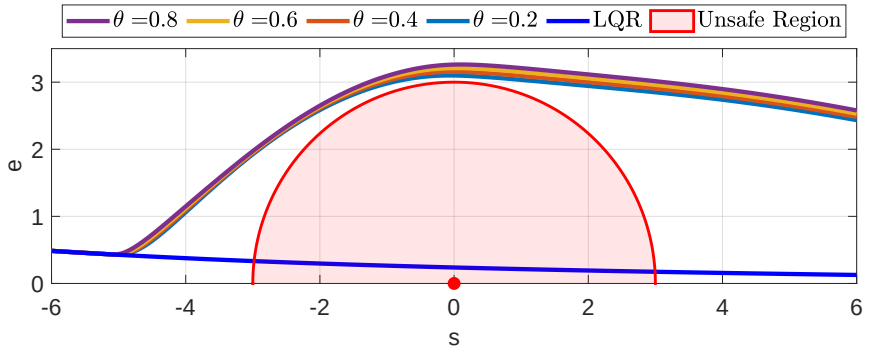

Figure 7.4: RECBF trajectories with varying uncertainty level θ.

$\theta = 0.2$ to 0.8, the RECBF-SOCP generates more cautious trajectories around the obstacle. The proposed design explicitly considers the uncertainty at the plant input and yields robustly-safe control actions.

7.6 Summary

This chapter presented a robust control barrier function approach to handle sector-bounded uncertainties at the plant input. The proposed optimization was solved online to demonstrate robust safety in a lateral vehicle control example.

Chapter 8

Conclusions and Future Research

8.1 Conclusions

This dissertation focused on advancing finite horizon LTV robustness analysis and synthesis tools from both a theoretical and computational standpoint. As discussed in Chapters 1 and 2, the critical application of interest is the analysis of a missile engagement where an interceptor missile engages a threat. We assumed that a representative engagement model is available that roughly captures reality. As noted before, the nonlinear worst-case analysis problem is non-convex due to many complicating elements such as disturbances, unmodeled dynamics, parametric uncertainties, etc. There are no efficient tools available that provide reliable performance guarantees for such a general problem. Thus, the current practice is to perform nonlinear Monte-Carlo analysis to provide statistical guarantees on the performance. We recommend that before running Monte-Carlo simulations, an engineer or analyst should first consider performing a worst-case LTV analysis as it is more computationally efficient. The proposed analysis also returns the worst-case disturbance, uncertainty, or parameter combinations, which should be used as a guide in performing the Monte-Carlo analysis. This approach can help find edge-cases sooner in the process. The insights from LTV analysis can also guide the design improvements before running Monte-Carlo simulations. For example, autopilot gains or crossover frequency can be tuned further to better reject the disturbances obtained through LTV analysis. Furthermore, it is possible to study key system nonlinearities (e.g., actuator saturation) using proposed finite horizon analysis.

Chapter 3 focused on nominal performance and presented two finite-horizon metrics of interest. These include the finite horizon H_2 norm and different induced gains ($\mathcal{L}_2$-to-$\mathcal{L}_2$, $\mathcal{L}_2$-to-Euclidean, and various combinations of them). For the computation of induced gain, we proposed a new algorithm that is 8 to 10 times faster than the RDE bisection and provides guaranteed bounds to within a desired numerical tolerance for higher-order systems. This algorithm combines the complementary benefits of the power iteration and RDE bisection. We also proved the convergence of the power iteration for the LTV system. Finally, the chapter concluded with Gramian based computational conditions for the induced $\mathcal{L}_2$-to-Euclidean gain. These conditions allow us to compute a tube around the trajectory efficiently.

Chapter 4 focused on robust performance by considering the impact of model uncertainties. This chapter considers the worst-case induced gain as a metric of performance. The robustness analysis framework relied on the time-domain dissipation inequalities and integral quadratic constraints (IQCs) to compute an upper bound on the worst-case gain. A numerical algorithm from [10] was used to compute an upper bound. In addition, this chapter discussed the computation of lower bounds using sample-based analysis and power iterations. Finally, the first-order trajectory sensitivity approach was proposed to analyze the impact of parametric uncertainties.

Chapter 5 considered the robust synthesis problem to design a controller that minimizes the impact of worst-case uncertainties and disturbances. In general, this problem is non-convex. We developed a new algorithm that iterates on the nominal synthesis and IQC analysis steps. These steps are linked by constructing a scaled plant that uses a time-varying IQC factorization. We also proved that the worst-case gain upper bound in the proposed algorithm is non-increasing. The effectiveness of the proposed design was demonstrated on a nonlinear two-link robot example by studying robustness and performance trade-offs.

Chapter 6 revisited the single interceptor and threat engagement to present a metric that aims to maximize the perpendicular deviation to the Line-of-Sight at the nominal intercept time. The LTV analysis provided a worst-case disturbance, which was scaled to induce a desired miss-distance at the final time. This led to the interceptor missing the threat. This disturbance is further studied in a high-fidelity nonlinear simulation to

assess its impact. We used nonlinear Monte-Carlo simulations to complement the worst-case LTV analysis. Finally, the trajectory sensitivity approach was used to evaluate the effect of uncertainties in interceptor aerodynamic parameters.

Chapter 7 presented safety-critical control using robust control barrier functions. The specific problem considered sector-bounded uncertainties representing static nonlinearities or time-varying memoryless uncertainties at the plant input. The optimization problem was formulated to prioritize robust safety. The proposed approach was demonstrated on a lateral vehicle control example to study robust safety.

8.2 Future Research

The investigations pursued in this thesis leave open several topics for future research.

8.2.1 Robust Performance

We focused on the worst-case gain of the system as a metric for robust performance; however, the theoretical framework can also be extended to the stochastic analysis problem (e.g., worst-case H_2 norm). The IQC upper bound algorithm in [10] is not efficient for higher-order systems. Future work is required to improve computational speed for such analysis and make these tools numerically reliable. Moreover, as noted in Chapter 4, the worst-case gain lower bound computation also requires investigation. Specifically, the power iteration algorithm should further be investigated for the convergence proof and construction of a worst-case uncertainty. Finally, a shift in the nominal trajectory [31] should further be investigated to perform a reliable linearized analysis to match more closely with nonlinear performance. Other areas for future research include alternative methods for worst-case analysis and exploring successive linearization methods.

8.2.2 Construction of a Worst-Case Uncertainty

The sample-based analysis (Section 4.5.1) also provides a sample of the worst-case uncertainty, which can be studied in a nonlinear high-fidelity model. This allows us to understand the worst-case scenario better and assess its likelihood in practice. However, the sample-based analysis is limited to the sampled uncertainty space. The analysis may be compromised if specific bad uncertainties are not included in the sample space.

Moreover, how to sample a particular class of uncertainty is not well understood as it requires domain expertise. In contrast, the LTV power iteration (Section 4.5.2) provides the instance of a worst-case uncertain input-output pair, but the uncertainty itself is not known. The following subsection briefly summarizes the worst-case uncertainty construction problem and technical challenges therein.

Problem Statement: Consider the interconnection $\mathcal{F}_u(N, \Delta)$. As noted earlier, the power iteration algorithm for the LTV system outputs the worst-case uncertain input v_o and the output w_o of the uncertainty Δ. The goal is to construct the uncertainty Δ such that $w = \Delta(v)$ and that the uncertainty satisfies the following additional conditions:

1. Δ is causal and $\|\Delta\|_{2\to 2,[0,T]} \leq 1$, i.e. we must satisfy:

$$\int_0^t w(\tau)^\top w(\tau)\, d\tau \leq \int_0^t v(\tau)^\top v(\tau)\, d\tau, \qquad \forall t \in [0, T]. \tag{8.1}$$

2. Δ interpolates the specific input signal v_o to the specific output signal w_o.

3. $\Delta(0) = 0$.

Technical Challenges: There are two main challenges: First, we only enforced the norm-bound constraint $\|v\|_{2,[0,T]} = \|w\|_{2,[0,T]}$ on the signals v and w while running the power iteration algorithm. The alignment conditions used in Algorithm 4 do not enforce the causality constraints given by inequality (8.1). Thus, the output signals v_o and w_o of Algorithm 4 cannot be directly used to construct the worst-case uncertainty Δ with causal input-output behavior. The power iteration algorithm must be modified (or some alternative approach must be used) to handle such constraints explicitly. Similarly, if Δ is an LTI uncertainty, then linearity and time-invariance of Δ must be enforced by appropriate alignment conditions on signals v and w. Second, there is no systematic way of constructing the specific uncertainty Δ once the worst-case signals v_o and w_o are obtained from power iterations. Exiting results on time-domain model validation [146] and Nevanlinna-Pick interpolation for time-varying input-output maps [147] should be further investigated to construct a causal and realizable uncertainty Δ.

8.2.3 Robust Synthesis

There are a few new directions to explore. First, we used a block diagonal assumption for the IQC filter and decision variables (Assumption 5.4.1). Future work will consider relaxing this assumption to include full block IQC multipliers in the robust synthesis. Second, note that the proposed method allows time-varying uncertainty and performance weights. This is useful in many scenarios, where the uncertainty or performance requirements are not evenly spread across the time horizon. Future work in this area is required to exploit the full potential of this method. Moreover, we recognize that the proposed robust synthesis is computationally expensive. Future research is needed to speed up the numerical computations for finite horizon synthesis.

8.2.4 Analysis of Optimization-based Controllers

We can use the theoretical and computational tools developed in this dissertation to analyze the performance of optimization-based control methods (e.g., MPC or adaptive control). For example, we ignored the external disturbances in the proposed safety-critical control using RCBF-SOCP. Future work can study the impact of worst-case disturbances in such safety-critical situations. Moreover, we can also perform analysis of neural network controllers trained using data-driven or reinforcement learning methods. Future work can use the insights from LTV analysis to guide the learning process and synthesize controllers with robustness properties.

8.2.5 Other Applications

The proposed finite horizon LTV analysis tools can be helpful for many engineering applications to be explored in future work. These include surgical robotics, power systems, renewable energy, autonomous systems, to name a few. For example, a flexible endoscope is used in robotic bronchoscopy [148] to collect malignant tissue samples from lung nodules. Suppose we have a high-fidelity dynamic model of the scope. In that case, the proposed analysis can be used to assess the impact on scope trajectory due to patient breathing patterns (treated as external disturbances). The tools developed in this thesis can aid engineering V&V analysis and potentially support regulatory discussions.

References

[1] W. Tan, A. K. Packard, and G. J. Balas, "Quasi-LPV modeling and LPV control of a generic missile," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*, vol. 5, pp. 3692–3696, IEEE, 2000.

[2] K. S. Schweidel, A. K. Packard, M. Arcak, P. J. Seiler, and D. O. Philbrick, "Classifier-based supervisory control with application to threat engagement," in *2020 American Control Conference (ACC)*, pp. 3757–3762, IEEE, 2020.

[3] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robot Manipulation*. CRC Press, 1994.

[4] H. W. Bode *et al.*, "Network analysis and feedback amplifier design," 1945.

[5] P. Seiler, A. Packard, and P. Gahinet, "An introduction to disk margins [lecture notes]," *IEEE Control Systems Magazine*, vol. 40, no. 5, pp. 78–95, 2020.

[6] A. Packard and J. Doyle, "The complex structured singular value," *Automatica*, vol. 29, no. 1, pp. 71–109, 1993.

[7] G. Balas, R. Chiang, A. Packard, and M. Safonov, "Robust Control Toolbox 3," *The Mathworks, Inc., Natick, MA*, 2005.

[8] J. E. Tierno, R. M. Murray, J. C. Doyle, and I. M. Gregory, "Numerically efficient robustness analysis of trajectory tracking for nonlinear systems," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 4, pp. 640–647, 1997.

[9] P. Seiler, J. Buch, R. M. Moore, C. Meissen, M. Arcak, and A. Packard, "LTVTools, A MATLAB Toolbox for Linear Time-Varying Systems." `https://z.umn.edu/LTVTools`, 2018.

[10] P. Seiler, R. M. Moore, C. Meissen, M. Arcak, and A. Packard, "Finite horizon robustness analysis of LTV systems using integral quadratic constraints," *Automatica*, vol. 100, pp. 135–143, 2019.

[11] J. Buch and P. Seiler, "Finite horizon robust synthesis," in *2020 American Control Conference (ACC)*, pp. 1551–1556, IEEE, 2020.

[12] J. Buch and P. Seiler, "Finite horizon robust synthesis using integral quadratic constraints," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 8, pp. 3011–3035, 2021.

[13] J. Buch, M. Arcak, and P. Seiler, "An efficient algorithm to compute norms for finite horizon, linear time-varying systems," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1597–1602, 2020.

[14] J. Buch, S. C. Liao, and P. Seiler, "Robust control barrier functions with sector-bounded uncertainties," *IEEE Control Systems Letters*, 2021.

[15] S. Eastman and D. Estep, "A power method for nonlinear operators," *Applicable Analysis*, vol. 86, no. 10, pp. 1303–1314, 2007.

[16] R. W. Brockett, *Finite Dimensional Linear Systems*, vol. 74. SIAM, 2015.

[17] R. Moore, "Finite horizon robustness analysis using integral quadratic constraints," Master's thesis, University of California, Berkeley, 2015.

[18] R. Holland and P. Young, "A skew μ lower bound," in *Proceedings of the 2002 American Control Conference*, IEEE, 2002.

[19] R. Holland, P. Young, and C. Zhu, "Development of a skew μ lower bound," *International Journal of Robust and Nonlinear Control*, 2005.

[20] P. M. Young and J. C. Doyle, "A lower bound for the mixed *mu* problem," *IEEE Transactions on Automatic Control*, vol. 42, no. 1, pp. 123–128, 1997.

[21] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall, New Jersey, 1996.

[22] G. E. Dullerud and F. Paganini, *A Course in Robust Control Theory: A Convex Approach*, vol. 36. Springer Science and Business Media, 2013.

[23] M. Green and D. J. Limebeer, *Linear Robust Control*. Courier Corporation, 2012.

[24] W. Liren, "A Monte-Carlo simulation of guidance accuracy evaluation," tech. rep., National Air Intelligence Center Wright-Patterson AFB, OH, 1996.

[25] P. He, Y. Wu, and X. Luo, "Study on the effectiveness of multiple ballistic missiles penetrating anti-missile system based on monte-carlo simulation," *Journal of the Academy of Equipment Command & Technology*, vol. 4, 2007.

[26] J. Hanson and B. Beard, *Applying Monte Carlo simulation to launch vehicle design and requirements analysis*. National Aeronautics and Space Administration, Marshall Space Flight Center, 2010.

[27] H. Yin, A. Packard, M. Arcak, and P. Seiler, "Reachability analysis using dissipation inequalities for uncertain nonlinear systems," *Systems and Control Letters*, vol. 142, p. 104736, 2020.

[28] S. Geng and I. A. Hiskens, "Second-order trajectory sensitivity analysis of hybrid systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 5, pp. 1922–1934, 2019.

[29] M. Weiss, M. Rol, W. Falkena, and C. Scherer, "Guidance performance analysis in the presence of model uncertainties," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, p. 6786, 2007.

[30] A. Iannelli, P. Seiler, and A. Marcos, "Worst-case disturbances for time-varying systems with application to flexible aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 6, pp. 1261–1271, 2019.

[31] K. S. Schweidel, J. R. Buch, P. J. Seiler, and M. Arcak, "Computing worst-case disturbances for finite-horizon linear time-varying approximations of uncertain systems," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1753–1758, 2020.

[32] B. M. Wade, "Creating surrogate models for an air and missile defense simulation using design of experiments and neural networks," *The Journal of Defense Modeling and Simulation*, p. 1548512919877987, 2019.

[33] H. Yin, *Robustness in Nonlinear and Learning Based Control*. PhD thesis, University of California, Berkeley, 2021.

[34] J. Doyle, A. Packard, and K. Zhou, "Review of LFTs, LMIs, and μ," 1991.

[35] P. Zarchan, *Tactical and strategic missile guidance*. American Institute of Aeronautics and Astronautics, Inc., 2012.

[36] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd, "Basic principles of homing guidance," *Johns Hopkins APL Technical Digest*, vol. 29, no. 1, pp. 25–41, 2010.

[37] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd, "Modern homing missile guidance theory and techniques," *Johns Hopkins APL technical digest*, vol. 29, no. 1, pp. 42–59, 2010.

[38] E. J. Ohlmeyer, "Analysis of third-order autopilots for missile flight control.," 1990.

[39] J. Imae and G. Wanyoike, "H_∞ norm computation for LTV systems using nonlinear optimal control algorithms," *International Journal of Control*, vol. 63, no. 1, pp. 161–182, 1996.

[40] Y. Yamamoto and P. P. Khargonekar, "Frequency response of sampled-data systems," *IEEE Transactions on Automatic Control*, vol. 41, no. 2, pp. 166–176, 1996.

[41] N. Bruinsma and M. Steinbuch, "A fast algorithm to compute the H_∞-norm of a transfer function matrix," *Systems and Control Letters*, vol. 14, no. 4, pp. 287–293, 1990.

[42] G. E. Dullerud, "Computing the L_2-induced norm of a compression operator," *Systems and Control Letters*, vol. 37, no. 2, pp. 87–91, 1999.

[43] B. Bamieh, "On computing the L_2-induced norm of finite-horizon systems," in *42nd IEEE International Conference on Decision and Control*, vol. 2, pp. 1860–1868 Vol.2, 2003.

[44] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*, vol. 2. Citeseer, 2007.

[45] P. P. Khargonekar, K. M. Nagpal, and K. R. Poolla, "H_∞ control with transients," *SIAM Journal on Control and Optimization*, vol. 29, no. 6, pp. 1373–1393, 1991.

[46] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[47] G. Golub and C. Van Loan, *Matrix Computations*. The Johns Hopkins University Press, 2013.

[48] J. W. Demmel, *Applied Numerical Linear Algebra*. SIAM, 1997.

[49] A. Iannelli, P. Seiler, and A. Marcos, "Worst-case disturbances for time-varying systems with application to flexible aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 6, pp. 1261–1271, 2019.

[50] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*. Springer Science and Business Media, 2010.

[51] B. N. Parlett and W. G. Poole, Jr, "A geometric theory for the QR, LU and power iterations," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 389–412, 1973.

[52] G. W. Stewart, *Matrix Algorithms: Vol II: Eigensystems*. SIAM, 2001.

[53] N. Young, *An Introduction to Hilbert space*. Cambridge University Press, 1988.

[54] J. K. Hunter and B. Nachtergaele, *Applied Analysis*. World Scientific Publishing Company, 2001.

[55] Y. Ito, T. Hagiwara, H. Maeda, and M. Araki, "Bisection algorithm for computing the frequency response gain of sampled-data systems-infinite-dimensional congruent transformation approach," *IEEE Transactions on Automatic Control*, vol. 46, no. 3, pp. 369–381, 2001.

[56] E. Kreyszig, *Introductory functional analysis with applications*. Wiley New York, 1978.

[57] T. Kato, *Perturbation theory for linear operators*. Springer Science and Business Media, 2013.

[58] W. Rudin, *Functional analysis*. McGraw-Hill, New York, 1973.

[59] D. G. Luenberger, *Optimization by vector space methods*. John Wiley and Sons, 1997.

[60] D. A. Wilson, "The Hankel operator and its induced norms," *International Journal of Control*, vol. 42, no. 1, pp. 65–70, 1985.

[61] W. W. Lu and G. J. Balas, "A comparison between Hankel norms and induced system norms," *IEEE transactions on automatic control*, vol. 43, no. 11, pp. 1658–1662, 1998.

[62] M. Corless, G. Zhu, and R. Skelton, "Improved robustness bounds using covariance matrices," in *Proceedings of the 28th IEEE Conference on Decision and Control*, IEEE, 1989.

[63] D. A. Wilson, "Convolution and Hankel operator norms for linear systems," *IEEE Transactions on Automatic Control*, 1989.

[64] M. A. Rotea, "The generalized H_2 control problem," *Automatica*, vol. 29, no. 2, pp. 373–385, 1993.

[65] A. Megretski, "Interpretations for standard optimization setup," 2004.

[66] V. Balakrishnan, "Lyapunov functionals in complex μ analysis," *IEEE Transactions on Automatic Control*, vol. 47, no. 9, pp. 1466–1479, 2002.

[67] F. Paganini and E. Feron, "LMI methods for robust H_2 analysis: a survey with comparions," LMI Methods in optimisation, identification and control, Seminar, Compiegne, 1999.

[68] V. Yakubovich, "Frequency conditions for the absolute stability of control systems with several nonlinear or linear nonstationary blocks," *Avtomatika i telemekhanika*, vol. 6, pp. 5–30, 1967.

[69] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE Transactions on Automatic Control*, vol. 42, no. 6, pp. 819–830, 1997.

[70] F. Biertümpfel, H. Pfifer, and S. Bennani, "Finite horizon worst case analysis of launch vehicles," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 31–36, 2019.

[71] F. Biertümpfel and H. Pfifer, "Worst case gain computation of linear time-varying systems over a finite horizon," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 952–957, 2018.

[72] F. Biertümpfel, S. Bennani, and H. Pfifer, "Finite time horizon analysis of launch vehicles under mass and thrust uncertainty," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 7452–7457, 2020.

[73] F. Biertümpfel and H. Pfifer, "Finite horizon touchdown analysis of autolanded aircraft under crosswind," *IFAC-PapersOnLine*, vol. 54, no. 8, pp. 124–129, 2021.

[74] J. E. Tierno, R. M. Murray, and J. C. Doyle, "An efficient algorithm for performance analysis of nonlinear control systems," in *Proceedings of 1995 American Control Conference*, IEEE, 1995.

[75] O. Narvaez-Aroche, A. Packard, P.-J. Meyer, and M. Arcak, "Reachability analysis for robustness evaluation of the sit-to-stand movement for powered lower limb orthoses," in *Dynamic systems and control conference*, vol. 51890, p. V001T07A006, American Society of Mechanical Engineers, 2018.

[76] H. Choi, P. J. Seiler, and S. V. Dhople, "Propagating uncertainty in power-system dae models with semidefinite programming," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3146–3156, 2016.

[77] H. Pfifer and P. Seiler, "Robustness analysis with parameter-varying integral quadratic constraints," in *American Control Conference (ACC)*, pp. 138–143, IEEE, 2015.

[78] J. Veenman, C. W. Scherer, and H. Köroğlu, "Robust stability and performance analysis based on integral quadratic constraints," *European Journal of Control*, vol. 31, pp. 1–32, 2016.

[79] P. Seiler, "Stability analysis with dissipation inequalities and integral quadratic constraints," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1704–1709, 2014.

[80] J. C. Willems, "Dissipative Dynamical Systems Part I: General Theory," *Archive for rational mechanics and analysis*, vol. 45, no. 5, pp. 321–351, 1972.

[81] Schaft, *L_2-gain and Passivity in Nonlinear Control*. Springer-Verlag, 1999.

[82] H. K. Khalil, *Nonlinear systems*, vol. 3. Prentice hall, NJ, 2002.

[83] J. C. Doyle, "Guaranteed margins for LQG regulators," *IEEE Transactions on automatic Control*, vol. 23, no. 4, pp. 756–757, 1978.

[84] M. B. Subrahmanyam, *Finite Horizon H_∞ and Related Control Problems*. Springer Science and Business Media, 2012.

[85] M. R. Tucker, *Continuous H_∞ and discrete time-varying finite horizon robust control with industrial applications*. PhD thesis, University of Leicester, 1998.

[86] A. Marcos and S. Bennani, "LPV modeling, analysis and design in space systems: Rationale, objectives and limitations," in *AIAA Guidance, Navigation, and Control Conference*, pp. AIAA 2009–5633, 2009.

[87] J. Veenman, H. Köroğlu, and C. Scherer, "Analysis of the controlled NASA HL20 atmospheric re-entry vehicle based on dynamic IQCs," in *AIAA Guidance, Navigation, and Control Conference*, pp. AIAA 2009–5637, 2009.

[88] J. Doyle, K. Lenz, and A. Packard, *Design examples using μ-synthesis: Space shuttle lateral axis FCS during reentry*. Springer, 1987.

[89] G. Balas and A. Packard, "Design of robust, time-varying controllers for missile autopilots," in *Proceedings of the first IEEE Conference on Control Applications*, pp. 104–110, IEEE, 1992.

[90] A. Packard, J. Doyle, and G. Balas, "Linear, multivariable robust control with a μ perspective," *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, no. 2B, pp. 426–438, 1993.

[91] J. Veenman and C. W. Scherer, "IQC-synthesis with general dynamic multipliers," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 17, pp. 3027–3056, 2014.

[92] I. Petersen, V. Ugrinovskii, and A. Savkin, *Robust Control Design Using H_∞ Methods*. Springer, 2000.

[93] G. Tadmor, "Worst-case design in the time domain: The maximum principle and the standard H_∞ problem," *Mathematics of Control, Signals and Systems*, vol. 3, no. 4, pp. 301–324, 1990.

[94] D. J. Limebeer, B. D. Anderson, P. P. Khargonekar, and M. Green, "A game theoretic approach to H_∞ control for time-varying systems," *SIAM Journal on Control and Optimization*, vol. 30, no. 2, pp. 262–283, 1992.

[95] R. Ravi, K. M. Nagpal, and P. P. Khargonekar, "H_∞ control of linear time-varying systems: A state-space approach," *SIAM Journal on Control and optimization*, vol. 29, no. 6, pp. 1394–1413, 1991.

[96] K. Uchida and M. Fujita, "Finite horizon H_∞ control problems with terminal penalties," *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1762–1767, 1992.

[97] S. Lall and K. Glover, "Riccati differential inequalities: suboptimal H_∞ controllers for finite horizon time varying systems," in *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol. 1, pp. 955–956, IEEE, 1995.

[98] S. Wang, H. Pfifer, and P. Seiler, "Robust synthesis for linear parameter varying systems using integral quadratic constraints," *Automatica*, vol. 68, pp. 111–118, 2016.

[99] J. Veenman and C. W. Scherer, "On robust LPV controller synthesis: A dynamic integral quadratic constraint based approach," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 591–596, IEEE, 2010.

[100] R. O'Brien Jr and P. A. Iglesias, "Robust controller design for linear, time-varying systems," *European Journal of Control*, vol. 5, no. 2-4, pp. 222–241, 1999.

[101] C. Pirie and G. E. Dullerud, "Robust controller synthesis for uncertain time-varying systems," *SIAM Journal on Control and Optimization*, vol. 40, no. 4, pp. 1312–1331, 2002.

[102] M. Guelman, "The closed-form solution of true proportional navigation," *IEEE Transactions on Aerospace and Electronic Systems*, no. 4, pp. 472–482, 1976.

[103] D. Ghose, "True proportional navigation with maneuvering target," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 229–237, 1994.

[104] K. Becker, "Closed-form solution of pure proportional navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 3, pp. 526–533, 1990.

[105] A. Chakravarthy and D. Ghose, "Capturability of realistic generalized true proportional navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 1, pp. 407–418, 1996.

[106] A. Dhar and D. Ghose, "Capture region for a realistic TPN guidance law," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 3, pp. 995–1003, 1993.

[107] U. S. Shukla and P. R. Mahapatra, "The proportional navigation dilemma-pure or true?," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 2, pp. 382–392, 1990.

[108] F. Tyan, "Analysis of general ideal proportional navigation guidance laws," *Asian Journal of Control*, vol. 18, no. 3, pp. 899–919, 2016.

[109] G. Tarrant, "The method of adjoint systems and its application to guided missile noise studies," *Rept. PB186191, EASAMS, Camberly, UK*, 1969.

[110] M. Weiss, "Adjoint method for missile performance analysis on state-space models," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 2, pp. 236–248, 2005.

[111] A. Rajagopalan and D. Bucco, "Applications of adjoint theory to problems in aerospace and defence science," *Anaziam Journal*, vol. 51, pp. C697–C714, 2009.

[112] D. Bucco and M. Weiss, "Adjoint analysis of guidance systems with nonstandard inputs," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 9, pp. 1800–1809, 2015.

[113] T. He and W. Chen, "A new interpretation of adjoint method in linear time-varying system analysis," in *2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 58–63, IEEE, 2017.

[114] P. Zarchan, "Complete statistical analysis of nonlinear missile guidance systems-slam," *Journal of guidance and control*, vol. 2, no. 1, pp. 71–78, 1979.

[115] C. F. Price and R. S. Warren, "Performance evaluation of homing guidance laws for tactical missiles," tech. rep., Analytic Sciences Corp Reading, MA, 1973.

[116] P. Fan, J. Liu, and Z. Ouyang, "Miss distance algorithm of terminal ship-to-air missile based on vector operation," in *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, pp. 2572–2576, IEEE, 2014.

[117] M. Weiss and T. Shima, "Minimum effort pursuit/evasion guidance with specified miss distance," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 5, pp. 1069–1079, 2016.

[118] S. A. Murtaugh and H. E. Criel, "Fundamentals of proportional navigation," *IEEE spectrum*, vol. 3, no. 12, pp. 75–85, 1966.

[119] J. C. Knight, "Safety critical systems: challenges and directions," in *Proceedings of the 24th international conference on software engineering*, pp. 547–550, 2002.

[120] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[121] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conference (ECC)*, pp. 3420–3431, IEEE, 2019.

[122] E. Squires, P. Pierpaoli, and M. Egerstedt, "Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1656–1661, IEEE, 2018.

[123] P. Panyakeow and M. Mesbahi, "Decentralized deconfliction algorithms for unicycle UAVs," in *Proceedings of the 2010 American Control Conference*, pp. 794–799, IEEE, 2010.

[124] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.

[125] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Learning for Dynamics and Control*, pp. 708–717, PMLR, 2020.

[126] J. Choi, F. Castaneda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions," *arXiv:2004.07584*, 2020.

[127] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," *IEEE control systems letters*, vol. 3, no. 1, pp. 108–113, 2018.

[128] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[129] K. Garg and D. Panagou, "Robust control barrier and control Lyapunov functions with fixed-time convergence guarantees," in *American Control Conference*, 2021.

[130] J. Breeden and D. Panagou, "Robust control barrier functions under high relative degree and input constraints for satellite trajectories," *arXiv:2107.04094*, 2021.

[131] R. Takano, H. Oyama, and M. Yamakita, "Application of robust control barrier function with stochastic disturbance model for discrete time systems," *IFAC-PapersOnLine*, vol. 51, no. 31, pp. 46–51, 2018.

[132] Q. Nguyen and K. Sreenath, "Robust safety-critical control for dynamic robotics," *IEEE Transactions on Automatic Control*, 2021.

[133] P. Seiler, M. Jankovic, and E. Hellstrom, "Control barrier functions with unmodeled input dynamics using integral quadratic constraints," *IEEE Control Systems Letters*, 2021.

[134] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.

[135] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier-value functions for safety-critical control," *arXiv:2104.02808*, 2021.

[136] S. Dean, A. J. Taylor, R. K. Cosner, B. Recht, and A. Ames, "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions," in *CoRL*, 2020.

[137] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, 2015.

[138] G. C. Calafiore and L. El Ghaoui, *Optimization models*. Cambridge university press, 2014.

[139] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[140] J. Zeng, B. Zhang, Z. Li, and K. Sreenath, "Safety-critical control using optimal-decay control barrier function with guaranteed point-wise feasibility," in *American Control Conference (ACC)*, pp. 3856–3863, 2021.

[141] E. M. Bednarczuk and K. E. Rutkowski, "On Lipschitz continuity of projections onto polyhedral moving sets," *Applied Mathematics and Optimization*, pp. 1–29, 2020.

[142] N. Weaver, *Lipschitz algebras*. World Scientific, 2018.

[143] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference*, pp. 322–328, IEEE, 2016.

[144] R. Freeman and P. V. Kokotovic, *Robust nonlinear control design: state-space and Lyapunov techniques*. Springer Science and Business Media, 2008.

[145] A. Alleyne, "A comparison of alternative intervention strategies for unintended roadway departure (URD) control," *Vehicle System Dynamics*, vol. 27, no. 3, pp. 157–186, 1997.

[146] K. Poolla, P. Khargonekar, A. Tikku, J. Krause, and K. Nagpal, "A time-domain approach to model validation," *IEEE Transactions on automatic control*, vol. 39, no. 5, pp. 951–959, 1994.

[147] I. Gohberg, *Time-variant systems and interpolation*, vol. 56. Birkhäuser, 2012.

[148] C. F. Graetzel, A. Sheehy, and D. P. Noonan, "Robotic bronchoscopy drive mode of the Auris Monarch platform," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3895–3901, IEEE, 2019.

[149] B. Francis, "A course in H_∞ control theory," *Lecture notes in Control and Information Sciences*, vol. 88, p. R5, 1987.

[150] B. Molinari, "Nonnegativity of a quadratic functional," *SIAM Journal on Control*, vol. 13, no. 4, pp. 792–806, 1975.

[151] V. Kučera, "A review of the matrix riccati equation," *Kybernetika*, vol. 9, no. 1, pp. 42–61, 1973.

[152] G. J. Balas, A. Packard, P. J. Seiler, and A. Hjartarson, "LPVTools."

[153] A. Hjartarson, P. Seiler, and A. Packard, "LPVTools: A toolbox for modeling, analysis, and synthesis of parameter varying control systems," *IFAC-PapersOnLine*, vol. 48, no. 26, pp. 139–145, 2015.

Appendix A

Supplementary Material for Chapter 3

A.1 Compactness of Operator $G^{\sim}G$

Lemma A.1.1. *Assume that G has no feedthrough i.e. $D_I = 0$, then G, $G^{\sim}$, $G^{\sim}G$ and $GG^{\sim}$ are compact operators.*

Proof. Let $\Phi(t,\tau)$ denote the unique state transition matrix of the unforced system $\dot{x}(t) = A(t)\,x(t)$ satisfying the following differential equation:

$$\frac{d\Phi(t,\tau)}{dt} = A(t)\,\Phi(t,\tau), \qquad \Phi(\tau,\tau) = I_{n_x} \tag{A.1}$$

If $D_I = 0$ then G can be represented by a matrix valued kernel function as $e(t) = \int_0^t H(t,\tau)\,d(\tau)d\tau$, where $H(t,\tau) := C(t)\Phi(t,\tau)B(\tau)$ is an impulse response matrix. This impulse response matrix satisfies:

$$\int_0^T \int_0^T trace\left[H(t,\tau)H(t,\tau)^\top\right]\,dt\,d\tau < \infty \tag{A.2}$$

This bound follows from Theorem 3.3.1 in [23] and the connection to the finite-horizon H_2 norm of G. It follows that G is a Hilbert-Schmidt operator and hence compact (Theorem 8.8 of [53]). Finally, compactness of G implies that $G^{\sim}$, $G^{\sim}G$, and $GG^{\sim}$ are all compact (Theorems 4.8 & 4.10 in [57]). □

A.2 Generic Quadratic Cost

Chapter 3 considers an induced norm $\|G\|_{[0,T]}$ (defined by Equation 3.5) as a performance metric whereas a generic quadratic cost is considered in [10]. This appendix describes the equivalence between these two formulations. First, consider the finite-horizon linear-quadratic optimal control problem as follows:

$$J^*(\gamma) := \sup_{0 \neq d \in \mathcal{L}_2[0,T]} x(T)^\top F x(T) + \int_0^T \begin{bmatrix} x(t) \\ d(t) \end{bmatrix}^\top \begin{bmatrix} Q(t) & S(t) \\ S(t)^\top & R(t,\gamma) \end{bmatrix} \begin{bmatrix} x(t) \\ d(t) \end{bmatrix} dt$$
$$\text{s.t. } \dot{x}(t) = A(t)\,x(t) + B(t)\,d(t) \text{ and } x(0) = 0. \tag{A.3}$$

where $Q : [0,T] \to \mathbb{S}^{n_x}$, $S : [0,T] \to \mathbb{R}^{n_x \times n_d}$, $R : [0,T] \oplus \mathbb{R}^+ \to \mathbb{S}^{n_d}$ and $F \in \mathbb{R}^{n_x \times n_x}$. We assume $Q(t) \succeq 0$, for all $t \in [0,T]$ and $F \succeq 0$. Moreover, we assume a form $R(t,\gamma) = R_0(t) - \gamma^2 I_{n_d}$, where $\gamma > 0$, $R_0(t) \succeq 0$ and $R(t,\gamma) \prec 0$, for all $t \in [0,T]$. There are two directions to the equivalence. First, assume a system G is given as defined by Equation (3.1) and (3.4). Note that the induced norm $\|G\|_{[0,T]}$ is defined by the state-space matrices (A, B, C_I, C_E, D_I). Define (Q, S, R, F) as in Theorem 3.5.1. Then for any $\gamma > 0$, $\|G\|_{[0,T]} < \gamma$ if and only if $J^*(\gamma) < 0$. This is shown in Section 2 of [10]. Conversely, assume the generic quadratic cost defined by Equation (A.3) is given with cost matrices (Q, S, R, F) satisfying the assumptions above. If we further assume that $Q(t) - S(t)R_0(t)^{-1}S(t)^\top \succ 0$ then we can perform the following factorization:

$$\begin{bmatrix} Q(t) & S(t) \\ S(t)^\top & R_0(t) \end{bmatrix} = \begin{bmatrix} C_I(t)^\top \\ D_I(t)^\top \end{bmatrix} \begin{bmatrix} C_I(t) & D_I(t) \end{bmatrix} \tag{A.4}$$

In addition, define $C_E := F^{\frac{1}{2}}$. Then the generic quadratic cost is re-written as:

$$J^*(\gamma) = \sup_{0 \neq d \in \mathcal{L}_2[0,T]} e_E(T)^\top e_E(T) + \int_0^T e_I(t)^\top e_I(t)\,dt - \gamma^2 \int_0^T d(t)^\top\, d(t)\,dt$$
$$\text{s.t. Equation (3.1), (3.4) and } x(0) = 0. \tag{A.5}$$

This cost satisfies $J^*(\gamma) < 0$ if and only if $\|G\|_{[0,T]} < \gamma$.

Appendix B

Supplementary Material for Chapter 5

B.1 Finite Horizon Factorization

For infinite horizon LTI systems, spectral factorization results are found in standard robust control textbooks [21, 22, 149]. The following lemma provides a time-varying finite horizon generalization of this result.

Lemma B.1.1. *Consider an LTV system* $\Psi : \mathcal{L}_2^{n_d}[0,T] \to \mathcal{L}_2^{n_e}[0,T]$ *be given with state-space realization as follows:*

$$\begin{aligned} \dot{x}(t) &= A(t)\,x(t) + B(t)\,d(t) \\ e(t) &= C(t)\,x(t) + D(t)\,d(t) \end{aligned} \tag{B.1}$$

with $x \in \mathbb{R}^{n_x}$, $e \in \mathbb{R}^{n_e}$, $d \in \mathbb{R}^{n_d}$ *and* $D(t)$ *is full column rank* $\forall t \in [0,T]$. *Let* $M : [0,T] \to \mathbb{S}^{n_e}$ *be a given piecewise continuous matrix valued function with* $M(t) \succ 0, \forall t \in [0,T]$. *Let* $Q : [0,T] \to \mathbb{S}^{n_x}$, $S : [0,T] \to \mathbb{R}^{n_x \times n_d}$, $R : [0,T] \to \mathbb{S}^{n_d}$ *be defined as follows.*

$$Q := C^\top M C, \quad S := C^\top M D, \quad R := D^\top M D \tag{B.2}$$

with $R(t) \succ 0, \forall t \in [0,T]$. *The following statements hold.*

1. *There exist a differentiable function* $X : [0, T] \to \mathbb{S}^n$ *such that* $X(T) = 0$ *and*

$$\dot{X} + A^\top X + XA + Q - (XB + S)R^{-1}(XB + S)^\top = 0 \tag{B.3}$$

2. $\Phi := \Psi^\sim M\Psi$ *has a finite horizon factorization* $\Phi = U^\sim U$ *where* U *is square invertible LTV system defined on* $[0, T]$ *with the following state-space realization:*

$$U = \left[\begin{array}{c|c} A & B \\ \hline W^{-T}(B^\top X + S^\top) & W \end{array}\right] \tag{B.4}$$

where $R(t) = W(t)^\top W(t)$, $\forall t \in [0, T]$.

Proof. Since $R(t) \succ 0$ and by Schur complement lemma $Q(t) - S(t)R(t)^{-1}S(t)^\top \succ 0, \forall t \in [0, T]$, the RDE does not have a finite escape time and thus always have a bounded unique solution regardless of the boundary condition (Corollary 2.3 of [150], Theorem 8 in [151]). Further, it can be verified that the time-varying state-space realization of $\Psi^\sim M\Psi$ is related to that of a system $U^\sim U$ by a similarity transformation matrix $\left[\begin{smallmatrix} I & 0 \\ X(t) & I \end{smallmatrix}\right]$. □

Appendix C

Supplementary Material for Chapter 7

C.1 Worst-Case Uncertain Input

Let $\lambda \geq 0$ be a Lagrange multiplier to rewrite the inner optimization problem (7.10) as follows:

$$\inf_{w} \sup_{\lambda \geq 0} \left[p(x) + L_{\tilde{g}} h(x)(u + w) - \lambda(\theta^2 u^\top u - w^\top w) \right] \geq 0$$

Define the cost in bracket as $J(u, w, \lambda)$. Strong duality holds because the objective and constraints are both convex [139], thus $\sup_{\lambda \geq 0} \inf_w J(u, w, \lambda) \geq 0$. Complete the square in w by adding and subtracting the term $\frac{1}{4\lambda} L_{\tilde{g}} h(x) L_{\tilde{g}} h(x)^\top$ in the cost function $J(u, w, \lambda)$. Then, minimizing over w yields the worst-case uncertain input as:

$$w^* = -\frac{1}{2\lambda^*} L_{\tilde{g}} h(x)^\top \tag{C.1}$$

Plug in w^* in the cost function $J(u, w, \lambda)$ to obtain the cost function $J(u, \lambda)$ as follows:

$$J(u, \lambda) := p(x) + L_{\tilde{g}} h(x) u - \lambda \theta^2 u^\top u - \frac{1}{4\lambda} L_{\tilde{g}} h(x) L_{\tilde{g}} h(x)^\top$$

Use the first-order optimality condition to solve for optimal Lagrange multiplier λ^* as follows:

$$\lambda^* = \frac{\|L_{\tilde{g}}h(x)\|_2}{2\theta\|u\|_2} \tag{C.2}$$

Plug λ^* in (C.1) to obtain the expression for $w^*(u)$ as in (7.11).

C.2 Scalar Input Case ($m = 1$)

C.2.1 Linear Programming Trick

This section demonstrates that the RCBF optimization in (7.14) can be re-formulated as a linear program if the input is scalar. If u_0 satisfies the constraint in (7.14) then $u^* = u_0$ and there is no need to run the optimization. However, if u_0 is infeasible then:

$$p(x) + L_{\tilde{g}}h(x)\, u_0 - \theta|u_0||L_{\tilde{g}}h(x)| < 0 \tag{C.3}$$

In this case, substitute $u = u_p - u_n$ and $|u| = u_p + u_n$ to rewrite the optimization problem (7.14) as follows:

$$\begin{aligned} \begin{bmatrix} u_p^*(x) \\ u_n^*(x) \end{bmatrix} &= arg\min_{u_p,u_n} \frac{1}{2}\|(u_p - u_n) - u_0\|_2^2 \\ \text{s.t. } & p(x) + L_{\tilde{g}}h(x)(u_p - u_n) - \theta(u_p + u_n)|L_{\tilde{g}}h(x)| \geq 0 \\ & u_p \geq 0,\, u_n \geq 0 \end{aligned} \tag{C.4}$$

The reformulated problem has linear constraints. Hence this is a quadratic program with decision variables $u_p \geq 0$ and $u_n \geq 0$. This problem can be efficiently solved online instead of RCBF-SOCP.

Next, we show that solutions of optimization problem (C.4) are unique, i.e. u_p and u_n both can not simultaneously be greater than 0, one of the variables must be zero for another to be nonzero. To show this by contradiction, let's assume that $u_p^* > 0$ and $u_n^* > 0$. There are 3 cases to consider: Case-1: If $u_p^* - u_n^* = u_0$ then the constraint with

(u_p^*, u_n^*) is

$$\begin{aligned} 0 &\leq p(x) + L_{\tilde{g}}h(x)u_0 - \theta(u_p^* + u_n^*)|L_{\tilde{g}}h(x)| \\ &\leq p(x) + L_{\tilde{g}}h(x)u_0 - \theta|u_0||L_{\tilde{g}}h(x)| \end{aligned} \tag{C.5}$$

This follows from $|u_0| \leq \max(u_p^* - u_n^*, u_n^* - u_p^*) \leq u_p^* + u_n^*$, which implies that u_0 is feasible. However, this case cannot happen as we do not run the optimization if u_0 is feasible.

Case-2: Assume $u_p^* - u_n^* < u_0$. If $L_{\tilde{g}}h(x) > 0$ then define $\tilde{u}_n := u_n^* - \epsilon$ for $\epsilon > 0$. The assumption $u_n^* > 0$ implies that $\tilde{u}_n > 0$ for sufficiently small ϵ and the constraint remains satisfied with $(u_p^*, \tilde{u}_n)$ because,

$$\begin{aligned} &p(x) + L_{\tilde{g}}h(x)(u_p^* - \tilde{u}_n) - \theta(u_p^* + \tilde{u}_n)|L_{\tilde{g}}h(x)| \\ &= \left[p(x) + L_{\tilde{g}}h(x)(u_p^* - u_n^*) - \theta(u_p^* + u_n^*)|L_{\tilde{g}}h(x)|\right] \ldots \\ &\qquad + \epsilon\left(L_{\tilde{g}}h(x) + \theta|L_{\tilde{g}}h(x)|\right) \geq 0 \end{aligned} \tag{C.6}$$

The term in brackets is ≥ 0 by feasibility of (u_p^*, u_n^*) and the term with ϵ is ≥ 0 because $L_{\tilde{g}}h(x) > 0$. Thus, the pair $(u_p^*, \tilde{u}_n)$ is feasible and moreover it gives lower cost than (u_p^*, u_n^*), because we have slightly increased $u_p^* - \tilde{u}_n$ toward u_0. If $L_{\tilde{g}}h(x) < 0$ then set $\tilde{u}_p = u_p^* + \epsilon$ and follow a similar argument to show that $(\tilde{u}_p, u_n^*)$ is feasible and gives lower cost than (u_p^*, u_n^*). Thus, in either case ($L_{\tilde{g}}h(x) < 0$ or > 0) we have that if $u_p^* > 0$ and $u_n^* > 0$ then it can not be an optimal point. Hence, by contradiction at least one of them must be zero.

Case-3: Assume $u_p^* - u_n^* > u_0$. This case is similar to Case-2 and is not included.

C.2.2 Connection Between RCBF-SOCP and CBF-QP

For the scalar input case ($u(t) \in \mathbb{R}$), we can scale the optimal input u^* of the CBF-QP problem (i.e. $\theta = 0$) to obtain a more efficient implementation, instead of solving more complex RCBF-SOCP. To see this, rewrite the worst-case uncertain input $w^*(u)$ from Equation (7.11) as follows:

$$w^*(u) = -\theta\,|u|\,sgn(L_{\tilde{g}}h(x)) \tag{C.7}$$

Plug in $w^*(u)$ from Equation (C.7) to the RCBF constraint of the optimization problem (7.13) and use the fact $L_{\tilde{g}}h(x) = \frac{1}{2}(\alpha+\beta)L_g h(x)$ to yield:

$$L_f h(x) + \frac{1}{2}(\alpha+\beta)L_g h(x)\Big[u - \theta\,|u|\,sgn(L_{\tilde{g}}h(x))\Big] \geq -\eta(h(x)) \tag{C.8}$$

Define $\rho(x,u) := \frac{1}{2}(\alpha+\beta)\left[1 - \frac{\theta\,sgn(L_{\tilde{g}}h(x))}{sgn(u)}\right]$ and rewrite above constraint as:

$$L_f h(x) + \rho(x,u)L_g h(x)u \geq -\eta(h(x)) \tag{C.9}$$

The optimization problem (7.13) is thus of the form:

$$\begin{aligned} u^*(x) = \arg\min_{u\in\mathcal{U}} \frac{1}{2}\|u-u_0\|_2^2 \\ \text{s.t. } L_f h(x) + \rho(x,u)L_g h(x)u \geq -\eta(h(x)) \end{aligned} \tag{C.10}$$

In the above problem $\rho(x,u)$ depends on the sign of u. Let u^* be the optimal solution for the nominal case with $\theta = 0$, (i.e. α and β both tend to 1), then:

$$L_f h(x) + L_g h(x)\,u^* \geq -\eta(h(x)) \tag{C.11}$$

We know that $L_g h(x)$ does not change sign by $L_{\tilde{g}}h(x) \neq 0$, $\forall x \in D$ assumption. Assume $L_g h(x) > 0$ without loss of generality. There are three cases to consider:

1. $u^* = 0$: The solution $u = u^*$ is optimal for the RCBF, because $L_f h(x) \geq -\eta(h(x))$.
2. $u^* > 0$: The solution $u = u^*/\alpha$ is optimal for the RCBF, because $\rho(x,u^*) = \frac{1}{2}(\alpha+\beta)\left[1 - \frac{\beta-\alpha}{\beta+\alpha}\right] = \alpha$.
3. $u^* < 0$: The solution $u = u^*/\beta$ is optimal for the RCBF, because $\rho(x,u^*) = \frac{1}{2}(\alpha+\beta)\left[1 + \frac{\beta-\alpha}{\beta+\alpha}\right] = \beta$.

Finally, this scaling only works for the scalar input case and is not true in general for the multiple input case, because w^* in Equation (7.11) depends on the direction of $L_{\tilde{g}}h(x)$. This direction is not captured by simply scaling the input u^* of the original CBF-QP.

Appendix D

LTVTools

LTVTools [9] is a MATLAB toolbox for modeling, analysis and control design in a finite horizon Linear Time-Varying (LTV) framework. It provides a user friendly interface to interact with time-varying matrices (`tvmat`) and time-varying state-space (`tvss`) system objects. These are similar to the "gridded" objects used in LPVTools [152, 153]. The first (beta) version of LTVTools was developed as part of the work in [17] and [10]. During this research, several new functions were added, and the user interface was significantly improved. Next, a summary of available functions is provided, followed by a summary of specific contributions.

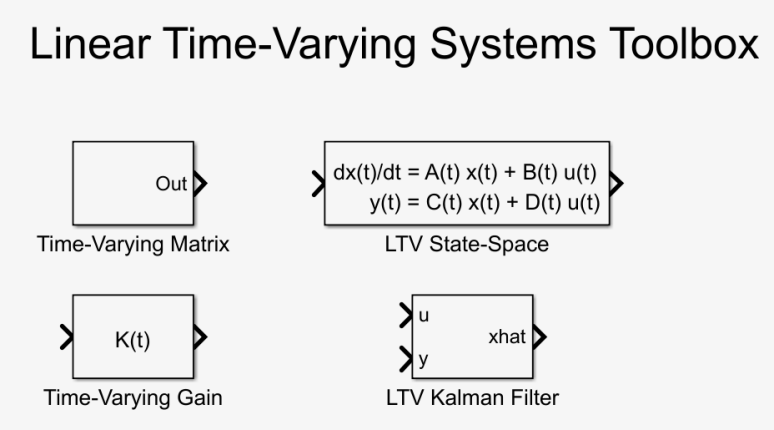

Figure D.1: Simulink Library `ltvlib`

MATLAB® and Simulink® are registered trademarks of The MathWorks, Inc.

Time-Varying Objects:

Object	Description
`tvmat`	Create a continuous-time or discrete-time time-varying matrix.
`tvss`	Create a continuous-time or discrete-time time-varying state-space system.

Time-Varying Object Manipulations:

Function	Description
`tvsplit`	Split the time-varying object at a specified time points.
`tvmerge`	Merge the time-varying object.
`evalt`	Evaluate time-varying object on a specified time-grid.
`tvsubs`	Get time-varying object data by substituting specific time/grid values.
`tvmax`	Find maximum entry from the time-varying matrix (`tvmat`) data.
`tvmin`	Find minimum entry from the time-varying matrix (`tvmat`) data.
`tvdiff`	Differentiate time-varying matrix with respect to time.
`tv2ts`	Convert time-varying matrix to a `timeseries` object.
`tv2fh`	Convert time-varying matrix to a function handle.

Plotting Utilities:

Function	Description
`tvplot`	Plot time-varying matrix data across time.
`tvsplot`	Plot individual entry of a time-varying matrix in a subplot.
`tvstem`	Plot time-varying discrete-time data across sampling grid.
`tvsemilogy`	Plot time-varying signal on a `semilogy` plot.
`tvsemilogx`	Plot time-varying signal on a `semilogy` plot.
`tvloglog`	Plot time-varying signal on a `loglog` plot.

Special Differential Equations*:

Function	Description
`cdre`	Solve Continuous-time Differential Riccati Equation (CDRE).
`cdle`	Solve Continuous-time Differential Lyapunov Equation (CDLE).

*The discrete-time counter part `ddre` and `ddle` functions for difference equation can be implemented.

LTV Simulations:

Function	Description
`tvlsim`	Simulate LTV system using specified initial condition and input.
`tvimpulse`	Simulate impulse response of the LTV system.
`tvinitial`	Simulate initial condition response of the LTV system.
`tvstep`	Simulate step response of the LTV system.

Analysis Functions:

Function	Description
`tvnorm`	Finite horizon $\mathcal{L}_2[0,T]$ norm of a signal.
`tvnormp`	Induced gain of the `tvss` system using power iteration (Algorithm 1).
`tvnormb`	Induced gain of the `tvss` system using RDE bisection (Algorithm 2).
`tvnorm`	Induced norm of a system using combined algorithm (Algorithm 3).
`tvnormg`	Routine to check if specified γ_{try} is an upper bound on induced gain.
`tvh2norm`	Finite horizon H_2 norm of the LTV system (Section 3.4).
`tvgram`	Compute finite horizon Controllability and Observability Gramians.
`tvcovar`	Output/state covariance of the LTV system driven by white noise.
`tvsens`	First-order LTV sensitivity analysis (Section 4.6).
`powerit`	Power iteration lower bound algorithm for nonlinear systems.
`tvwcgain`	Worst-case gain IQC upper bound algorithm [10] (Section 4.4).

Customization Options:

Object	Description
`tvodeOptions`	Specify options for ODE integration.
`tvnormOptions`	Specify options for `tvnorm` function.
`tvlmiOptions`	Specify options for the DLMI step used in `tvwcgain`.
`tvwcOptions`	Specify options for `tvwcgain` function.
`tvstepOptions`	Specify options for `tvstep` function.
`tvhinfsynOptions`	Specify options for finite horizon H_∞ synthesis.
`tvrobsynOptions`	Specify options for finite horizon robust synthesis.
`poweritOptions`	Specify options for power iterations.

Synthesis Functions:

Function	**Description**
`tvlqr`	Finite horizon Linear Quadratic Regulator (LQR) design.
`tvkalman`	Finite horizon Kalman filter design.
`tvlqg`	Finite horizon Linear Quadratic Gaussian (LQG) design.
`tvh2syn`	Finite horizon H_2 synthesis [23].
`tvhinffi`	Finite horizon H_∞ full-information control design [23].
`tvhinfsfb`	Finite horizon H_∞ state-feedback synthesis.
`tvhinfsyn`	Finite horizon H_∞ output-feedback synthesis (Section 5.3).
`tvsyn`	A common interface for H_∞ state or output feedback synthesis.
`tvrobsyn`	Finite horizon IQC-robust synthesis (Algorithm 5).
`tvfact`	Compute finite horizon factorization (Section B.1).

The work in [17] provided foundational objects `tvmat` and `tvss` with functions `tvnorm`, `evalt`, `tvlsim`, `tvdiff`, `tvlqr` and `cdre`. The work in [10] provided an initial implementation for the function `tvwcgain` with related examples. The rest of the features are developed as part of this dissertation. These include updated user interface and discrete-time support (for `tvmat`, `tvss`, and `tvlsim`), new finite horizon H_2 analysis tools (e.g. `cdle`, `tvcovar`, `tvh2norm`, `tvgram`), plotting utilities, synthesis functions etc. Several of the new functions link back to algorithms developed in this dissertation (e.g. `tvnorm`, `tvrobsyn`). New customization options are provided for a user to customize specific function behavior. Moreover, a new `ltvlib` Simulink library (Figure D.1) is added to easily manipulate with `tvmat` and `tvss` objects in a Simulink environment. Finally, some new demos are added including examples from this dissertation to demonstrate use of these tools.