# ANALYSIS AND DESIGN OF ROBUST CONTROL FOR LINEAR PARAMETER-VARYING SYSTEMS

A Dissertation

by

SHAO-CHEN HSU

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

| | |
|---|---|
| Chair of Committee, | Raktim Bhattacharya |
| Committee Members, | Robert E. Skelton |
| | Moble Benedict |
| | Pilwon Hur |
| Head of Department, | Rodney W. Bowersox |

August 2019

Major Subject: Aerospace Engineering

# ABSTRACT

Gain-scheduling approach is a powerful tool but it only guarantees the local stability and performance for a slow varying system. Linear parameter varying (LPV) systems hence were developed to overcome this drawback. The LPV system is a linear system with parameter-dependent system matrices, which can be formulated from a nonlinear system via either approximation or function substitution. Three major control design methods includes linear fractional transformation, polytopic system design and gridding approach. All methods results in a convex optimization with either parameter-dependent or parameter-independent linear matrix inequalities (LMIs) and some conservatism may be introduced. Gridding based approach is the main focus in this research because it has no further assumptions about the structure and hence admits less conservatism. However, the number of samples for gridding approach grows up exponentially as the dimension of the problem increases. This drawback hence inspires the approach developed in this research.

Several stability and performance conditions are introduced in this research and all controller syntheses arrive at optimization problems with parameter-dependent LMIs. Hence the objective of this research is to solve these problems. We present two methodologies to handle with generic LPV control systems. The first approach is to consider the problem in a stochastic framework so that the stability and performance are guaranteed in the stochastic sense. Two algorithms, i.e. polynomial chaos expansion and stochastic collocation, are used to formulate the convex optimization problems. The other method is to directly interpolate the parameter-dependent LMIs by sparse grid with Smolyak algorithm, which extremely reduces the amount of the sample points and successfully solve the infinite-dimensional optimization by the proposed algorithm. Two examples are shown to compare two proposed controllers with existing methods, where the benefits of the method we develop are shown and some limitations of the current methodologies are discussed.

# DEDICATION

To my father and mother, who support me with endless love and encouragement.

# ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Raktim Bhattacharya for the patient guidance, encouragement, immense knowledge and advice he has provided during my Ph.D career. His extensive imagination inspires my research extended to an advanced aspect. Besides, I would like to thank the rest of my committee for their hard questions and insightful comments which makes my thesis more solid.

I would also like to thank all the members in the ISRL group, especially Niladri, Vaishnav, Vedang and Sunsoo for any kind of help to me. Your kindness and friendliness make me feel like home in the lab.

I gratefully acknowledge the funding received towards my first year PhD from TIAS Heep Fellowship which helps me with the cost of my living expense.

Last but not the least, I would like to especially thank my beloved wife Yi-Chun, who completely backs me up both mentally and physically. Without her love, I would have never become who I am today.

# CONTRIBUTORS AND FUNDING SOURCES

## Contributors

This work was advised by a dissertation committee consisting of Professors Raktim Bhattacharya, Robert E. Skelton and Moble Benedict of the Department of Aerospace Engineering and Professor Pilwon Hur of the Department of Mechanical Engineering.

The simulation code for Tensegrity dynamical systems in Chapter 5 was developed in cooperation with Vaishnav Tadiparthi and Professor Raktim Bhattacharya.

All other work conducted for the dissertation was completed by the student independently.

## Funding Sources

Graduate study was supported by TIAS Heep Fellowship.

# NOMENCLATURE

| | |
|---|---|
| LPV | Linear Parameter Varying |
| LTV | Linear Time Varying |
| LTI | Linear Time Invariant |
| LFT | Linear Fractional Transformation |
| LMI | Linear Matrix Inequality |
| PC | Polynomial Chaos |
| SC | Stochastic Collocation |
| SG | Sparse Grid |
| $\mathsf{E}\left[\bullet\right]$ | Expected Value of $\bullet$ |
| $\mathsf{sym}\left(\boldsymbol{A}\right)$ | Symmetric Structure, i.e. $\boldsymbol{A}+\boldsymbol{A}^T$ |

## TABLE OF CONTENTS

Page

# LIST OF FIGURES

FIGURE Page

# LIST OF TABLES

TABLE Page

# 1. INTRODUCTION

## 1.1 Origin of Linear Parameter-Varying Systems

Linear control is a well studied subject that is under a big assumption that the dynamical system is a linear mapping from inputs to outputs. However, real physical systems are never performing as linear systems even a simple mass-spring-damper system which can be nonlinear with respect to its state-dependent spring and damper coefficients. A linear approximation technique, e.g. Jacobian linearization, thus is applied to the systems at an operating point (or so-called equilibrium point) such that the nonlinearities are eliminated, but it is limited by a "small" deviation from the point and the error becomes significant when a trajectory is deviated too much.

As a result of the drawback of linear control, A gain-scheduling method, which is adaptive to the original nonlinear system, has been developed. With a big assumption of the slowly varying scheduling variables, the outline of gain-scheduling design is described as following steps:

1. Choose different operating points from a trajectory.
2. Linearize the system about each chosen point to formulate a family of linear systems.
3. Implement linear control theory for each linearized system to design a set of controllers and satisfy specific specifications.
4. Use interpolation between each controller design to obtain a smooth structure.

Because of the simplicity of the design process, it has been widely used in various applications, e.g. flight systems, ground vehicles, chemical process [4, 5, 6], etc., yet the performance and stability of the controlled system are only guaranteed at each design point instead of thorough trajectories due to the absence of theoretical analysis. The current major resolution is to exam the system by simulation explicitly. Furthermore, Shamma [7] has discussed the stability issue of the parameter dependent system when designing controllers with ignorance of the parameter time variations. An example of frozen parameter design has been demonstrated that it may still cause unstable system at designed point if relatively fast time varying parameters exist. Consequently, the research on linear parameter-varying systems are inspired, that the whole linearized models should be investigated synchronously when designing controllers.

Linear parameter varying (LPV) systems are a special class of nonlinear systems, which were introduced in [8] and can be described by the ordinary differential equations of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{\rho}(t))\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\rho}(t))\boldsymbol{u}, \tag{1.1}$$

where $\boldsymbol{\rho}(t)$ is a time varying exogenous parameter vector and measurable in real time. To simplify the expression, $\boldsymbol{\rho}(t)$ and $\boldsymbol{\rho}$ are changeable and both will exist later on, but they fundamentally represent the same meaning. The system matrices, $\boldsymbol{A}(\boldsymbol{\rho}(t))$ and $\boldsymbol{B}(\boldsymbol{\rho}(t))$ are both depending on the parameters, which can be either polytopic or fully nonlinear. Further, the model with the parameters containing the states $\boldsymbol{x}$ is called quasi-LPV system. LPV modeling provides a mechanism for transforming many nonlinear systems and designing controllers using parameter dependent convex optimization problems.

## 1.2 Literature Review

This section surveys some important research about LPV systems, which includes modeling and controller design. One should note that the modeling indicated in this section means the ways to transform an original nonlinear system with or without exogenous parameters to a standard LPV or quasi-LPV formulation defined in (1.1). Generally speaking, Two catogories of modeling techniques have been investigated with different philosophies to transform nonlinear systems to LPV form either approximately or exactly [9]. In addition, three kinds of control syntheses have been developed and all of them turn out to become solutions to convex optimization problems. A brief overview of LPV modeling and control are shown as follows.

The goal of modeling is to transform an original nonlinear system to a standard LPV or quasi-LPV formulation by several techniques. The first direction is to utilize Jacobian linearization to the original nonlinear system around chosen points and formulate an LPV system with interpolation of a family of linearized models, which is similar to conventional gain-scheduling technique. The points can be chosen from equilibrium conditions, nominal trajectories, or operating points deviated from the equilibrium. Linearization based method is the most straightforward method and easy to apply to various types of nonlinear systems. However, the result is a local approximation around the trim points , so the linearized model may get a divergent behavior for large inputs and to capture the transient response of the original system seems to be impossible [10].

The second category is substitution based transformations, which somehow substitute nonlinear terms into linear ones. The easiest way is to simply rewrite a nonlinear function as a virtual scheduling parameter, but the result of the model is not unique because of the different choices of scheduling parameters assignment from the user. Some application can be seen in [11, 12]. One drawback is that the singularities may not been seen in the scheduling parameters if the substitutions are not used carefully. The detail is shown in [9]. The second way of substitution based method is state transformation first introduced in [13], which eliminate the nonlinear terms independent of scheduling parameters, but it is applied only in a special form of nonlinear system as

$$\begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{\boldsymbol{w}} \end{bmatrix} = \boldsymbol{A}(\boldsymbol{\rho}(t)) \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{w} \end{bmatrix} + \boldsymbol{B}(\boldsymbol{\rho}(t))\boldsymbol{u}(t) + \boldsymbol{K}(\boldsymbol{\rho}(t)) \tag{1.2}$$

with

$$\boldsymbol{A}(\boldsymbol{\rho}(t)) := \begin{bmatrix} \boldsymbol{A}_{11}(\boldsymbol{\rho}(t)) & \boldsymbol{A}12(\boldsymbol{\rho}(t)) \\ \boldsymbol{A}_{21}(\boldsymbol{\rho}(t)) & \boldsymbol{A}22(\boldsymbol{\rho}(t)) \end{bmatrix}, \boldsymbol{B}(\boldsymbol{\rho}(t)) := \begin{bmatrix} \boldsymbol{B}_1(\boldsymbol{\rho}(t)) \\ \boldsymbol{B}_2(\boldsymbol{\rho}(t)) \end{bmatrix}, \quad \text{and } \boldsymbol{K}(\boldsymbol{\rho}) := \begin{bmatrix} \boldsymbol{K}_1(\boldsymbol{\rho}) \\ \boldsymbol{K}_2(\boldsymbol{\rho}) \end{bmatrix}$$

Under an assumption that there exist continuously differentiable functions $\boldsymbol{w}_{eq}$ and $\boldsymbol{u}_{eq}$ such that an equilibrium condition

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_1(\boldsymbol{\rho}(t)) \\ \boldsymbol{K}_2(\boldsymbol{\rho}(t)) \end{bmatrix} + \begin{bmatrix} \boldsymbol{A}_{11}(\boldsymbol{\rho}(t)) & \boldsymbol{A}_{12}(\boldsymbol{\rho}(t)) \\ \boldsymbol{A}_{21}(\boldsymbol{\rho}(t)) & \boldsymbol{A}_{22}(\boldsymbol{\rho}(t)) \end{bmatrix} \begin{bmatrix} \boldsymbol{z}(\boldsymbol{\rho}(t)) \\ \boldsymbol{w}_{eq}(\boldsymbol{\rho}(t)) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1(\boldsymbol{\rho}(t)) \\ \boldsymbol{B}_2(\boldsymbol{\rho}(t)) \end{bmatrix} \boldsymbol{u}_{eq}(\boldsymbol{\rho}(t))$$

is satisfied, the nonlinear system (1.2) can be transformed to

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{\boldsymbol{w}} - \dot{\boldsymbol{w}}_{eq} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{0}) & \boldsymbol{A}_{12}(\boldsymbol{\rho}(t)) \\ \boldsymbol{0} & \boldsymbol{A}_{22}(\boldsymbol{\rho}(t)) - \frac{\partial \boldsymbol{w}_{eq}(\boldsymbol{\rho}(t))}{\partial \boldsymbol{z}(t)} \boldsymbol{A}_{12}(\boldsymbol{\rho}(t)) \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{w} - \boldsymbol{w}_{eq} \end{bmatrix} \\ &+ \begin{bmatrix} \boldsymbol{B}_1(\boldsymbol{\rho}(t)) \\ \boldsymbol{B}_2(\boldsymbol{\rho}(t)) - \frac{\partial \boldsymbol{w}_{eq}(\boldsymbol{\rho}(t))}{\partial \boldsymbol{z}(t)} \boldsymbol{B}_1(\boldsymbol{\rho}(t)) \end{bmatrix} (\boldsymbol{u} - \boldsymbol{u}_{eq}) \end{aligned}$$

by basic algebra manipulations. One should notice that this approach does yield an exact transformation without any approximation, which is different from Jacobian linearization, but it is limited to some conditions described above. More detail and related application can be found in [10, 14, 15, 16]. The third technique named function substitution was proposed in [17]. This approach approximates a quasi-LPV model with nonlinear terms of control inputs and non-scheduling states to a standard quasi-LPV form. The idea is to introduce function decomposition that is linear in scheduling parameter to obtain a linearization of the nonlinear input. By solving minimization problems subject to some smoothness constraints at grid points that are chosen by the user, one can obtain a final decomposition function and the final quasi-LPV form is carried out by substituting this approximation back to the original quasi-LPV system. It is noted that the method is also an approximation as the Jacobian linearization and the accuracy is highly depending on the choice of grid points. Recently Shin has resolved the local statility issue caused by misrepresentation of the model [18], but it is still an open question for choosing the points to get the optimal model. Nevertheless, function substitution seems to be more accurate for some models, such as the model of F-16 [19, 20, 21], Boeing 747-100/200 [17, 10] and missile systems [22].

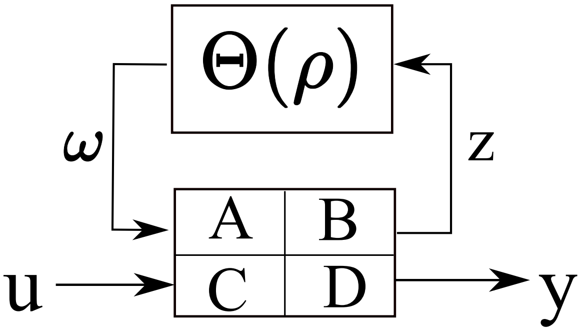


Figure 1.1: A block diagram describes the structure of LFT model where upper block represents the structure of parameters and lower block denotes the nominal linear system with parameter inputs.

Different from the gain-scheduling method, the LPV control theoretically proves the guarantee of performance and stability of the system. This can be achieved through several approaches. One way is to use linear fractional transformation (LFT) shown in Fig.(1.1), which is also omnipresent in the area of systems with uncertainties, e.g. $\mu$-analysis [23]. This allows an LPV system to be transformed into a two-block interconnection, where one represents the nominal system and one is associated with parameters describing how they enter the system. In addition, scaled small-gain theorem is a key technique to be used for bounded conditions [24, 25, 26]. However, a certain conservatism is existing due to a hypothesis on the multipliers with a block-diagonal structure. Scherer [26, 27, 28, 29] thus has introduced so-called full block S-procedure, which allows for arbitrary fast parameter variations by subdividing the parameter set, for LPV systems in the form of LFT so that the conservatism is reduced to an arbitrary degree.

The other approach of LPV control is based on Lyapunov functions, which can be either parameter-dependent or parameter-independent. Since the unknown matrix of the latter is considered as a constant and the associated stability condition formulated from a fixed value matrix needs to be satisfied for each element in the set of parameters, it would be more conservative than the former one that is adaptive to the parameter variations. Furthermore, the LPV control with a parameter-dependent Lyapunov function turns out to be a parameter-dependent convex optimization problem, which is infinite dimensional, so the research on specific structure of parameter-dependent

Lyapunov function is indispensable. The most straightforward and common way is to design a controller based on the LPV system matrices affine in parameters, or called a polytopic system. A parameter-affine Lyapunov function thus can be defined and evntually yield a convex optimization problem in finite dimensional [30, 31, 32, 33, 34]. An obvious drawback of polytopic control system is conservatism due to an assumption of linearity of the parameters in the structure of Lyapunov function, although a polytopic system may require less effort on numerical computations.

To reduce the conservatism from the affine parameter-dependent Lyapunov function, a generic LPV system is considered, where no further assumption about the structure of the system matrices is needed. Hence, an infinite-dimensional parameter-dependent Lyapunov function is parameterized by user-defined functions, which are highly depending on the nonlinear terms in the LPV system. Moreover, the resulting control synthesis arrives at parameter-dependent linear matrix inequalities (LMIs) [35, 36, 37, 38], which is convex if the parameters are constants. Intuitively, a gridding technique is used to find a solution to the parameter-dependent optimization problem, where a set of sample points are chosen and substituted into the LMIs respectively. Accordingly, the controller is obtained by solving the problem with a large amount of LMIs simultaneously. One should notice that the controller obtained here does not necessarily guarantee the performance and stability for the whole parameter set but the samples chosen from the set. A further check-up should be done by analyzing the stability with a set of finer grids or implementing the control system in simulation directly. Another open question is the choice of the grid points, which involves the amount of points and the location of the points. Moreover, the size of the problem exponentially increase as the dimension of the parameters increase, which is known as "the curse of dimensionality" [39].

All these approaches result in convex optimization problems with finite but large number of variables and there is a tradeoff between computational complexity and conservatism in the design. While the polytopic framework renders somehow less computation cost, the generic one carries out a less conservative result. However, regarding an extreme case in which a highly nonlinear system is given, the polytopic system may decrease computational efficiency more than the generic one since more virtual scheduling parameters should be introduced to achieve the polytopic framework. Hoffmann [40] has shown a decision chart which gives an advice how to choose a method for an LPV system.

LPV framework is quite elegant and powerful and many engineering problems exploit this to design controllers that guarantee performance within certain specifications. For example, Lu et. al. [41] assigned rotational speed of motors in active magnetic bearing systems as a scheduling variable and designed a robust $H_\infty$ control that improves the performance. Johansen et. al. [42] designed a switched vehicle-speed-dependent LQR control for an anti-lock brake system to maintain steerability. Fialho et. al. [43] combined LPV control with backstepping techniques to achieve adaptive active suspension by scheduling both suspension deflections and vehicle speed. LPV control systems has also been used widely in the area of aerospace engineering. Tan et. al. [22] reduced a nonlinear coupled generic missile model to a quasi-LPV form. Marcos and Balas [17] have also investigated different approaches to obtain quasi-LPV models for Boeing 747 series 100/200 and showed that these models match at trim points. Various control techniques have been developed based on F-16 aircraft and missile autopilot LPV model, e.g. $L_2$ norm control [3, 44] and model predictive control [21], by scheduling angle of attack and vehicle velocity.

## 1.3 Motivation

### 1.3.1 Modeling of Multibody Dynamics in Non-minimal Coordinates

LPV systems have been proved that it is a successful framework to investigate many engineering applications, especially in the area of aerospace, robotics, automotive and etc. When looking deeply into the fundamental development of dynamics, a minimal set of generalized coordinates is usually used to describe the multibody system involving lots of nonlinear term, e.g. trigonometric functions, so the configuration space is naturally reduced in the presence of joints and other constraints. Changing the generalized coordinates with Cartesian coordinates, which is a non-minimal representation, to describe the motion of bodies provides two notable advantages. First, A non-minimal coordinate system avoids singularity issues that cause the dynamical model fail to be well-behaved. As an example of tesegrity system, Skelton observed [45] that in three dimensions, a minimal coordinates approach is prone to singularities developed in the mass matrix, and therefore, the dynamics necessitates an excess coordinates description. Additionally, non-minimal descriptions of vector kinematics allows us to write elegant differential-algebraic equations (DAE), free of trigonometric terms.

### 1.3.2 LPV Control with Gridding Method

Conservatism is always an important issue that needs to be concerned during LPV control design process, since some assumptions are necessarily made to simplify the problem and render a convex optimization problem with a family of LMIs. An LFT based control, for example, assumes a system having rational dependence on a set of scheduling parameters, and an LPV control based on polytopic system also assumes system matrices affine in the parameters. As a result, these methods have no capability of resolving generic LPV problems directly unless introducing a larger set of parameters or linearizing the problems, which introduces conservatism further. The gridding method, however, is able to handle a more general system even with highly nonlinear terms, although desired conditions are only guaranteed at grid points. The other advantage is more degree of freedom of parameterization for infinite-dimensional function in convex optimization that renders a controller with better resolution. Therefore, it has great potentiality to explore a feasible solution to a complex system with less conservatism.

As mentioned in the previous section, the choices of location and number of grid points are still open questions. Generally speaking, equidistant samples in each dimension are chosen since beforehand there is no clue about variations of parameters in time. Besides, a finer grid yields guarantees of feasibility at more sample points, but adversely the computational expense increases exponentially. Hence, a novel gridding method that can guarantee the stability and performance for all elements in the parameter set with both less computation effort and conservatism becomes a priority. For instance, Bandeira et. al. [46] and Araujo et. al. [47] recently have proposed a gridding-based algorithm for stability analysis of LPV systems by using wavelet theory, in which the system matrix are approximated as a Haar series expansion and an upper bound of the associated approximation error is developed. Using the error bound one can ensures the stability for the whole parameter set. Although the conservatism may be introduced, this can be alleviated by increasing the order of the Haar expansion.

### 1.4 Contribution of Thesis

This section states major contributions of the thesis separated in two parts: an LPV model of a special class of rigidbody motions, i.e. tensegrity structures, and control theories in the LPV framework.

Regarding LPV model, this thesis is tempted to obtain a mathematical description of rigid multibody dynamics in the LPV framework. First of all, a Lagrangian formulation based on Cartesian coordinates is used for deriving DAEs of the governing equations of motion (EOM) for a rigid multibody system. In addition, a quasi-LPV system is transformed from the EOM by introducing virtual scheduling parameters, which are system states themself, after model reduction by coordinate partitioning [48] is implemented. Finally, a novel technique for improving the accuracy of the simulation is developed to ensure that the errors in states arising from numerical integration are corrected on the position and velocity levels according to both geometric and energy constraints.

With respect to LPV control, We are focusing on both linear quadratic regulator (LQR), which is an optimal control with respect to a quadratic cost function, and $H_2$ control, which yields a robust performance under certain disturbances. Both control methods in the LPV framework result in a parameter-dependent convex optimization problem. This thesis develops two approaches to solve this problem. Firstly, a new LPV regulator synthesis algorithm, in the stochastic framework, which generates a parameter dependent gain, is developed. In this framework, orthogonal basis functions are determined by polynomial chaos theory. The controller is optimal with respect to a quadratic cost in states and control. To see the performance we look at the LPV-LQR problem with application to a realistic autopilot missile control problem. Secondly, an interpolation technique, i.e. Smolyak algorithm, is utilized directly to the parameter-dependent optimization problem so that the size of optimization is extremely reduced comparing with the convectional tensor-grid method. An LQR and $H_2$ control are developed with this methodology and applied to a robotic arm based on tensegrity structures as an example.

## 1.5 Organization of Thesis

This section gives a brief overview of each chapter. In Chapter 2, we analyze stability conditions of generic LPV systems, which contain quadratic stability and robust stability. Moreover, a necessary condition to guarantee a linear quadratic regulator for LPV systems is developed, where the control gain is parameter-dependent and solved via an convex optimization problem. A full-state feedback $H_2$ control synthesis for LPV systems is also being established, where a convex optimization formulated with two kinds of parameter-dependent LMIs guarantees both the stability and performance.

Chapter 3 presents a new theoretical framework for designing linear parameter varying controllers in the stochastic framework. We assume the scheduling variables to be random and implement the polynomial chaos expansion to synthesize the controller for the resulting linear stochastic dynamical system. Two algorithms are presented that minimize the performance objective with respect to the stochastic system. The first algorithm is based on the polynomial chaos expansion and the second algorithm is based on the stochastic collocation.

Instead of considering a stochastic framework as Chapter 3, Chapter 4 introduces another novel approach that improves the conventional gridding method. A high dimensional polynomial interpolation with sparse grid, i.e. Smolyak algorithm, is used to approximate parameter-dependent LMIs derived in Chapter 3. Using the algorithm, a convex optimization problem with finite-dimensional LMIs is obtain. The control gain then can be determined by solving the problem.

In Chapter 5, we present a Lagrangian approach to simulating multibody dynamics in a tensegrity framework with an ability to tackle holonomic constraint violations in an energy-preserving scheme. Governing equations are described using non-minimum coordinates to simplify descriptions of the general multibody structure's kinematics. This leads to an undesirable constraint drift with numerical computation errors. To minimize this constraint drift, the direct correction method has been employed in conjunction with a novel energy-correcting scheme that treats the total mechanical energy of the system as a constraint to nullify any numerical violations occurring in integration. The coordinate partitioning method is applied transform the nonminimum description to a minimum coordinate system, then the nonlinear terms in system matrices are considered as matrices function of parameter and we finally reach a standard quasi-LPV formulation.

Chapter 6 shows two examples to demonstrate the performance of the proposed methods. The first problem is a common LPV application, i.e. a nonlinear autopilot missile system. The control objective of the problem is to stabilize the trajectory with LQR. Both proposed methods are applied to the system and we compare the results with each other. The second problem is a robotic arm built with tensegrity structures in 2 dimensions, where the derived mathematical model and the associate simulation algorithms are from the Chapter 5. We opt to have a control objective for this model: a tracking problem. Since the LPV control with stochastic framework is not suitable for high dimension problem, we only test Smolyak algorithm based control in this example.

Finally, Chapter 7 concludes the work done in this thesis and discuss related future work that may improve the performance and computation time for controller synthesis.

# 2. ANALYSIS AND CONTROLLER DESIGN OF LPV SYSTEMS

In this chapter we state stability results for generic LPV systems. This involves quadratic stability condition and robust stability condition. In addition, we develop a linear quadratic regular in an LPV framework, in which a sufficient condition, that guarantees both stability and performance criteria, is derived. Finally, we introduced an LPV system with disturbances and derive certain conditions of $H_2$ control that ensure disturbance rejection and stability of the systems.

## 2.1 Mathematical Preliminary

Before presenting the main content of this chapter, some definitions and notations have to be made.

### 2.1.1 Stability of Dynamical Systems

Consider a dynamical system of the form

$$\dot{\boldsymbol{x}}(t, \boldsymbol{\rho}) = f(\boldsymbol{x}(t)) \tag{2.1}$$

with an initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$, where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the system state vector, $f(\boldsymbol{x}(t)) : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous function vector. We assume there exists an equilibrium point of the system (2.1), $\boldsymbol{x}_e$, such that $f(\boldsymbol{x}_e) = 0$, then we define the following stability conditions.

1. If, for each $\epsilon > 0$, there exists $\delta > 0$ such that $\|\boldsymbol{x}(t) - \boldsymbol{x}_e\| < \epsilon$ for all $t \geq 0$ if $\|\boldsymbol{x}_0 - \boldsymbol{x}_e\| < \delta$, then $\boldsymbol{x}_e$ is said to be *Lyapunov stable*.

2. If there exists $\delta > 0$ such that $\lim_{t\to\infty} \|\boldsymbol{x}(t) - \boldsymbol{x}_e\| = 0$ if $\|\boldsymbol{x}_0 - \boldsymbol{x}_e\| < \delta$, then $\boldsymbol{x}_e$ is said to be *asymptotically stable*.

3. If there exists $\delta > 0$, $m > 0$, $\lambda > 0$ such that $\|\boldsymbol{x}(t) - \boldsymbol{x}_e\| \leq m\|\boldsymbol{x}_0 - \boldsymbol{x}_e\|e^{-\lambda t}$ for all $t \geq 0$

if $\|\boldsymbol{x}_0 - \boldsymbol{x}_e\| < \delta$, then $\boldsymbol{x}_e$ is said to be *exponentially stable*.

Based on these conditions we arrive a well-known theorem related to Lyapunov functions [49].

**Theorem 1.** *Consider the dynamical system* (2.1) *and an equilibrium point* $\boldsymbol{x}_e = 0$*. If there exists a continuous differentiable function* $V(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ *such that*

$$V(0) = 0 \, \textit{for } \boldsymbol{x} = \boldsymbol{0} \tag{2.2}$$

$$V(\boldsymbol{x}) > 0 \, \textit{for all } \boldsymbol{x} \neq \boldsymbol{0} \tag{2.3}$$

$$\dot{V}(\boldsymbol{x}) < 0 \, \textit{for all } \boldsymbol{x} \neq \boldsymbol{0}, \tag{2.4}$$

*then the equilibrium point* $\boldsymbol{x}_e$ *is globally asymptotically stable.*

The function $V$ in the theorem is called Lyapunov function, named after Aleksandr Mikhailovich Lyapunov [50]. Furthermore, a theorem with stronger stability condition is also derive in [49].

**Theorem 2.** *Consider the dynamical system* (2.1) *and an equilibrium point* $\boldsymbol{x}_e = 0$*. If there exists a continuous differentiable function* $V(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ *such that*

$$k_1\|\boldsymbol{x}\|^a \leq V(\boldsymbol{x}) \leq k_2\|\boldsymbol{x}\|^a$$

$$\dot{V}(\boldsymbol{x}) \leq -k_3\|\boldsymbol{x}\|^a \, \textit{for all } \boldsymbol{x} \neq \boldsymbol{0},$$

*where* $k_1, k_2, k_3$ *and* $a$ *are positive constants, then the equilibrium point* $\boldsymbol{x}_e$ *is globally exponentially stable.*

So far, we have results to satisfy two kinds of stability conditions for general nonlinear system. We further define stability conditions for general LPV systems. Consider a general LPV system of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(\boldsymbol{\rho}(t))\boldsymbol{x}(t) \tag{2.5}$$

with an initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$, where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector of the system, $\boldsymbol{A}(\boldsymbol{\rho}(t)) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n \times n}$ is the system matrix that is a continuous differentiable matrix function of the parameters $\boldsymbol{\rho}(t)$. We assume the parameter $\boldsymbol{\rho}(t)$ is bounded in magnitude and rate, i.e. $\boldsymbol{\rho}(t) \in \mathcal{D}_{\boldsymbol{\rho}} \subset \mathbb{R}^{n_\rho}$ and $\dot{\boldsymbol{\rho}}(t) \in \mathcal{D}_{\dot{\boldsymbol{\rho}}} \subset \mathbb{R}^{n_\rho}$. It is noted that we do not assume any structure in (2.5), that is, the LPV model here fits most cases of parameter-varying nonlinear systems even it is a quasi-LPV system. We then define the following stability conditions, which is also shown in [39].

1. If, for each $\epsilon > 0$, there exists $\delta > 0$ such that $\|\boldsymbol{x}(t, \boldsymbol{\rho}) - \boldsymbol{x}_e\| < \epsilon$ for all $t \geq 0$ and $\boldsymbol{\rho}(t) \in \mathcal{D}_{\boldsymbol{\rho}}$ if $\|\boldsymbol{x}_0 - \boldsymbol{x}_e\| < \delta$, then $\boldsymbol{x}_e$ is said to be *Lyapunov stable*.
2. If there exists $\delta > 0$ such that $\lim_{t\to\infty} \|\boldsymbol{x}(t, \boldsymbol{\rho}) - \boldsymbol{x}_e\| = 0$ for all $\boldsymbol{\rho}(t) \in \mathcal{D}_{\boldsymbol{\rho}}$ if $\|\boldsymbol{x}_0 - \boldsymbol{x}_e\| < \delta$, then $\boldsymbol{x}_e$ is said to be *asymptotically stable*.
3. If there exists $\delta > 0$, $m > 0$, $\lambda > 0$ such that $\|\boldsymbol{x}(t, \boldsymbol{\rho}) - \boldsymbol{x}_e\| \leq m\|\boldsymbol{x}_0 - \boldsymbol{x}_e\| e^{-\lambda t}$ for all $t \geq 0$ and $\boldsymbol{\rho}(t) \in \mathcal{D}_{\boldsymbol{\rho}}$ if $\|\boldsymbol{x}_0 - \boldsymbol{x}_e\| < \delta$, then $\boldsymbol{x}_e$ is said to be *asymptotically stable*.

### 2.1.2 Norms for Signals

To specify the performance of a system we need to compare the "size" of matrices and vectors somehow. The "size" mentioned here does not actually mean dimension of them but in the sense of certain properties we are interested in. Therefore, some spaces and norms needs to be defined so that we have a level to compare with.

First of all, we give a definition of a norm for signals.

**Definition 3.** *Suppose there exists a signal $u(t)$ defined on $\mathbb{R}$, then $\|u(t)\|$ is said to be a norm if it has the following 4 properties:*

- $\|u(t)\| \geq 0$
- $\|u(t)\| = 0$ *if and only if* $u(t) = 0$ *for all* $t$

- *For any* $a \in \mathbb{R}$*,* $\|au(t)\| = \|a\|\|u(t)\|$

- *For any* $v(t) \in \mathbb{R}$*,* $\|u(t) + v(t)\| \leq \|u(t)\| + \|v\|$

In particular, we can define an $\mathcal{L}_p$ norm of $u(t)$ as

$$\|u(t)\|_p := (\int_{-\infty}^{\infty} |u(t)|^p dt)^{\frac{1}{p}} \quad \text{for } p \geq 1$$

A special case of the norm is $\|u\|_\infty := \sup_t |u(t)|$, which is the least upper bound of the absolute value. Another interesting property is the average power of a signal, which is defined by

$$\mathbf{pow}(u(t)) := \left( \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} u(t)^2 dt \right)^{\frac{1}{2}}.$$

One should notice that the average power is not a norm since the power can be zero even if the $u(t)$ is not a zero signal.

The norm can also be defined in the frequency domain. For $\boldsymbol{u}(j\omega) \in \mathbb{C}$ as an example, The $\mathcal{L}_2$ norm can be defined as

$$\|\hat{u}(j\omega)\|_2 := \left( \frac{1}{2\pi} \int_{-\infty}^{\infty} |\hat{u}(j\omega))|^2 d\omega \right)^{\frac{1}{2}}.$$

According to Parseval's theorem, the $\mathcal{L}_2$ norm defined in time domain and the one defined in frequency domain are equivalent, i.e. $\|\hat{u}(j\omega)\|_2 = \|u(t)\|_2$. The relation is very useful while analyzing the system and deriving the controller.

### 2.1.3 Norms for Systems

Here we consider a linear time-varying (LTV) system $\mathcal{G}(t)$ in the form of

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t) \tag{2.6}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}(t)\boldsymbol{x}(t) \tag{2.7}$$

with $\boldsymbol{x}_0 = 0$, where $\boldsymbol{A}(t)$, $\boldsymbol{B}(t)$, $\boldsymbol{C}(t)$ are time-varying matrices, and $\boldsymbol{u}(t)$ is assumed to be unit variance white noise with zero mean. The interesting property of the system in this thesis is $H_2$ norm, so we define it as

$$\|\mathcal{G}(t)\|_2 := \lim_{T\to\infty} \frac{1}{T}\mathsf{E}\left[ (\int_0^T \boldsymbol{y}(t)^T \boldsymbol{y}(t) dt) \right]^{\frac{1}{2}} \tag{2.8}$$

in time domain, which is the root mean square (RMS) value of the output signal, and it can also be defined in frequency domain as

$$\|\hat{\mathcal{G}}(j\omega)\|_2 := \left( \frac{1}{2\pi} \int_{-\infty}^{\infty} \operatorname{tr}(\hat{\mathcal{G}}(j\omega))\hat{\mathcal{G}}^*(j\omega)))d\omega \right)^{\frac{1}{2}}, \tag{2.9}$$

where $\mathcal{G}^*$ means the complex conjugate transpose of $\mathcal{G}$. Analogous to the signal norms, (2.8) and (2.9) are identical according to Parseval's theorem. One can notice that there does not exist direct feedthrough from the input $\boldsymbol{u}$ to the output $\boldsymbol{y}$, i.e. strictly proper, so this results in finite $H_2$ norm.

Since it is challenging to compute $H_2$ norm in the frequency domain, we compute the norm in the state-space realization of system. Thus, the following result presents a way to calculate the $H_2$ norm in terms of the controllability or observability gramian [51].

**Theorem 4.** *Consider the LTV system* (2.7)*. The $H_2$ norm is determined by*

$$\begin{aligned}\|\mathcal{G}(t)\|_2 &= \lim_{T\to\infty}\frac{1}{T}\int_0^T \mathrm{tr}(\boldsymbol{C}(t)\boldsymbol{L}_c(t)\boldsymbol{C}^T(t))dt \\ &= \lim_{T\to\infty}\frac{1}{T}\int_0^T \mathrm{tr}(\boldsymbol{B}^T(t)\boldsymbol{B}^T(t))dt,\end{aligned}$$

*in which $\boldsymbol{L}_c(t)$ is the controllability gramian such that*

$$\dot{\boldsymbol{L}}_c(t) = \boldsymbol{A}(t)\boldsymbol{L}_c(t) + \boldsymbol{L}_c(t)\boldsymbol{A}^T(t) + \boldsymbol{B}(t)\boldsymbol{B}^T(t) \tag{2.10}$$

*with $\boldsymbol{L}_c(0) = 0$, and $\boldsymbol{L}_c(t)$ is the observability gramian such that*

$$-\dot{\boldsymbol{L}}_o(t) = \boldsymbol{L}_c(t)\boldsymbol{A}(t) + \boldsymbol{A}^T(t)\boldsymbol{L}_c(t) + \boldsymbol{C}^T(t)\boldsymbol{C}(t) \tag{2.11}$$

*with $\boldsymbol{L}_c(\infty) = 0$.*

*Proof.* The solution to the system $\mathcal{G}$ with white noise signal inputs is

$$\boldsymbol{y}(t) = \begin{cases} \lim_{T\to\infty}\frac{1}{T}\int_0^T \boldsymbol{C}(t)\Phi(t,\tau)\boldsymbol{B}(\tau)u(\tau)d\tau, & \text{for } t \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{2.12}$$

where $\Phi(t,\tau)$ is the transition matrix for $\boldsymbol{A}(t)$. According to the property of trace, we can rewrite the $H_2$ norm in time domain as

$$\|\mathcal{G}(t)\|_2 = \lim_{T\to\infty}\frac{1}{T}\mathbf{E}\left[(\int_0^T \mathrm{tr}(\boldsymbol{y}(t)\boldsymbol{y}(t)^T)dt)\right]^{\frac{1}{2}}, \tag{2.13}$$

and substituting (2.12) into (2.13) yields

$$
\begin{aligned}
\|\mathcal{G}(t)\|_2 &= \lim_{T\to\infty}\frac{1}{T}\int_0^T\int_0^T\int_0^T \operatorname{tr}(\boldsymbol{C}(t)\Phi(t,\tau)\boldsymbol{B}(\tau)\mathsf{E}\left[\boldsymbol{u}(\tau)\boldsymbol{u}(\sigma)\right]\boldsymbol{B}^T(\tau)\Phi^T(t,\tau)\boldsymbol{C}^T(t)d\tau d\sigma dt \\
&= \lim_{T\to\infty}\frac{1}{T}\int_0^T\int_0^T \operatorname{tr}(\boldsymbol{C}(t)\Phi(t,\tau)\boldsymbol{B}(\tau)\boldsymbol{B}^T(\tau)\Phi^T(t,\tau)\boldsymbol{C}^T(t))d\tau dt \qquad (2.14) \\
&= \lim_{T\to\infty}\frac{1}{T}\int_0^T \operatorname{tr}(\boldsymbol{C}(t)\boldsymbol{L}_c(t)\boldsymbol{C}^T(t))dt \qquad (2.15)
\end{aligned}
$$

with $\boldsymbol{L}_c(t) := \int_0^t \Phi(t,\tau)\boldsymbol{B}(\tau)\boldsymbol{B}^T(\tau)\Phi^T(t,\tau)d\tau$. Using Leibniz's rule and taking time derivative of (2.15), we arrive (2.10).

To prove the observability gramian (2.11), we rearrange (2.14) as

$$
\begin{aligned}
\|\mathcal{G}(t)\|_2 &= \lim_{T\to\infty}\frac{1}{T}\int_0^T\int_0^T \operatorname{tr}(\boldsymbol{B}^T(\tau)\Phi^T(t,\tau)\boldsymbol{C}^T(t)\boldsymbol{C}(t)\Phi(t,\tau)\boldsymbol{B}(\tau))d\tau dt \\
&= \lim_{T\to\infty}\frac{1}{T}\int_0^T \operatorname{tr}(\boldsymbol{B}(t)\boldsymbol{L}_o(t)\boldsymbol{B}^T(t))dt \qquad (2.16) \\
& \qquad (2.17)
\end{aligned}
$$

with $\boldsymbol{L}_o(t) := \lim_{T\to\infty}\int_\tau^T \Phi^T(t,\tau)\boldsymbol{C}^T(t)\boldsymbol{C}(t)\Phi(t,\tau)$. Similarly, taking time derivative of (2.16) renders (2.11). This concludes the theorem.

■

Different than LTI system, the differential matrix Lyapunov equations (2.10) and (2.11) involve the time-varying gramian functions. If the system $\mathcal{G}$ is asymptotically stable, two differential Lyapuvnov equation will converge to the solution constant matrices, which are the unique solution to the Lyapunov equations for the LTI system $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ [51].

## 2.2 Quadratic Stability

Based on the LPV system (2.5) and Theorem 1, we define a common stability condition of the LPV system.

**Definition 5.** *If the continuous differentiable function $V(\boldsymbol{x})$ is expressed in the form of*

$$V(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{P} \boldsymbol{x} \tag{2.18}$$

*where $\boldsymbol{P} >$ is a parameter-independent matrix, such that* (2.2)-(2.4) *are satisfied, the system* (2.5) *is said to be quadratically stable.*

One should notice that quadratic stability is only a sufficient condition of asymptotical stability of LPV systems, which means the system is not necessarily quadtatically stable if it is asymptotically stable. The following example [39] is shown to illustrate the relationship.

**Example 1.** *Consider an LPV system with matrix*

$$\boldsymbol{A}(\rho) = \begin{bmatrix} 1 & \rho \\ -\frac{4}{\rho} & -3 \end{bmatrix} \tag{2.19}$$

*with $\rho \in [-1, -\frac{1}{2}] \cup [1, \frac{1}{2}]$. The eigenvalues of $A(\rho)$ are all negative so that the system is asymptotically stable for all $\rho$ in the parameter set. However, it is not quadratically stable. To see this, we assume there exist a matrix*

$$\boldsymbol{P} = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} > 0$$

*such that the LMI*

$$\boldsymbol{M}(\rho) := \boldsymbol{A}(\rho)^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}(\rho) = \begin{bmatrix} 2p_1 - \frac{8p_2}{\rho} & p_1\rho - 2p_2 - \frac{4p_3}{\rho} \\ * & 2p_2\rho - 6p_3 \end{bmatrix} < 0$$

*holds all $\rho$ in the parameter set. This means $\boldsymbol{M}(\rho_0) < 0$ and $\boldsymbol{M}(-\rho_0) < 0$ for any $\rho \in [\frac{1}{2}1]$, and thus $\boldsymbol{M}(\rho_0) + \boldsymbol{M}(-\rho_0) < 0$. However,*

$$\begin{aligned} \boldsymbol{M}(\rho_0) + \boldsymbol{M}(-\rho_0) &= [\boldsymbol{A}(-\rho_0) + \boldsymbol{A}(\rho_0)]^T\boldsymbol{P} + \boldsymbol{P}[\boldsymbol{A}(-\rho_0) + \boldsymbol{A}(\rho_0)] \\ &= \begin{bmatrix} 4p_1 & -4p_2 \\ * & -12p_3 \end{bmatrix} \end{aligned}$$

*cannot be negative definite because of the positive term in left-upper block, which gives a contradiction. This concludes that the system is not quadratically stable.*

Besides, quadratic stability is very conservative since the Lyapunov function is not depending on the parameters due to the constant matrix $\boldsymbol{P}$ and it needs to satisfy (2.2)-(2.4) for all elements in the set of parameters.

With the Definition 5 we can arrive the following result.

**Theorem 6.** *Consider the system* (2.5)*. The following statements are equivalent.*

1. *The system* (2.5) *is quadratic stable.*

2. *There exist a symmetric matrix $\boldsymbol{P} > 0$ such that*

$$\boldsymbol{A}^T(\boldsymbol{\rho})\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}(\boldsymbol{\rho}) < 0 \tag{2.20}$$

*holds for all $\boldsymbol{\rho} \in \mathcal{D}_{\boldsymbol{\rho}}$.*

*Proof.* $1 \rightarrow 2$ : First, pre- and post-multiplying (2.20) by $\boldsymbol{x}^T$ and $\boldsymbol{x}$ respectively, we get

$$\begin{aligned}\boldsymbol{x}^T(\boldsymbol{A}(\boldsymbol{\rho})^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}(\boldsymbol{\rho}))\boldsymbol{x} &= \dot{x}^T\boldsymbol{P}\boldsymbol{x} + \boldsymbol{x}^T P\dot{\boldsymbol{x}} \\ &= \frac{d}{dt}(\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}) < 0\end{aligned}$$

Integrating on both side from $\tau = t$ to $\tau \rightarrow \infty$ we arrive

$$\begin{aligned}&\int_{\tau=t}^{\infty} \frac{d}{d\tau}(\boldsymbol{x}(\tau)^T\boldsymbol{P}\boldsymbol{x}(\tau)) < 0 \\ &\rightarrow \lim_{\tau\rightarrow\infty} \boldsymbol{x}(\tau)^T\boldsymbol{P}\boldsymbol{x}(\tau) - \boldsymbol{x}(t)^T\boldsymbol{P}\boldsymbol{x}(t) < 0.\end{aligned}$$

From the statement 1, we know $\lim_{\tau\rightarrow\infty} \boldsymbol{x}(\tau) = 0$, and this yields $\lim_{\tau\rightarrow\infty} \boldsymbol{x}(\tau)^T\boldsymbol{P}\boldsymbol{x}(\tau) = 0$. Therefore, we can conclude $\boldsymbol{x}(t)^T\boldsymbol{P}\boldsymbol{x}(t) > 0$ or $\boldsymbol{P} > 0$, and the statement 2 follows.

$2 \rightarrow 1$ : We let $V(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}$. Since $\boldsymbol{P}$ is positive definite, (2.2) and (2.3) are thus automatically satisfied. Taking time derivative of $V$ we get

$$\begin{aligned}\dot{V} &= \dot{x}^T\boldsymbol{P}\boldsymbol{x} + \boldsymbol{x}^T P\dot{\boldsymbol{x}} \\ &= \boldsymbol{x}^T(\boldsymbol{A}(\boldsymbol{\rho})^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}(\boldsymbol{\rho}))\boldsymbol{x}.\end{aligned}$$

Therefore, (2.4) renders (2.20) and the system (2.5) is asymptotically stable from Theorem 1. Finally, from the Definition 5 we can conclude statement 1. ■

(2.20) is a semi-infinite dimensional LMI that can be solved in two ways commonly. If the system is polytopic, i.e. $\boldsymbol{A}(\boldsymbol{\rho}) = \boldsymbol{A}_0 + \sum_{i=1}^{n_\rho} \rho_i\boldsymbol{A}_i$, the LMI (2.20) can be transformed into $2^{n_\rho}$ finite-dimensional LMIs. If the system is generic, we can set up a discretization of parameter set so that the semi-infinite dimensional LMIs is approximated as a finite dimensional LMIs. In this thesis, we will be focusing on the gridding approach and will be discussed later.

## 2.3 Robust Stability

Rather than a parameter-independent matrix used in the Lyapunov function, $\boldsymbol{P}$ can actually be more adaptive to the parameters and carry out less conservatism. We thus give the following definition.

**Definition 7.** *If the continuous differentiable function $V(\boldsymbol{x})$ is expressed in the form of*

$$V(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{P}(\boldsymbol{\rho})\boldsymbol{x} \tag{2.21}$$

*with $\boldsymbol{P}(\boldsymbol{\rho}) > 0$ a parameter-dependent matrix, such that* (2.2)-(2.4) *are satisfied, the system* (2.5) *is said to be robustly stable.*

Comparing with quadratic stability, robust stability is only a sufficient condition for quadratic stability, and the quadratic stability is a special case of robust stability where the matrix $\boldsymbol{P}$ is a constant. Another example in [39] shows the relations.

**Example 2.** *Consider the same example system stated in* (2.19), *and a parameter-dependent matrix*

$$\boldsymbol{P}(\rho) = \boldsymbol{P}_0 + \boldsymbol{P}_1\rho + \boldsymbol{P}_2\rho^2 = \begin{bmatrix} p_1(\rho) & p_2(\rho) \\ * & p_3(\rho) \end{bmatrix}$$

*As previous example we check $\boldsymbol{M}(\rho_0) + \boldsymbol{M}(-\rho_0) < 0$ numerically which renders*

$$\boldsymbol{P}(\rho) = \begin{bmatrix} 50 + 6\rho^2 & 16\rho \\ * & 1 + 7\rho^2 \end{bmatrix} > 0$$

*Therefore, the system is robustly stable.*

Beside, it is obvious to see that a parameter-dependent matrix function gives more freedom on determination of Lyapunov function and yields less conservatism.

With the Definition 7, the following result is given.

**Theorem 8.** *Consider the system* (2.5)*. If there there exists a continuous differentiable matrix function* $\boldsymbol{P}(\boldsymbol{\rho}) > 0$ *such that*

$$\boldsymbol{A}(\boldsymbol{\rho})^T\boldsymbol{P}(\boldsymbol{\rho}) + \boldsymbol{P}(\boldsymbol{\rho})\boldsymbol{A}(\boldsymbol{\rho}) + \sum_{i=1}^{n_{\boldsymbol{\rho}}} \dot{\rho}_i \frac{\partial \boldsymbol{P}(\boldsymbol{\rho})}{\partial \rho_i} < 0 \tag{2.22}$$

*for all* $(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) \in \mathcal{D}_{\boldsymbol{\rho}} \times \mathcal{D}_{\dot{\boldsymbol{\rho}}}$*, the system* (2.5) *is robustly stable.*

*Proof.* We let $V(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{P}(\boldsymbol{\rho})\boldsymbol{x}$. Since $\boldsymbol{P}(\boldsymbol{\rho})$ is positive definite, (2.2) and (2.3) are thus automatically satisfied. Taking time derivative of $V$ we get

$$\begin{aligned}\dot{V} &= \dot{x}^T\boldsymbol{P}(\boldsymbol{\rho})\boldsymbol{x} + \boldsymbol{x}^T P(\boldsymbol{\rho})\dot{\boldsymbol{x}} + \boldsymbol{x}^T(\sum_{i=1}^{n_{\boldsymbol{\rho}}} \dot{\rho}_i \frac{\partial \boldsymbol{P}(\boldsymbol{\rho})}{\partial \rho_i})\boldsymbol{x} \\ &= \boldsymbol{x}^T(\boldsymbol{A}(\boldsymbol{\rho})^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}(\boldsymbol{\rho}) + \sum_{i=1}^{n_{\boldsymbol{\rho}}} \dot{\rho}_i \frac{\partial \boldsymbol{P}(\boldsymbol{\rho})}{\partial \rho_i})\boldsymbol{x}.\end{aligned}$$

Therefore, (2.4) renders (2.22) and the system (2.5) is asymptotically stable from Theorem 1. Finally, from the Definition 7 the system is robustly stable. ■

Different than (2.20), since $\boldsymbol{P}(\boldsymbol{\rho})$ is infinite-dimensional as well, the LMI (2.22) becomes infinite-dimensional. Therefore, a further structure of $\boldsymbol{P}(\boldsymbol{\rho})$ should be defined. For example, for the LPV system in a polytopic form, one can define $\boldsymbol{P}(\boldsymbol{\rho})$ as an affine function of $\boldsymbol{\rho}$ and the LMIs can be solved by substituting extreme values in the parameter set in each dimension. For a general LPV system, $\boldsymbol{P}(\boldsymbol{\rho})$ is defined by several basis function of $\boldsymbol{\rho}$ and one can solve it via gridding method as mentioned before. This is still an open question about the choice of basis functions and grid points to get an optimal solution.

|  | $\|u\|_2$ | $\|u\|_\infty$ | $\mathbf{pow}(u)$ |
|---|---|---|---|
| $\|y\|_2$ | $\|\hat{G}(j\omega)\|_\infty$ | $\infty$ | $\infty$ |
| $\|y\|_\infty$ | $\|\hat{G}(j\omega)\|_2$ | $\|\hat{G}(j\omega)\|_1$ | $\infty$ |
| $\mathbf{pow}(y)$ | 0 | $\leq \|\hat{G}(j\omega)\|_\infty$ | $\|\hat{G}(j\omega)\|_\infty$ |

Table 2.1: The relationship between different types of inputs and outputs, where $u$ represents input and $y$ indicates output.

## 2.4 $H_2$ Performance

In addition to the stability of the system, the performance of the system is also an important factor to design a controller. In this thesis we are concerning the bounds for desired output given an input with certain energy. In other words, the extreme value of the output function should be under a desired quantity. Table 2.1 shows the relationship between inputs and outputs in the norm [1]. To match our concerning objective, it is obvious to choose $H_2$ norm of the system so that the output can be bounded. Therefore, in this section we present some analytical results that involves several conditions for $H_2$ performance of LPV systems.

We consider an autonomous LPV system with disturbances in the form of

$$\hat{\boldsymbol{G}}(j\omega): \begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}(\boldsymbol{\rho})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\rho})\boldsymbol{w} \\ \boldsymbol{z} &= \boldsymbol{C}_z(\boldsymbol{\rho})\boldsymbol{x} \end{aligned} \tag{2.23}$$

with an initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$, where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is the state vector of the system, $\boldsymbol{z}$ is the vector of desire outputs that we are interested in, $\boldsymbol{w}$ is the vector of the exogenous signals, e.g. disturbance, sensor noise, reference, $\boldsymbol{A}(\boldsymbol{\rho}) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{B}(\boldsymbol{\rho}) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_x \times m}$, and $\boldsymbol{C}_z(\boldsymbol{\rho}) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_z}$ are the system matrices that are continuous differentiable matrix functions of the parameters $\boldsymbol{\rho}(t)$. Let the initial condition $\boldsymbol{x}(0) = 0$ and $\boldsymbol{w}$ is a vector of zero-mean white noise with an identity power spectrum, then, analogous to (2.8), an $H_2$ norm in time-domain is defined as

$$\|\boldsymbol{G}(t)\|_2 := \lim_{T\to\infty} \frac{1}{T}\mathsf{E}\left[(\int_0^T \boldsymbol{z}(t)^T\boldsymbol{z}(t)dt)\right]^{\frac{1}{2}}, \tag{2.24}$$

which can be computed through

$$\|\boldsymbol{G}(t)\|_2 = \lim_{T\to\infty} \frac{1}{T}\int_0^T \operatorname{tr}(\boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{L}_c(\boldsymbol{\rho}(t))\boldsymbol{C}^T(\boldsymbol{\rho}(t)))dt$$

with controllability gramian $\boldsymbol{L}_c(\boldsymbol{\rho}(t))$ solved by differential matrix Lyapunov equation

$$\dot{\boldsymbol{L}}_c(\boldsymbol{\rho}(t)) = \boldsymbol{A}(\boldsymbol{\rho}(t))\boldsymbol{L}_c(\boldsymbol{\rho}(t)) + \boldsymbol{L}_c(\boldsymbol{\rho}(t))\boldsymbol{A}^T(\boldsymbol{\rho}(t)) + \boldsymbol{B}(\boldsymbol{\rho}(t))\boldsymbol{B}^T(\boldsymbol{\rho}(t)), \tag{2.25}$$

or calculated through

$$\|\boldsymbol{G}(t)\|_2 = \lim_{T\to\infty} \frac{1}{T}\int_0^T \operatorname{tr}(\boldsymbol{B}(\boldsymbol{\rho}(t))\boldsymbol{L}_o(\boldsymbol{\rho}(t))\boldsymbol{B}^T(\boldsymbol{\rho}(t)))dt$$

with observability gramian $\boldsymbol{L}_o(\boldsymbol{\rho}(t))$ solved by

$$\dot{\boldsymbol{L}}_o(\boldsymbol{\rho}(t)) = \boldsymbol{B}(\boldsymbol{\rho}(t))\boldsymbol{L}_o(\boldsymbol{\rho}(t)) + \boldsymbol{L}_o(\boldsymbol{\rho}(t))\boldsymbol{B}^T(\boldsymbol{\rho}(t)) + \boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{C}^T(\boldsymbol{\rho}(t)).$$

Furthermore, rather than find a solution to the differential Lyapunov equation, one can solve the parameter-dependent LMIs via the following result.

**Theorem 9.** *Consider the LPV system* (2.23). *If there exist two parameter-independent matrices* $\boldsymbol{P} = \boldsymbol{P}^T > 0$ *and* $\boldsymbol{W} = \boldsymbol{W}^T > 0$ *such that*

$$\boldsymbol{A}(\boldsymbol{\rho}(t))\boldsymbol{X} + \boldsymbol{X}\boldsymbol{A}^T(\boldsymbol{\rho}(t)) + \boldsymbol{B}(\boldsymbol{\rho}(t))\boldsymbol{B}^T(\boldsymbol{\rho}(t)) < 0 \tag{2.26}$$

$$\begin{bmatrix} \boldsymbol{W} & \boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{X} \\ \boldsymbol{X}^T\boldsymbol{C}^T(\boldsymbol{\rho}(t)) & \boldsymbol{X}^{-1} \end{bmatrix} > 0 \tag{2.27}$$

*for all* $\boldsymbol{\rho}(t) \in \mathcal{D}_{\boldsymbol{\rho}}$*, then the system is asymptotically stable and* $\|\hat{\boldsymbol{G}}(j\omega)\|_2 < \sqrt{\mathrm{tr}(\boldsymbol{W})}$*.*

*Proof.* (2.26) implies $\boldsymbol{A}(\boldsymbol{\rho}(t))\boldsymbol{X} + \boldsymbol{X}\boldsymbol{A}^T(\boldsymbol{\rho}(t) < 0$, or $\boldsymbol{Y}\boldsymbol{A}(\boldsymbol{\rho}(t)) + \boldsymbol{A}^T(\boldsymbol{\rho}(t)\boldsymbol{Y} < 0$ with $\boldsymbol{X}^{-1} = \boldsymbol{Y}$, so the system $\hat{\boldsymbol{G}}(j\omega)$ is quadratically stable according to Theorem 6, i.e. asymptotically stable.

(2.25) subtracted from (2.26) yields

$$\boldsymbol{A}(\boldsymbol{\rho}(t))(\boldsymbol{X} - \boldsymbol{L}_c) + (\boldsymbol{X} - \boldsymbol{L}_c)\boldsymbol{A}^T(\boldsymbol{\rho}(t)) + \dot{\boldsymbol{L}}_c < 0. \tag{2.28}$$

Since the quadratic stability is a sufficient condition of the robust stability, (2.28) renders $\boldsymbol{X} - \boldsymbol{L}_c > 0$ which implies $\|\hat{\boldsymbol{G}}(j\omega)\|_2^2 < \lim_{T\to\infty} \frac{1}{T}\int_0^T \mathrm{tr}(\boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{X}\boldsymbol{C}^T(\boldsymbol{\rho}(t)))$. In addition, (2.27) implies $\boldsymbol{W} > \boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{X}\boldsymbol{C}(\boldsymbol{\rho}(t))$, and thus $\mathrm{tr}(\boldsymbol{W}) > \mathrm{tr}(\boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{X}\boldsymbol{C}(\boldsymbol{\rho}(t)))$. As a result, the $H_2$ of the system is bounded by $\mathrm{tr}(\boldsymbol{W})$ which concludes the result. ■

**Remark.** *From the proof, the system is shown to be quadratically stable, which is very conservative, since two parameter-independent matrices confine the feasible set of* (2.26) *and* (2.27)*. In addition, the relationship between* $\boldsymbol{L}_c$*,* $\boldsymbol{X}$*,* $\boldsymbol{W}$ *is*

$$\mathrm{tr}(\boldsymbol{W}) > \mathrm{tr}(\boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{X}\boldsymbol{C}(\boldsymbol{\rho}(t)) > \mathrm{tr}(\boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{L}_c(\boldsymbol{\rho}(t))\boldsymbol{C}(\boldsymbol{\rho}(t)),$$

*so the upper bound of the norm in Theorem 9 is supremum of* $\mathrm{tr}(\boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{L}_c(\boldsymbol{\rho}(t))\boldsymbol{C}(\boldsymbol{\rho}(t))$*. Moreover,* (2.26) *and* (2.27) *are only sufficient conditions of* $H_2$ *performance of the system* $\hat{\boldsymbol{G}}(j\omega)$*. Therefore, if the system is stable and has a bounded* $H_2$ *performance, there does not necessarily exist a solution to* (2.26) *and* (2.27)*.*

To reduce the conservatism, the conditions for $H_2$ performance can be reconsider as parameter dependence, which yields the following result.

**Theorem 10.** *Consider the LPV system* (2.23)*. If there exist two parameter-dependent matrices*

$\boldsymbol{P}(\boldsymbol{\rho}(t)) = \boldsymbol{P}(\boldsymbol{\rho}(t))^T > 0$ *and* $\boldsymbol{W}(\boldsymbol{\rho}(t)) = \boldsymbol{W}(\boldsymbol{\rho}(t))^T > 0$ *such that*

$$\boldsymbol{A}(\boldsymbol{\rho}(t))\boldsymbol{X}(\boldsymbol{\rho}(t)) + \boldsymbol{X}(\boldsymbol{\rho}(t))\boldsymbol{A}^T(\boldsymbol{\rho}(t)) + \boldsymbol{B}(\boldsymbol{\rho}(t))\boldsymbol{B}^T(\boldsymbol{\rho}(t)) - \dot{\boldsymbol{X}}(\boldsymbol{\rho}(t), \dot{\boldsymbol{\rho}}(t)) < 0 \tag{2.29}$$

$$\begin{bmatrix} \boldsymbol{W}(\boldsymbol{\rho}(t)) & \boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{X}(\boldsymbol{\rho}(t)) \\ \boldsymbol{X}(\boldsymbol{\rho}(t))\boldsymbol{C}^T(\boldsymbol{\rho}(t)) & \boldsymbol{X}^{-1}(\boldsymbol{\rho}(t)) \end{bmatrix} > 0 \tag{2.30}$$

*for all* $\boldsymbol{\rho}(t) \in \mathcal{D}_{\boldsymbol{\rho}}$ *and* $\dot{\boldsymbol{\rho}}(t) \in \mathcal{D}_{\dot{\boldsymbol{\rho}}}$*, then the system is asymptotically stable and* $\|\hat{\boldsymbol{G}}(j\omega)\|_2 < \sqrt{\lim_{T\to\infty} \frac{1}{T}\int_0^T \mathrm{tr}(\boldsymbol{W}(\boldsymbol{\rho}(t)))}$.

*Proof.* (2.29) implies

$$\boldsymbol{A}(\boldsymbol{\rho}(t))\boldsymbol{X}(\boldsymbol{\rho}(t)) + \boldsymbol{X}(\boldsymbol{\rho}(t))\boldsymbol{A}^T(\boldsymbol{\rho}(t)) - \dot{\boldsymbol{X}}(\boldsymbol{\rho}(t)) < 0,$$

or

$$\boldsymbol{Y}(\boldsymbol{\rho}(t))\boldsymbol{A}(\boldsymbol{\rho}(t)) + \boldsymbol{A}^T(\boldsymbol{\rho}(t))(\boldsymbol{\rho}(t)) + \dot{\boldsymbol{Y}}(\boldsymbol{\rho}(t)) < 0$$

with $\boldsymbol{X}^{-1}(\boldsymbol{\rho}(t)) = \boldsymbol{Y}(\boldsymbol{\rho}(t))$ and an identity that

$$\begin{aligned} \dot{\boldsymbol{Y}}(\boldsymbol{\rho}(t)) &= \sum_{k=1}^{n_\rho} \dot{\boldsymbol{\rho}}_k \frac{\partial \boldsymbol{Y}(\boldsymbol{\rho}(t))}{\partial \boldsymbol{\rho}_k} \\ &= \sum_{k=1}^{n_\rho} \dot{\boldsymbol{\rho}}_k \frac{\partial \boldsymbol{X}^{-1}(\boldsymbol{\rho}(t))}{\partial \boldsymbol{\rho}_k} \\ &= -\boldsymbol{X}^{-1} \sum_{k=1}^{n_\rho} \dot{\boldsymbol{\rho}}_k \frac{\partial \boldsymbol{X}(\boldsymbol{\rho}(t))}{\partial \boldsymbol{\rho}_k} \boldsymbol{X}^{-1}, \end{aligned} \tag{2.31}$$

so the system $\hat{\boldsymbol{G}}(j\omega)$ is robustly stable according to Theorem 8, i.e. asymptotically stable.

(2.25) subtracted from (2.29) yields

$$\boldsymbol{A}(\boldsymbol{\rho}(t))(\boldsymbol{X}(\boldsymbol{\rho}(t)) - \boldsymbol{L}_c) + (\boldsymbol{X}(\boldsymbol{\rho}(t)) - \boldsymbol{L}_c)\boldsymbol{A}^T(\boldsymbol{\rho}(t)) + (\dot{\boldsymbol{L}}_c - \dot{\boldsymbol{X}}(\boldsymbol{\rho}(t))) < 0. \tag{2.32}$$

Since the robust stability has been confirmed, (2.32) renders $\boldsymbol{X}(\boldsymbol{\rho}(t)) - \boldsymbol{L}_c(\boldsymbol{\rho}(t)) > 0$ which implies $\|\hat{\boldsymbol{G}}(j\omega)\|_2 < \lim_{T\to\infty} \frac{1}{T}\int_0^T \operatorname{tr}(\boldsymbol{C}(\boldsymbol{\rho}(t)\boldsymbol{X}(\boldsymbol{\rho}(t)\boldsymbol{C}^T(\boldsymbol{\rho}(t)))))$. In addition, (2.30) implies $\boldsymbol{W}(\boldsymbol{\rho}(t)) > \boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{X}(\boldsymbol{\rho}(t))\boldsymbol{C}(\boldsymbol{\rho}(t))$, and thus $\operatorname{tr}(\boldsymbol{W}(\boldsymbol{\rho}(t))) > \operatorname{tr}(\boldsymbol{C}(\boldsymbol{\rho}(t))\boldsymbol{X}(\boldsymbol{\rho}(t))\boldsymbol{C}(\boldsymbol{\rho}(t)))$. As a result, the $H_2$ of the system is bounded by $\operatorname{tr}(\boldsymbol{W}(\boldsymbol{\rho}(t)))$ which concludes the result. ■

**Remark.** *Comparing with Theorem 9, the $H_2$ norm bounded by a RMS value of $\operatorname{tr}(\boldsymbol{W}(\boldsymbol{\rho}(t)))$ is less conservative because it is adaptive to the parameters. The stability also involves less conservatism since the system is robustly stable. However, the challenge in Theorem 10 will show up in terms of computation since* (2.29) *and* (2.30) *are in infinite dimensional that requires more computational effort. This will be seen in the later section.*

### 2.5 Linear Quadratic Regulator

In this section, we consider the LPV dynamical system (1.1) and we are interested in a full-state feedback control $\boldsymbol{u} = \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{x}$ that minimizes the cost function

$$J := \int_0^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})\,dt. \tag{2.33}$$

Based on the objective, we have the following result.

**Theorem 11.** *Consider the LPV system* (1.1) *with the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ and control law $\boldsymbol{u} = \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{x}$. Given two positive definite constant matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, if there exists a parameter-dependent matrix $\boldsymbol{P}(\boldsymbol{\rho}) = \boldsymbol{P}^T(\boldsymbol{\rho}) > 0$ such that*

$$\boldsymbol{P}(\boldsymbol{\rho})\left(\boldsymbol{A}(\boldsymbol{\rho}) + \boldsymbol{B}(\boldsymbol{\rho})\boldsymbol{K}(\boldsymbol{\rho})\right) + (\bullet)^T\boldsymbol{P}(\boldsymbol{\rho}) + \sum_i^{n_\rho} \frac{\partial \boldsymbol{P}(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}_i}\dot{\boldsymbol{\rho}}_i + \boldsymbol{Q} + \boldsymbol{K}^T(\boldsymbol{\rho})\boldsymbol{R}\boldsymbol{K}(\boldsymbol{\rho}) < 0 \tag{2.34}$$

*for all $\boldsymbol{\rho} \in \mathcal{D}_{\boldsymbol{\rho}}$ and $\dot{\boldsymbol{\rho}} \in \mathcal{D}_{\boldsymbol{\rho}}$, then the system is asymptotically stable $0 < J < V(\boldsymbol{x}_0)$.*

*Proof.* Using standard arguments[52], suppose $\exists\, V(\boldsymbol{x}, \boldsymbol{\rho}) > 0$, which is a scalar function of states, such that

$$\frac{dV(\boldsymbol{x}, \boldsymbol{\rho})}{dt} \leq -\left(\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u}\right), \tag{2.35}$$

which implies the asymptotic stability since $\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive definite. Integrating from $[0, T]$ gives us

$$\begin{aligned} \int_0^T \frac{dV(\boldsymbol{x}, \boldsymbol{\rho})}{dt} dt &\leq -\int_0^T \left(\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u}\right) dt, \\ V(\boldsymbol{x}(T), \boldsymbol{\rho}(T)) - V(\boldsymbol{x}(0), \boldsymbol{\rho}(0)) &\leq -\int_0^T \left(\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u}\right) dt. \end{aligned}$$

$V(\boldsymbol{x}(T), \boldsymbol{\rho}(T)) \geq 0$ implies

$$-V(\boldsymbol{x}(0), \boldsymbol{\rho}(0)) \leq -\int_0^T \left(\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u}\right) dt, \text{ for all } T > 0, \tag{2.36}$$

or,

$$V(\boldsymbol{x}(0), \boldsymbol{\rho}(0)) \geq \int_0^\infty \left(\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u}\right) dt = J. \tag{2.37}$$

Therefore, (2.35) provides a sufficient condition for upper bound on the cost-to-go, and the control law is synthesized as an optimization problem that minimizes the upper bound $V(\boldsymbol{x}(0))$ and is subject to (2.35).

Regarding the LPV system, we substitute the control law $\boldsymbol{u} = \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{x}$ into (2.35), let $V(\boldsymbol{x}, \boldsymbol{\rho}) := \boldsymbol{x}^T \boldsymbol{P}(\boldsymbol{\rho}) \boldsymbol{x}$, and take the time derivative of it, then we arrive

$$\begin{aligned} \dot{V}(\boldsymbol{x}, \boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) &= \mathsf{sym}\left(\boldsymbol{x}^T \left(\boldsymbol{A}(\boldsymbol{\rho}) + \boldsymbol{B}(\boldsymbol{\rho})\boldsymbol{K}(\boldsymbol{\rho})\right)^T \boldsymbol{P}(\boldsymbol{\rho}) \boldsymbol{x}\right) + \boldsymbol{x}^T \dot{\boldsymbol{P}}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) \boldsymbol{x} \\ &= \boldsymbol{x}^T \left(\mathsf{sym}\left(\boldsymbol{P}(\boldsymbol{\rho})\left(\boldsymbol{A}(\boldsymbol{\rho}) + \boldsymbol{B}(\boldsymbol{\rho})\boldsymbol{K}(\boldsymbol{\rho})\right)\right) + \sum_i^{n_\rho} \frac{\partial \boldsymbol{P}(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}_i} \dot{\boldsymbol{\rho}}_i\right) \boldsymbol{x}. \end{aligned} \tag{2.38}$$

Substituting (2.38) into (2.35), we arrive (2.34) which holds for all $\boldsymbol{\rho} \in \mathcal{D}_{\boldsymbol{\rho}}$ and $\dot{\boldsymbol{\rho}} \in \mathcal{D}_{\boldsymbol{\rho}}$. Finally, we conclude the theorem. ■

It is a challenge to solve (2.34) directly due to non-convexity. However, it can be convexified by using LMI techniques, which gives the following result.

**Theorem 12.** *Consider the LPV system* (1.1) *with the initial condition* $\boldsymbol{x}(0) = \boldsymbol{x}_0$ *and control law* $\boldsymbol{u} = \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{x}$*. Given two positive definite constant matrices* $\boldsymbol{Q}$ *and* $\boldsymbol{R}$*, if a constant matrix* $\boldsymbol{Z} = \boldsymbol{Z}^T > 0$ *and two parameter-dependent matrices* $\boldsymbol{Y}(\boldsymbol{\rho}) = \boldsymbol{Y}^T(\boldsymbol{\rho}) > 0$ *and* $\boldsymbol{W}(\boldsymbol{\rho})$ *are solutions to the optimization*

$$\underset{\boldsymbol{Y},\boldsymbol{W},\boldsymbol{Z}}{\mathit{max}} \quad \mathrm{tr}(\boldsymbol{Z}) \tag{2.39}$$

$$\textit{subject to} \quad \begin{bmatrix} \boldsymbol{A}((\boldsymbol{\rho}))\boldsymbol{Y}(\boldsymbol{\rho}) + \boldsymbol{B}(\boldsymbol{\rho})\boldsymbol{W}(\boldsymbol{\rho}) + (\bullet)^T - \sum_i^{n_\rho} \frac{\partial \boldsymbol{Y}(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}_i}\dot{\boldsymbol{\rho}}_i & \boldsymbol{Y}(\boldsymbol{\rho}) & \boldsymbol{W}^T(\boldsymbol{\rho}) \\ \boldsymbol{Y}(\boldsymbol{\rho}) & -\boldsymbol{Q}^{-1} & \boldsymbol{0} \\ \boldsymbol{W}(\boldsymbol{\rho}) & \boldsymbol{0} & -\boldsymbol{R}^{-1} \end{bmatrix} < 0, \tag{2.40}$$

$$\boldsymbol{Y}(\boldsymbol{\rho}) - \boldsymbol{Z} > 0, \tag{2.41}$$

$$\boldsymbol{\rho}(t) \in \mathcal{D}_{\boldsymbol{\rho}},$$

$$\dot{\boldsymbol{\rho}}(t) \in \mathbb{D}_{\dot{\boldsymbol{\rho}}},$$

*where* $\mathbb{D}_{\dot{\boldsymbol{\rho}}}$ *is the vertices of the set* $\mathcal{D}_{\dot{\boldsymbol{\rho}}}$*, then the control gain determined by* $\boldsymbol{K}(\boldsymbol{\rho}) = \boldsymbol{Y}^{-1}(\boldsymbol{\rho})\boldsymbol{W}(\boldsymbol{\rho})$ *stabilize the system and minimizes the cost function* $J$*.*

*Proof.* First of all, let $\boldsymbol{Y}(\boldsymbol{\rho}) = \boldsymbol{P}^{-1}(\boldsymbol{\rho})$, and pre and post-multiply (2.34) with $\boldsymbol{Y}(\boldsymbol{\rho})$, then with

$\boldsymbol{W}(\boldsymbol{\rho}) = \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{Y}(\boldsymbol{\rho})$ it becomes

$$\mathsf{sym}\left(\boldsymbol{A}(\boldsymbol{\rho})\boldsymbol{Y}(\boldsymbol{\rho}) + \boldsymbol{B}(\boldsymbol{\rho})\boldsymbol{W}(\boldsymbol{\rho})\right) + \boldsymbol{Y}(\boldsymbol{\rho})\sum_{i}^{n_\rho}\frac{\partial \boldsymbol{P}(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}_i}\dot{\boldsymbol{\rho}}_i\boldsymbol{Y}(\boldsymbol{\rho}) + \boldsymbol{Y}(\boldsymbol{\rho})\boldsymbol{Q}\boldsymbol{Y}(\boldsymbol{\rho}) \\ + \boldsymbol{W}^T(\boldsymbol{\rho})\boldsymbol{R}\boldsymbol{W}(\boldsymbol{\rho}) < 0. \tag{2.42}$$

According to the identity in (2.31), (2.42) can be refined as

$$\boldsymbol{A}(\boldsymbol{\rho})\boldsymbol{Y}(\boldsymbol{\rho}) + \boldsymbol{B}(\boldsymbol{\rho})\boldsymbol{W}(\boldsymbol{\rho}) + (\bullet)^T - \sum_{i}^{n_\rho}\frac{\partial \boldsymbol{Y}(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}_i}\dot{\boldsymbol{\rho}}_i + \boldsymbol{Y}(\boldsymbol{\rho})\boldsymbol{Q}\boldsymbol{Y}(\boldsymbol{\rho}) \\ + \boldsymbol{W}^T(\boldsymbol{\rho})\boldsymbol{R}\boldsymbol{W}(\boldsymbol{\rho}) < 0. \tag{2.43}$$

Applied Schur's complement to (2.43) yields (2.40). Since (2.40) is affine in $\dot{\boldsymbol{\rho}}$, only the vertices of $\mathcal{D}_{\dot{\boldsymbol{\rho}}}$ is needed, i.e. $\mathbb{D}_{\dot{\boldsymbol{\rho}}}$.

From (2.37), the cost function $J$ is upper-bounded by $V(\boldsymbol{x}_0, \boldsymbol{\rho}(0))$, so if $V(\boldsymbol{x}_0, \boldsymbol{\rho}(0))$ is minimized, $J$ is also minimized. It is known that $V(\boldsymbol{x}_0, \boldsymbol{\rho}(0)) = \boldsymbol{x}_0^T\boldsymbol{P}(\boldsymbol{\rho}(0))\boldsymbol{x}_0 = \boldsymbol{x}_0^T\boldsymbol{Y}^{-1}(\boldsymbol{\rho}(0))\boldsymbol{x}_0$, so if one can minimize $\boldsymbol{P}(\boldsymbol{\rho}(0))$, or maximize $\boldsymbol{Y}(\boldsymbol{\rho}(0))$. $V(\boldsymbol{x}_0, \boldsymbol{\rho}(0))$ is minimized. However, $\boldsymbol{\rho}(0)$ is unknown until it is measured, so instead of maximizing $\boldsymbol{Y}(\boldsymbol{\rho}(0))$, another constant matrix $\boldsymbol{Z}$ is introduced such that (2.41) holds for all $\boldsymbol{\rho} \in \mathcal{D}_{\boldsymbol{\rho}}$. Therefore, maximizing $\mathrm{tr}(\boldsymbol{Z})$ implies minimizing $V(\boldsymbol{x}_0, \boldsymbol{\rho}(0))$ and so as $J$, and then the theorem follows. ■

**Remark.** *Theorem* (12) *presents a way to synthesize an LQR control via solving a convex optimization problem with parameter-dependent LMIs* (2.40) *and* (2.41). *There are two parts of mechanisms in the problem. One is the sufficient condition* (2.40) *that ensures there exists an upper bound* $V(\boldsymbol{x}_0, \boldsymbol{\rho}(0))$ *for the cost function* $J$*. The other one is the maximization of* $\mathrm{tr}(\boldsymbol{Z})$ *and the sufficient condition* (2.41) *for another upper bound for* $\boldsymbol{Y}(\boldsymbol{\rho})$*. Using the parameter-dependent feedback gain from the Theorem 12, the final controlled system is guaranteed to be optimal in the sense of states and control efforts.*

## 2.6 $H_2$ optimal Control

This section considers another LPV control system with a disturbance expressed as

$$\hat{\boldsymbol{G}}_c(j\omega) : \begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}(\boldsymbol{\rho})\boldsymbol{x} + \boldsymbol{B}_u\boldsymbol{u} + \boldsymbol{B}_w\boldsymbol{w} \\ \boldsymbol{z} &= \boldsymbol{C}_z(\boldsymbol{\rho})\boldsymbol{x} + \boldsymbol{D}_z(\boldsymbol{\rho})\boldsymbol{u}, \end{aligned} \tag{2.44}$$

with a known initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$, where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is the state vector of the system, $\boldsymbol{z} \in \mathbb{R}^{n_z}$ is the vector of desire outputs that we are interested in, $\boldsymbol{w}$ is a vector of zero-mean white noise with an identity power spectrum, $\boldsymbol{A}(\boldsymbol{\rho}) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{B}_u(\boldsymbol{\rho}) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_x \times n_u}$, $\boldsymbol{B}_w(\boldsymbol{\rho}) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_x \times n_w}$, $\boldsymbol{C}_z(\boldsymbol{\rho}) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_z}$, and $\boldsymbol{D}_z(\boldsymbol{\rho}) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_u}$ are the system matrices that are continuous differentiable matrix functions of the parameters $\boldsymbol{\rho}(t)$. The objective is to find a full-state feedback control gain $\boldsymbol{K}(\boldsymbol{\rho})$ such that the desired outputs keep small under disturbances, which involves two tasks: bounded output and disturbance rejection. Accordingly, the $H_2$ norm is well suited; that is, under a disturbance with given energy the bounds of the desire outputs are minimized. The control law thus is $\boldsymbol{u} = \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{x}$, and substituting the feedback control to (2.44) yields

$$\hat{\boldsymbol{G}}_c(j\omega) : \begin{aligned} \dot{\boldsymbol{x}} &= (\boldsymbol{A}(\boldsymbol{\rho}) + \boldsymbol{B}(\boldsymbol{\rho})\boldsymbol{K}(\boldsymbol{\rho}))\boldsymbol{x} + \boldsymbol{B}_w\boldsymbol{w} \\ \boldsymbol{z} &= (\boldsymbol{C}_z(\boldsymbol{\rho}) + \boldsymbol{D}_z(\boldsymbol{\rho})\boldsymbol{K}(\boldsymbol{\rho}))\boldsymbol{x} \end{aligned} \tag{2.45}$$

which is similar to (2.23).

We firstly apply the Theorem 9 and get the following result.

**Theorem 13.** *Consider the LPV controlled system $\hat{\boldsymbol{G}}_c(j\omega)$ presented in* (2.44)*. if two parameter-independent matrices $\boldsymbol{X} = \boldsymbol{X}^T > 0$ and $\boldsymbol{W}$, and a parameter-dependent matrix $\boldsymbol{Z}(\boldsymbol{\rho})$ are solu-*

*tions to the optimization problem*

$$\min_{\boldsymbol{X},\boldsymbol{Z}(\boldsymbol{\rho}),\boldsymbol{W}} \quad \mathrm{tr}(\boldsymbol{W}) \tag{2.46}$$

$$\textit{subject to} \quad \boldsymbol{A}(\boldsymbol{\rho})\boldsymbol{X} + \boldsymbol{B}_u(\boldsymbol{\rho})\boldsymbol{Z}(\boldsymbol{\rho}) + (\bullet)^T + \boldsymbol{B}_w(\boldsymbol{\rho})\boldsymbol{B}_w(\boldsymbol{\rho})^T < 0, \tag{2.47}$$

$$\begin{bmatrix} \boldsymbol{W} & \boldsymbol{C}_z(\boldsymbol{\rho})\boldsymbol{X} + \boldsymbol{D}_u(\boldsymbol{\rho})\boldsymbol{Z}(\boldsymbol{\rho}) \\ (\bullet)^T & \boldsymbol{X} \end{bmatrix} > 0, \tag{2.48}$$

$$\boldsymbol{\rho}(t) \in \mathcal{D}_{\boldsymbol{\rho}},$$

*the the control $\boldsymbol{u} = \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{x}$ with $\boldsymbol{K}(\boldsymbol{\rho}) = \boldsymbol{Z}(\boldsymbol{\rho})\boldsymbol{X}^{-1}$ stabilize the system and the $H_2$ norm is minimized as $\|\hat{\boldsymbol{G}}_c(j\omega)\|_2 < \sqrt{\mathrm{tr}(\boldsymbol{W})}$.*

*Proof.* With full-state feedback control law, we consider (2.45) and substitute the close-loop system into (2.26) (2.27) with $\boldsymbol{Z}(\boldsymbol{\rho}) = \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{X}$, then (2.47) and (2.48) follow. Using the Theorem 9 one can conclude the $H_2$ norm is bounded and minimized by $\mathrm{tr}(W)$. ■

**Remark.** *As the discussion of the Theorem 9, the control gain guarantees the quadratic stability so it is very conservative. Further, the minimization problem with a constant matrix $\boldsymbol{W}$, which bounds the $H_2$ norm for all $\boldsymbol{\rho}(t) \in \mathcal{D}_{\boldsymbol{\rho}}$, limits the performance of the control system.*

In order to improve the performance and reduce the conservatism, $\boldsymbol{X}$ and $\boldsymbol{W}$ in the Theorem 13 are changed to parameter-dependent matrices, and the following result is derived.

**Theorem 14.** *Consider the LPV controlled system $\hat{\boldsymbol{G}}_c(j\omega)$ presented in* (2.44). *if three parameter-dependent matrices $\boldsymbol{X}(\boldsymbol{\rho}) = \boldsymbol{X}(\boldsymbol{\rho})^T > 0$, $\boldsymbol{W}(\boldsymbol{\rho})$, $\boldsymbol{Z}(\boldsymbol{\rho})$ and $\boldsymbol{Y}$ are solution to the optimization* *problem*

$$\underset{\boldsymbol{X},\boldsymbol{Z},\boldsymbol{W},Y}{\mathit{min}} \quad \operatorname{tr}(\boldsymbol{Y}) \tag{2.49}$$

$$\textit{subject to} \quad \boldsymbol{A}(\boldsymbol{\rho})\boldsymbol{X}(\boldsymbol{\rho}) + \boldsymbol{B}_u(\boldsymbol{\rho})\boldsymbol{Z}(\boldsymbol{\rho}) + (\bullet)^T + \boldsymbol{B}_w(\boldsymbol{\rho})\boldsymbol{B}_w(\boldsymbol{\rho})^T - \sum_{i=1}^{n_{\boldsymbol{\rho}}} \frac{\partial \boldsymbol{X}(\boldsymbol{\rho})}{\partial \rho_i}\dot{\rho}_i < 0, \tag{2.50}$$

$$\begin{bmatrix} \boldsymbol{W}(\boldsymbol{\rho}) & \boldsymbol{C}_z(\boldsymbol{\rho})\boldsymbol{X}(\boldsymbol{\rho}) + \boldsymbol{D}_u(\boldsymbol{\rho})\boldsymbol{Z}(\boldsymbol{\rho}) \\ (\bullet)^T & \boldsymbol{X}(\boldsymbol{\rho}) \end{bmatrix} > 0, \tag{2.51}$$

$$\boldsymbol{W}(\boldsymbol{\rho}) - \boldsymbol{Y} \leq 0 \tag{2.52}$$

$$\boldsymbol{\rho}(t) \in \mathcal{D}_{\boldsymbol{\rho}},$$

$$\dot{\boldsymbol{\rho}} \in \mathbb{D}_{\dot{\boldsymbol{\rho}}},$$

*where $\mathbb{D}_{\dot{\boldsymbol{\rho}}}$ is the vertices of the set $\mathcal{D}_{\dot{\boldsymbol{\rho}}}$, then the control $\boldsymbol{u} = \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{x}$ with $\boldsymbol{K}(\boldsymbol{\rho}) = \boldsymbol{Z}(\boldsymbol{\rho})\boldsymbol{X}^{-1}(\boldsymbol{\rho})$ stabilize the system and the $H_2$ norm is minimized as*

$$\|\hat{\boldsymbol{G}}_c(j\omega)\|_2 < \sqrt{\lim_{T\to\infty} \frac{1}{T}\int_0^T \operatorname{tr}(\boldsymbol{W}(\boldsymbol{\rho}(t)))} \leq \sqrt{\operatorname{tr}(\boldsymbol{Y})}. \tag{2.53}$$

*Proof.* Similar with the proof of Theorem 13 but instead of using the Theorem 9, here we utilize the Theorem 10, where $\dot{\boldsymbol{X}}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) = \sum_{i=1}^{n_\rho} \frac{\partial \boldsymbol{X}(\boldsymbol{\rho})}{\partial \rho_i}\dot{\rho}_i$. Since it is affine in $\dot{\boldsymbol{\rho}}$, only the vertices of $\mathcal{D}_{\dot{\boldsymbol{\rho}}}$ is needed, i.e. $\mathbb{D}_{\dot{\boldsymbol{\rho}}}$.

Further, $\boldsymbol{W}(\boldsymbol{\rho}) - \boldsymbol{Y} \leq 0$ implies $\mathbf{tr}\ \boldsymbol{W}(\boldsymbol{\rho})) - \mathbf{tr}\ (\boldsymbol{Y}) \leq 0$, so we can conclude that the control gain $\boldsymbol{K}(\boldsymbol{\rho})$ stabilizes the system and the $H_2$ norm is bounded and minimized by (2.53). ■

**Remark.** *The controller synthesized from the Theorem 14 stabilizes the LPV system with less conservatism, where the parameter-dependent matrices $\boldsymbol{X}(\boldsymbol{\rho})$ and $\boldsymbol{Z}(\boldsymbol{\rho})$ involves larger area of feasible set than in the Theorem 13. Besides, the parameter-dependent matrix $\boldsymbol{W}(\boldsymbol{\rho})$ is more adaptive to the system so that intuitively the performance of the system should be better than the one with constant $\boldsymbol{W}$. However, parameter-dependent conditions require more computational effort on the solution process since the dimension increase as the size of the system goes up. This* *is a challenge in the thesis and we will address this issue later.*

# 3. LPV QUADRATIC REGULATOR IN POLYNOMIAL CHAOS FRAMEWORK

In this chapter, we present a new theoretical framework for designing linear parameter varying controllers in the polynomial chaos framework[1]. We assume the scheduling variable to be random and apply the polynomial chaos approach to synthesize the controller for the resulting linear stochastic dynamical system. Two algorithms are presented that minimize the performance objective with respect to the stochastic system. The first algorithm is based on the generalized polynomial expansion and the second algorithm is based on the stochastic collocation.

The same as Section 2.5 of Chapter 2, the LPV system is expressed as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{\rho})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\rho})\boldsymbol{u}, \tag{3.1}$$

where system matrices depend on unknown parameter $\boldsymbol{\rho}(t)$, which is measurable in real-time [53, 54]. We assume the parameter $\boldsymbol{\rho}(t)$ is bounded in magnitude and rate, i.e. $\boldsymbol{\rho}(t) \in \mathcal{D}_{\boldsymbol{\rho}} \subset \mathbb{R}^{n_\rho}$ and $\dot{\boldsymbol{\rho}}(t) \in \mathcal{D}_{\dot{\boldsymbol{\rho}}} \subset \mathbb{R}^{n_\rho}$. A key aspect of the formulation presented in this chapter is that $\boldsymbol{\rho}$ and $\dot{\boldsymbol{\rho}}$ are treated as uncertainties in the LPV control law synthesis. This is similar to the idea proposed by Packard [25], where $\boldsymbol{\rho}(t)$ is treated as a bounded uncertainty. The approach taken uses the optimally scaled small-gain theorem, and solves the control synthesis problem by reformulating the existence conditions into a finite-dimensional convex optimization using linear fractional transformations. The idea of treating $\boldsymbol{\rho}(t)$ as an uncertainty in the LPV design has also been introduced by Fujisaki et. al. [55], where $\boldsymbol{\rho}(t)$ is treated as probabilistic uncertainty. In their work, they addressed the computational complexity of such problems by presenting a probabilistic approach to solve these problems, via a sequential randomized algorithm, which significantly reduces the computational complexity. Here the parameter $\boldsymbol{\rho}(t)$, is treated as a random variable, with a distri-

[1] ©2017 IEEE. Part of this chapter is reprinted with permission from "Design of stochastic collocation based linear parameter varying quadratic regulator." by Shao-Chen Hsu and Raktim Bhattacharya, 2017 American Control Conference.

bution $f_{\boldsymbol{\rho}}(\boldsymbol{\rho})$ defined over $\mathcal{D}_{\boldsymbol{\rho}}$. The LPV synthesis problem is solved by sampling $\mathcal{D}_{\boldsymbol{\rho}}$ and solving the sampled LMIs using a sequential-gradient method. As with any probabilistic algorithm, there is a tradeoff between sample complexity and confidence in the solution. Often, a large number of samples are required to generate a solution with high confidence.

Inspired by [55], we treat $\boldsymbol{\rho}$ as a set of random-variables, with a given joint distribution. The LPV controller $\boldsymbol{K}(\boldsymbol{\rho})$ is determined such $\mathbf{E}\left[\dot{V}\right] < 0$ is guaranteed at every time instant. Since the condition is satisfied at every time instant, $\boldsymbol{\rho}$ can assume different values at different times, but as long as $\mathbf{E}\left[\dot{V}\right] < 0$ is satisfied, exponential mean-square stability is guaranteed [56]. The main contribution is a new LPV regulator synthesis algorithm, in the polynomial chaos framework, which generates a parameter dependent gain. In this framework, orthogonal basis functions are determined by polynomial chaos theory. The controller is optimal with respect to a quadratic cost in states and control. It is noted that this chapter extends the work in [57] where only stability was considered. We look at the LPV-LQR problem with application to a realistic autopilot missile control problem in the late chapter.

### 3.1 Polynomial Chaos Theory

Polynomial chaos (PC) is a *deterministic* method for the evolution of uncertainty in dynamical systems when there is probabilistic uncertainty in the system parameters. Polynomial chaos was first introduced by Wiener [58] where Hermite polynomials were used to model stochastic processes with Gaussian random variables. It can be thought of as an extension of Volterra's theory of nonlinear functionals for stochastic systems [59, 60]. According to Cameron and Martin [61] polynomial chaos expansion converges in the $\mathcal{L}_2$ sense for any arbitrary stochastic process with finite second moments. This applies to most physical systems. Xiu et. al. [2] generalized the result of Cameron-Martin to various continuous and discrete distributions using orthogonal polynomials from the so called Askey-scheme [62] and demonstrated $\mathcal{L}_2$ convergence in the corresponding Hilbert functional space. The PC framework has been applied to applications including stochastic fluid dynamics [63, 64, 65], stochastic finite elements [60], and solid mechanics [66, 67], feedback control [68, 69, 70, 71] and estimation [72]. It has been shown that PC based methods are computationally far superior to Monte-Carlo based methods [2, 63, 64, 65, 73]. See [74] for several benchmark problems solved by non-intrusive PC expansion and stochastic collocation.

Formally, the PC framework is described as follows. Let $(\Omega, \mathcal{F}, P)$ be a probability space, where $\Omega$ is the sample space, $\mathcal{F}$ is the $\sigma$-algebra of the subsets of $\Omega$, and $P$ is the probability measure. Let $\boldsymbol{\rho}(\omega) = (\boldsymbol{\rho}_1(\omega), \cdots, \boldsymbol{\rho}_d(\omega)) : (\Omega, \mathcal{F}) \rightarrow (\mathbb{R}^d, \mathcal{B}^d)$ be an $\mathbb{R}^d$-valued continuous random variable, where $d \in \mathbb{N}$, and $\mathcal{B}^d$ is the $\sigma$-algebra of Borel subsets of $\mathbb{R}^d$.

A general second order process $X(\omega) \in \mathcal{L}_2(\Omega, \mathcal{F}, P)$ can be expressed by polynomial chaos as

$$X(t, \omega) = \sum_{i=0}^{\infty} x_i(t) \phi_i(\boldsymbol{\rho}(\omega)), \tag{3.2}$$

where $\omega$ is the random event and $\phi_i(\boldsymbol{\rho}(\omega))$ denotes the polynomial chaos basis of degree $p$ in terms of the random variables $\boldsymbol{\rho}(\omega)$. In practice, the infinite series is truncated and $X(t, \omega)$ is approximated by

$$X(t, \omega) \approx \hat{X}(t, \omega) = \sum_{i=0}^{N} x_i(t) \phi_i(\boldsymbol{\rho}(\omega)),$$

and the truncated error can be defined as [75]

$$e_t = \|X(t, \omega) - \hat{X}(t, \omega)\|_{L_2},$$

which converges in the mean square sense and the convergence rate depends on the stochastic process and the chosen basis function [2].

| Random Variable $\boldsymbol{\rho}$ | $\phi_i(\boldsymbol{\rho})$ of the Wiener-Askey Scheme |
|---|---|
| Gaussian | Hermite |
| Uniform | Legendre |
| Gamma | Laguerre |
| Beta | Jacobi |

Table 3.1: Correspondence between choice of orthogonal polynomials and given distribution of $\boldsymbol{\rho}(\omega)$ [2].

The functions $\{\phi_i\}$ are a family of orthogonal basis in $\mathcal{L}_2(\Omega, \mathcal{F}, P)$ satisfying the relation

$$\mathbf{E}\left[\phi_i\phi_j\right] := \int_{\mathcal{D}_{\boldsymbol{\rho}}} \phi_i(\boldsymbol{\rho})\phi_j(\boldsymbol{\rho}) f_{\boldsymbol{\rho}}(\boldsymbol{\rho})\, d\boldsymbol{\rho} = \begin{cases} 0, & \text{if } i \neq j, \\ \int_{\mathcal{D}_{\boldsymbol{\rho}}} \phi_i^2(\boldsymbol{\rho}) f_{\boldsymbol{\rho}}(\boldsymbol{\rho})\, d\boldsymbol{\rho}, & \text{otherwise.} \end{cases} \tag{3.3}$$

where $\mathcal{D}_{\boldsymbol{\rho}}$ is the domain of the random variable $\boldsymbol{\rho}(\omega)$, and $f_{\boldsymbol{\rho}}(\boldsymbol{\rho})$ is a probability density function for $\boldsymbol{\rho}$. Table 3.1 shows the family of basis functions for random variables with common distributions.

Generally, there are three methods for expanding a random process in the stochastic framework – intrusive methods, e.g. Galerkin projection, non-intrusiv emethods, e.g. spectral projection, and stochastic collocation. In this chapter, we choose stochastic collocation method and it offers no mathematical advantage over other non-intrusive methods. It has been reported that the non-intrusive methods offer similar accuracy with stochastic collocation [76]. However, the comparison is implemented through simulations only without any theoratical proof. We will investigate the advantage of such methods in the LPV controller synthesis problem for our future work. We also assume components of $\boldsymbol{\rho}$ to be independent, which can be relaxed by utilizing new developments in polynomial chaos theory [77, 78].

### 3.1.1 Polynomial Chaos Expansion

With respect to the dynamical system defined as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_{cl}(\boldsymbol{\rho})\boldsymbol{x}, \tag{3.4}$$

the solution can be approximated by the polynomial chaos expansion as

$$\boldsymbol{x}(t,\boldsymbol{\rho}) \approx \hat{\boldsymbol{x}}(t,\boldsymbol{\rho}) = \sum_{i=0}^{N} \boldsymbol{x}_i(t)\phi_i(\boldsymbol{\rho}), \tag{3.5}$$

where the polynomial chaos coefficients $\boldsymbol{x}_i \in \mathbb{R}^n$. Define $\boldsymbol{\Phi}(\boldsymbol{\rho})$ to be

$$\boldsymbol{\Phi} \equiv \boldsymbol{\Phi}(\boldsymbol{\rho}) := \begin{pmatrix} \phi_0(\boldsymbol{\rho}) & \cdots & \phi_N(\boldsymbol{\rho}) \end{pmatrix}^T, \text{ and} \tag{3.6}$$

$$\boldsymbol{\Phi}_n \equiv \boldsymbol{\Phi}_n(\boldsymbol{\rho}) := \boldsymbol{\Phi}(\boldsymbol{\rho}) \otimes \boldsymbol{I}_n, \tag{3.7}$$

where $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ is an identity matrix. Also define matrix $\boldsymbol{X} \in \mathbb{R}^{n\times(N+1)}$, with polynomial chaos coefficients $\boldsymbol{x}_i$, as

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_0 & \cdots & \boldsymbol{x}_N \end{bmatrix}.$$

This lets us define $\hat{\boldsymbol{x}}(t,\boldsymbol{\rho})$ as

$$\hat{\boldsymbol{x}}(t,\boldsymbol{\rho}) := \boldsymbol{X}(t)\boldsymbol{\Phi}(\boldsymbol{\rho}). \tag{3.8}$$

Noting that $\hat{\boldsymbol{x}} \equiv \textbf{vec}\left(\hat{\boldsymbol{x}}\right)$, we obtain an alternate form for (3.8),

$$\hat{\boldsymbol{x}} \equiv \textbf{vec}\left(\hat{\boldsymbol{x}}\right) = \textbf{vec}\left(\boldsymbol{X}\boldsymbol{\Phi}\right) = \textbf{vec}\left(\boldsymbol{I}_n\boldsymbol{X}\boldsymbol{\Phi}\right) = (\boldsymbol{\Phi}^T \otimes \boldsymbol{I}_n)\textbf{vec}\left(\boldsymbol{X}\right) = \boldsymbol{\Phi}_n^T\boldsymbol{x}_{pc}, \tag{3.9}$$

where $\boldsymbol{x}_{pc} := \textbf{vec}\left(\boldsymbol{X}\right)$, and $\textbf{vec}\left(\cdot\right)$ is the vectorization operator [79]. This allows us to transform $\hat{\boldsymbol{x}}$ from a summation form to a matrix form, which reduces the complexity of the derivations presented later.

Since $\hat{\boldsymbol{x}}$ from (3.9) is an approximation, substituting it in (3.4), introduces a truncated error, i.e. $\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}_{cl}(\boldsymbol{\rho})\hat{\boldsymbol{x}} + \boldsymbol{e}$. We thus have the error $\boldsymbol{e}$, which is given by

$$\boldsymbol{e} = \dot{\hat{\boldsymbol{x}}} - \boldsymbol{A}_{cl}(\boldsymbol{\rho})\hat{\boldsymbol{x}} = \boldsymbol{\Phi}_n^T \dot{\boldsymbol{x}}_{pc} - \boldsymbol{A}_{cl}(\boldsymbol{\rho})\boldsymbol{\Phi}_n^T \boldsymbol{x}_{pc}. \tag{3.10}$$

The best $\mathcal{L}_2$ approximation is obtained by setting [2]

$$\langle \boldsymbol{e}\phi_i(\boldsymbol{\rho})\rangle := \mathsf{E}\left[\boldsymbol{e}\phi_i\right] = 0, \text{ for } i = 0, 1, \cdots, N. \tag{3.11}$$

Combining (3.10) and (3.11) we arrive

$$\mathsf{E}\left[\boldsymbol{\Phi}_n \boldsymbol{\Phi}_n^T\right] \dot{\boldsymbol{x}}_{pc} = \mathsf{E}\left[\boldsymbol{\Phi}_n \boldsymbol{A}_{cl} \boldsymbol{\Phi}_n^T\right] \boldsymbol{x}_{pc},$$

which yields

$$\dot{\boldsymbol{x}}_{pc} = \boldsymbol{A}_{pc}\boldsymbol{x}_{pc}, \tag{3.12}$$

where $\boldsymbol{A}_{pc} = \mathsf{E}\left[\boldsymbol{\Phi}_n \boldsymbol{\Phi}_n^T\right]^{-1} \mathsf{E}\left[\boldsymbol{\Phi}_n \boldsymbol{A}_{cl} \boldsymbol{\Phi}_n^T\right]$, $\boldsymbol{\Phi}_n$ and $\boldsymbol{A}_{cl}$ depend on $\boldsymbol{\rho}$ as defined earlier. Equation (3.12), is the best finite dimensional approximation of (3.4) in the $\mathcal{L}_2$ sense. It is noted that the methodology to obtain the coefficients in this section is called Galerkin projection.

### 3.1.2 Stochastic Collocation

The concept of this approach is to pick a set of specific sample points from polynomial chaos basis functions and enforce the error of approximation at the sample points to be zero. In this approach we first introduce Lagrange interpolating polynomials

$$l_i(\boldsymbol{\rho}) = \prod_{j=0, j\neq i}^{N} \frac{\boldsymbol{\rho} - \boldsymbol{\rho}_j}{\boldsymbol{\rho}_i - \boldsymbol{\rho}_j} \tag{3.13}$$

as basis functions, where $\boldsymbol{\rho}_i$ are the roots of the polynomial chaos basis of degree $N+1$. The Lagrange polynomials have an important property, i.e.

$$l_i(\boldsymbol{\rho}_k) = \begin{cases} 0 & k \neq i \\ 1 & k = i \end{cases},$$

and the polynomials are orthogonal to each other in the $\mathcal{L}_2$ sense, which can be proved using the Gaussian quadrature rule [80] as follows.

$$\begin{aligned} \mathbf{E}\left[l_i l_j\right] &= \int_{\mathcal{D}_{\boldsymbol{\rho}}} l_i(\boldsymbol{\rho}) l_j(\boldsymbol{\rho}) f_{\boldsymbol{\rho}}(\boldsymbol{\rho}) d\boldsymbol{\rho} \\ &= \sum_{k=0}^{N} \left[ l_i(\boldsymbol{\rho}_k) l_j(\boldsymbol{\rho}_k) \int_{\mathcal{D}_{\boldsymbol{\rho}}} l_k(\boldsymbol{\rho}) f_{\boldsymbol{\rho}}(\boldsymbol{\rho}) d\boldsymbol{\rho} \right]. \end{aligned}$$

Since $i \neq j$, we can conclude

$$\mathbf{E}\left[l_i l_j\right] = 0.$$

The solution to the dynamical system (3.4) thus can also be approximated as

$$\boldsymbol{x}(t, \boldsymbol{\rho}) \approx \tilde{\boldsymbol{x}}(t, \boldsymbol{\rho}) = \sum_{i=0}^{N} \boldsymbol{x}_{i,sc}(t) l_i(\boldsymbol{\rho}) = \boldsymbol{L}_n^T \boldsymbol{x}_{sc},$$

where $\boldsymbol{L}_n = \left[l_0 \dots l_N\right]^T \otimes \boldsymbol{I}_n$, and $\boldsymbol{x}_{sc} = \left[\boldsymbol{x}_{0,sc} \dots \boldsymbol{x}_{N,sc}\right]^T$ are coefficients determined by solving $\dot{\boldsymbol{x}}_{i,sc} = \boldsymbol{A}_{cl}(\boldsymbol{\rho}_i)\boldsymbol{x}_{i,sc}$. It implies that the solution $\tilde{\boldsymbol{x}}$ is exact at those specified sample points, which means that the error $\tilde{\boldsymbol{e}} = \dot{\tilde{\boldsymbol{x}}} - \boldsymbol{A}_{cl}(\boldsymbol{\rho})\tilde{\boldsymbol{x}}$ is forced to be zero at the sample points $\boldsymbol{\rho}_i$.

For a special case where $\boldsymbol{A}_{cl}$ is an univariate function of $\boldsymbol{\rho} \in \mathbb{R}$, we have the following observation.

**Theorem 15.** *Consider a linear system in* (3.4)*, where* $d = 1$ *and* $\boldsymbol{A}_{cl}$ *is linear in* $\boldsymbol{\rho}$*, the approximated solution* $\hat{\boldsymbol{x}}(t, \boldsymbol{\rho})$ *obtained by Galerkin projection is equivalent to the one* $\tilde{\boldsymbol{x}}(t, \boldsymbol{\rho})$ *obtained by stochastic collocation in* $L_2$ *sense.*

*Proof.* It is known that the stochastic collocation approximation is

$$\boldsymbol{x}(t, \boldsymbol{\rho}) \approx \tilde{\boldsymbol{x}}(t, \boldsymbol{\rho}) = \boldsymbol{L}_n^T \boldsymbol{x}_{sc}(t), \tag{3.14}$$

and

$$\begin{aligned} \tilde{e} &= \dot{\tilde{\boldsymbol{x}}} - \boldsymbol{A}_{cl}\tilde{\boldsymbol{x}} \\ &= \boldsymbol{L}_n^T \dot{\boldsymbol{x}}_{sc} - \boldsymbol{A}_{cl}\boldsymbol{L}_n^T \boldsymbol{x}_{sc} \\ &= \boldsymbol{L}_n^T \boldsymbol{A}_{sc}\boldsymbol{x}_{sc} - A_{cl}\boldsymbol{L}_n^T \boldsymbol{x}_{sc} \\ &= (\boldsymbol{L}_n^T \boldsymbol{A}_{sc} - \boldsymbol{A}_{cl}\boldsymbol{L}_n^T)\boldsymbol{x}_{sc}, \end{aligned}$$

where

$$\boldsymbol{A}_{sc} = \begin{bmatrix} \boldsymbol{A}_{cl}(\boldsymbol{\rho}_0) & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{A}_{cl}(\boldsymbol{\rho}_N) \end{bmatrix}.$$

To prove the equivalence, we need to show $\tilde{\boldsymbol{x}}$ is also the best $\mathcal{L}_2$ approximation, i.e. $\mathsf{E}\left[\boldsymbol{L}_n \tilde{e}\right] = 0$.

$$\begin{aligned} \mathsf{E}\left[\boldsymbol{L}_n \tilde{e}\right] &= \mathsf{E}\left[(\boldsymbol{L}_n \boldsymbol{L}_n^T \boldsymbol{A}_{sc} - \boldsymbol{L}_n \boldsymbol{A}_{cl}\boldsymbol{L}_n^T)\boldsymbol{x}_{sc}\right] \\ &= \left[\mathsf{E}\left[\boldsymbol{L}_n \boldsymbol{L}_n^T\right] \boldsymbol{A}_{sc} - \mathsf{E}\left[\boldsymbol{L}_n \boldsymbol{A}_{cl}\boldsymbol{L}_n^T\right]\right] \boldsymbol{x}_{sc}. \end{aligned} \tag{3.15}$$

Because of the orthogonality of the basis, the first term of (3.15) can be expanded as

$$
\begin{aligned}
&\mathsf{E}\left[\boldsymbol{L}_n\boldsymbol{L}_n^T\right]\boldsymbol{A}_{sc}\\
=&\mathsf{E}\left[\begin{bmatrix} l_{0n}l_{0n} & l_{0n}l_{1n} & \cdots & l_{0n}l_{Nn}\\ l_{1n}l_{0n} & l_{1n}l_{1n} & \cdots & l_{1n}l_{Nn}\\ \vdots & & \ddots & \\ l_{Nn}l_{0n} & l_{Nn}l_{1n} & \cdots & l_{Nn}l_{Nn}\end{bmatrix}\right]\boldsymbol{A}_{sc}\\
=&\mathsf{E}\left[\begin{bmatrix} l_{0n}^2 & & \mathbf{0}\\ & \ddots & \\ \mathbf{0} & & l_{Nn}^2\end{bmatrix}\right]\boldsymbol{A}_{sc},
\end{aligned}
$$

where $l_{in} = l_i \otimes \boldsymbol{I}_n$.

By the Gaussian quadrature rule,

$$
\begin{aligned}
\mathsf{E}\left[l_i^2\right] &= \int_{\mathcal{D}_{\boldsymbol{\rho}}} l_i(\boldsymbol{\rho})^2 f_{\boldsymbol{\rho}}(\boldsymbol{\rho})d\boldsymbol{\rho}\\
&= \sum_{j=0}^{N} l_i^2(\boldsymbol{\rho}_j)\int_{\mathcal{D}_{\boldsymbol{\rho}}} l_j(\boldsymbol{\rho})f_{\boldsymbol{\rho}}(\boldsymbol{\rho})d\boldsymbol{\rho}\\
&= \int_{\mathcal{D}_{\boldsymbol{\rho}}} l_i(\boldsymbol{\rho})f_{\boldsymbol{\rho}}(\boldsymbol{\rho})d\boldsymbol{\rho}\\
&= \mathsf{E}\left[l_i\right].
\end{aligned}
$$

Therefore, $\mathsf{E}\left[l_{in}^2\right] = \mathsf{E}\left[l_{in}\right]$ and it implies

$$
\mathsf{E}\left[\boldsymbol{L}_n\boldsymbol{L}_n^T\right]\boldsymbol{A}_{sc} = \mathsf{E}\left[\begin{bmatrix} l_{0n} & & \mathbf{0}\\ & \ddots & \\ \mathbf{0} & & l_{Nn}\end{bmatrix}\right]\boldsymbol{A}_{sc} = \mathsf{E}\left[\begin{bmatrix} l_{0n}\boldsymbol{A}(\boldsymbol{\rho}_0) & & \mathbf{0}\\ & \ddots & \\ \mathbf{0} & & l_{Nn}\boldsymbol{A}(\boldsymbol{\rho}_N)\end{bmatrix}\right]. \qquad (3.16)
$$

For the second term of (3.15), it can be expanded as

$$
\begin{aligned}
\mathbf{E}\left[\boldsymbol{L}_n\boldsymbol{A}_{cl}\boldsymbol{L}_n^T\right] &= \mathbf{E}\left[\begin{bmatrix} l_{on} \\ \vdots \\ l_{Nn} \end{bmatrix} \boldsymbol{A}_{cl} \begin{bmatrix} l_{on} \cdots l_{Nn} \end{bmatrix}\right] \\
&= \mathbf{E}\left[\begin{bmatrix} l_{0n}\boldsymbol{A}_{cl}l_{0n} & l_{0n}\boldsymbol{A}_{cl}l_{1n} & \cdots & l_{0n}\boldsymbol{A}_{cl}l_{Nn} \\ l_{1n}\boldsymbol{A}_{cl}l_{0n} & l_{1n}\boldsymbol{A}_{cl}l_{1n} & \cdots & l_{1n}\boldsymbol{A}_{cl}l_{Nn} \\ \vdots & & \ddots & \\ l_{Nn}\boldsymbol{A}l_{0n} & l_{Nn}\boldsymbol{A}_{cl}l_{1n} & \cdots & l_{Nn}\boldsymbol{A}_{cl}l_{Nn} \end{bmatrix}\right] \\
&= \mathbf{E}\left[\begin{bmatrix} l_{0n}l_{0n}\boldsymbol{A}_{cl} & l_{0n}l_{1n}\boldsymbol{A} & \cdots & l_{0n}l_{Nn}\boldsymbol{A}_{cl} \\ l_{1n}l_{0n}\boldsymbol{A}_{cl} & l_{1n}l_{1n}\boldsymbol{A} & \cdots & l_{1n}l_{Nn}\boldsymbol{A}_{cl} \\ \vdots & & \ddots & \\ l_{Nn}l_{0n}\boldsymbol{A}_{cl} & l_{Nn}l_{1n}\boldsymbol{A}_{cl} & \cdots & l_{Nn}l_{Nn}\boldsymbol{A}_{cl} \end{bmatrix}\right],
\end{aligned}
\qquad (3.17)
$$

where

$$
\mathbf{E}\left[l_{in}l_{jn}\boldsymbol{A}_{cl}\right] = \mathbf{E}\left[\begin{bmatrix} l_il_j & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & l_il_j \end{bmatrix} \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix}\right].
$$

Since we can show that

$$\begin{aligned}\mathsf{E}\left[l_i l_j A_{kl}\right] &= \int_{\mathcal{D}_{\boldsymbol{\rho}}} l_i l_j A_{kl} f_{\boldsymbol{\rho}}(\boldsymbol{\rho}) d\boldsymbol{\rho} \\ &= \sum_{m=0}^{N} l_i(\boldsymbol{\rho}_m) l_j(\boldsymbol{\rho}_m) A_{kl}(\boldsymbol{\rho}_m) \int_{\mathcal{D}_{\boldsymbol{\rho}}} l_m(\boldsymbol{\rho}) f_{\boldsymbol{\rho}}(\boldsymbol{\rho}) d\boldsymbol{\rho} \\ &= \begin{cases} 0 & i \neq j \\ A_{kl}(\boldsymbol{\rho}_i) \int_{\mathcal{D}_{\boldsymbol{\rho}}} l_i(\boldsymbol{\rho}) f_{\boldsymbol{\rho}}(\boldsymbol{\rho}) d\boldsymbol{\rho} & i = j \end{cases} \\ &= \begin{cases} 0 & i \neq j \\ \mathsf{E}\left[l_i\right] A_{kl}(\boldsymbol{\rho}_i) & i = j \end{cases}, \end{aligned} \tag{3.18}$$

it implies that

$$\begin{aligned}\mathsf{E}\left[l_{in} l_{jn} \boldsymbol{A}_{cl}\right] &= \begin{cases} 0 & i \neq j \\ \mathsf{E}\left[\begin{bmatrix} l_i & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & l_i \end{bmatrix}\right] \boldsymbol{A}_{cl}(\boldsymbol{\rho}_i) & i = j \end{cases} \\ &= \begin{cases} 0 & i \neq j \\ \mathsf{E}\left[l_{in}\right] \boldsymbol{A}_{cl}(\boldsymbol{\rho}_i) & i = j \end{cases}. \end{aligned}$$

(3.17) thus becomes

$$\mathsf{E}\left[\boldsymbol{L}_n \boldsymbol{A}_{cl} \boldsymbol{L}_n^T\right] = \mathsf{E}\left[\begin{bmatrix} l_{0n}\boldsymbol{A}_{cl}(\boldsymbol{\rho}_0) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & l_{Nn}\boldsymbol{A}_{cl}(\boldsymbol{\rho}_N) \end{bmatrix}\right]. \tag{3.19}$$

Substituting (3.16) and (3.19) into (3.15), we can conclude $\mathsf{E}\left[\boldsymbol{L}_n \tilde{e}\right] = 0$, which means that $\tilde{\boldsymbol{x}}$ is also the best $L_2$ approximation as $\hat{\boldsymbol{x}}$. Therefore, $\tilde{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}$ are identical. ■

**Remark.** *The theorem is an independent observation. It is noted that the equivalence only exists in a linear system and the system is linear in univariate random variable. We can justify this by following lemma.*

**Lemma 16.** *Consider two Lagrange polynomials $l_i(\boldsymbol{\rho})$ and $l_j(\boldsymbol{\rho})$ , and a scalar function $g(\boldsymbol{\rho})$, where $\boldsymbol{\rho} \in \mathbb{R}$ then*

$$\mathbf{E}\left[l_i(\boldsymbol{\rho})l_j(\boldsymbol{\rho})g(\boldsymbol{\rho})\right] \approx \begin{cases} 0, & i \neq j \\ \mathbf{E}\left[l_i\right] g(\boldsymbol{\rho}_i), & i = j \end{cases}, \tag{3.20}$$

*which is equivalent if $g(\boldsymbol{\rho})$ is a first order polynomial.*

*Proof.*

$$\begin{aligned} \mathbf{E}\left[l_i(\boldsymbol{\rho})l_j(\boldsymbol{\rho})g(\boldsymbol{\rho})\right] &= \int_{\mathcal{D}_{\boldsymbol{\rho}}} l_i(\boldsymbol{\rho})l_j(\boldsymbol{\rho})g(\boldsymbol{\rho})f_{\boldsymbol{\rho}}(\boldsymbol{\rho})d\boldsymbol{\rho} \\ &\approx \sum_{m=0}^{N} l_i(\boldsymbol{\rho}_m)l_j(\boldsymbol{\rho}_m)g(\boldsymbol{\rho}_m)\int_{\mathcal{D}_{\boldsymbol{\rho}}} l_m(\boldsymbol{\rho})f_{\boldsymbol{\rho}}(\boldsymbol{\rho})d\boldsymbol{\rho} \\ &= \begin{cases} 0 & i \neq j \\ g(\boldsymbol{\rho}_i)\int_{\mathcal{D}_{\boldsymbol{\rho}}} l_i(\boldsymbol{\rho})f_{\boldsymbol{\rho}}(\boldsymbol{\rho})d\boldsymbol{\rho} & i = j \end{cases} \\ &= \begin{cases} 0 & i \neq j \\ \mathbf{E}\left[l_i\right] g(\boldsymbol{\rho}_i) & i = j \end{cases}, \end{aligned}$$

According to the Gaussian quadrature rule, the expression is exact when $l_i(\boldsymbol{\rho})l_j(\boldsymbol{\rho})g(\boldsymbol{\rho})$ is a polynomial of degree at most $2N + 1$. It is known that the Lagrange interpolations are N-th order polynomials, so we can conclude that if $g$ is a first order polynomial (3.20) is exact. ■

Using Lemma 16, we can ensure that (3.18) must be exact since $\boldsymbol{A}(\boldsymbol{\rho})$ is linear in single random variable; otherwise, it becomes an approximation and two solutions are not identical. Therefore, for a linear stochastic system with single variable, one could choose either approximation for controller parameterizations and arrive at the same result.

## 3.2 Optimal Controller Synthesis

Recall the LQR derivations in Section 2.5, where a parameter-dependent convex optimization is formulated. However, in this chaper, we treat $\boldsymbol{\rho}$ as a vector of random variables and satisfy (2.35) in the mean-sense, i.e.

$$\mathbf{E}\left[\frac{dV}{dt}\right] \leq -\mathbf{E}\left[\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}\right], \tag{3.21}$$

which upper-bounds

$$\mathbf{E}\left[\int_0^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})\,dt\right], \tag{3.22}$$

i.e.

$$\mathbf{E}\left[V(\boldsymbol{x}(0))\right] \geq \mathbf{E}\left[\int_0^\infty \left(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}\right)dt\right]. \tag{3.23}$$

Satisfying (3.21) for all times, ensures the Lyapunov function decreases in the mean-sense, and consequently guarantees stability in the exponentially mean-square sense [81].

Although we assume $\mathcal{D}_{\boldsymbol{\rho}}$ and $\mathcal{D}_{\dot{\boldsymbol{\rho}}}$ to be compact, which is satisfied by all realistic systems, the framework presented here allows $\boldsymbol{\rho}$ and $\dot{\boldsymbol{\rho}}$ to be unbounded that can be for example modeled as Gaussian uncertainty. For such systems, we can guarantee exponential mean-square stability in this framework, which is not possible in the worst-case formulation.

Here we use polynomial chaos theory to determine the expectation operator in (3.21). We apply both polynomial chaos expansion and stochastic collocation techniques, and derive control synthesis problem in the respective frameworks, for the system considered in (3.1). We will see later, the polynomial chaos is more accurate than stochastic collocation technique, but results in more complex synthesis problems with increased computational time.

Before we proceed, we need the following result in the rest of the chapter.

**Proposition 17.** *For any vector* $\boldsymbol{v} \in \mathbb{R}^{N+1}$ *and matrix* $\boldsymbol{M} \in \mathbb{R}^{m \times n}$

$$\boldsymbol{M}(\boldsymbol{v}^T \otimes \boldsymbol{I}_n) = (\boldsymbol{v}^T \otimes \boldsymbol{I}_m)(\boldsymbol{I}_{N+1} \otimes \boldsymbol{M}), \tag{3.24}$$

*where* $\boldsymbol{I}_*$ *is identity matrix with indicated dimension.*

*Proof.*

$$\begin{aligned}\boldsymbol{M}(\boldsymbol{v}^T \otimes \boldsymbol{I}_n) &= (1 \otimes \boldsymbol{M})(\boldsymbol{v}^T \otimes \boldsymbol{I}_n)\\ &= \boldsymbol{v}^T \otimes \boldsymbol{M} = (\boldsymbol{v}^T \boldsymbol{I}_{N+1}) \otimes (\boldsymbol{I}_m \boldsymbol{M})\\ &= (\boldsymbol{v}^T \otimes \boldsymbol{I}_m)(\boldsymbol{I}_{N+1} \otimes \boldsymbol{M}).\end{aligned}$$

■

### 3.2.1 Galerkin Projection Based Formulation

Here we present the control synthesis formulation in the polynomial chaos framework, where the expectation is computed using exact integration of the corresponding function. Combining the idea with sufficient condition (3.21) yields the following result.

The following theorem poises the problem in terms of matrix variables $\boldsymbol{P}(\boldsymbol{\rho})$ and $\boldsymbol{K}(\boldsymbol{\rho})$ as arbitrary function of $\boldsymbol{\rho}$. Later we present a result that solves the problem in the polynomial chaos framework.

**Theorem 18.** *Controller gain* $\boldsymbol{K}(\boldsymbol{\rho}) := \boldsymbol{W}(\boldsymbol{\rho})\boldsymbol{Y}^{-1}(\boldsymbol{\rho})$ *minimizes* (3.22) *if* $\exists\, \boldsymbol{Y}(\boldsymbol{\rho}) = \boldsymbol{Y}^T(\boldsymbol{\rho}) >$

$0 \in \mathbb{R}^{n\times n}$ *and* $\boldsymbol{W}(\boldsymbol{\rho}) \in \mathbb{R}^{m\times n}$*, which are the solutions of the optimization problem*

$$\max \mathbf{tr}\ \mathsf{E}\left[\boldsymbol{Y}(\boldsymbol{\rho})\right] \tag{3.25}$$

*subject to*

$$\mathsf{E}\left[\mathsf{sym}\left(\mathcal{Y}\boldsymbol{\Phi}_n\boldsymbol{A}^T\boldsymbol{\Phi}_n^T + \mathcal{W}^T\boldsymbol{\Phi}_m\boldsymbol{B}^T\boldsymbol{\Phi}_n^T\right) + \mathcal{Y}\boldsymbol{\Phi}_n\boldsymbol{Q}\boldsymbol{\Phi}_n^T\mathcal{Y} + \mathcal{W}^T\boldsymbol{\Phi}_m\boldsymbol{R}\boldsymbol{\Phi}_m^T\mathcal{W}\right] + \mathcal{X}\boldsymbol{\Phi}_n\boldsymbol{\Phi}_n^T \leq 0, \tag{3.26}$$

$$\dot{\boldsymbol{\rho}} \in \mathbb{D}_{\dot{\boldsymbol{\rho}}} \tag{3.27}$$

*where* $\mathbb{D}_{\dot{\boldsymbol{\rho}}}$ *is the vertices of the set* $\mathcal{D}_{\dot{\boldsymbol{\rho}}}$*,* $\mathcal{Y} := \boldsymbol{I}_{N+1} \otimes \boldsymbol{Y}(\boldsymbol{\rho})$*,* $\mathcal{X} := \boldsymbol{I}_{N+1} \otimes \boldsymbol{X}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}})$*,* $\mathcal{W} := \boldsymbol{I}_{N+1} \otimes \boldsymbol{W}(\boldsymbol{\rho})$*, and* $\mathsf{sym}\left(\cdot\right) := (\cdot) + (\cdot)^T$*. The matrix functions* $\boldsymbol{Y}(\boldsymbol{\rho})$ *and* $\boldsymbol{X}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}})$ *are related by*

$$\boldsymbol{X}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) := -\sum_{i=1}^{n_{\boldsymbol{\rho}}} \mathsf{E}\left[\frac{\partial \boldsymbol{Y}(\boldsymbol{\rho})}{\partial \rho_i}\right]\dot{\rho}_i.$$

*Proof.* We proceed by formulating an optimization problem that minimizes the upper-bound on the cost-to-go. In the following derivation, we simplify the notation by not showing explicit dependence on $\boldsymbol{\rho}$ at every step of the derivation.

Let $V(\boldsymbol{x}) := \boldsymbol{x}^T\boldsymbol{P}(\boldsymbol{\rho})\boldsymbol{x}, \boldsymbol{P}(\boldsymbol{\rho}) = \boldsymbol{P}^T(\boldsymbol{\rho}) > 0 \in \mathbb{R}^{n\times n}$. Therefore, (3.21) can be written as

$$\mathsf{E}\left[\boldsymbol{x}^T\left(\mathsf{sym}\left((\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K})^T\boldsymbol{P}\right) + \boldsymbol{S}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) + \boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K}\right)\boldsymbol{x}\right] \leq 0,$$

or

$$\mathsf{E}\left[\boldsymbol{x}^T\left(\mathsf{sym}\left(\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{K}^T\boldsymbol{B}^T\boldsymbol{P}\right) + \boldsymbol{S}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) + \boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K}\right)\boldsymbol{x}\right] \leq 0, \tag{3.28}$$

where

$$\boldsymbol{S}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) := \sum_{i=1}^{n_\rho} \frac{\partial \boldsymbol{P}(\boldsymbol{\rho})}{\partial \rho_i} \dot{\rho}_i. \tag{3.29}$$

The above condition is not convex in $\boldsymbol{P}$ and $\boldsymbol{K}$. We can convexify the constraint using standard substitution [52], $\boldsymbol{P} := \boldsymbol{Y}^{-1}$ and $\boldsymbol{x} := \boldsymbol{Y}\boldsymbol{z}$, in the above quadratic form, to get

$$\begin{aligned}
&\boldsymbol{x}^T \left(\mathsf{sym}\left(\boldsymbol{PA} + \boldsymbol{PBK}\right) + \boldsymbol{S} + \boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{RK}\right)\boldsymbol{x} \\
&\quad = \boldsymbol{z}^T\boldsymbol{Y}\left(\mathsf{sym}\left(\boldsymbol{Y}^{-1}\boldsymbol{A} + \boldsymbol{Y}^{-1}\boldsymbol{BK}\right) + \boldsymbol{S} + \boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{RK}\right)\boldsymbol{Y}\boldsymbol{z} \\
&\quad = \boldsymbol{z}^T\left(\mathsf{sym}\left(\boldsymbol{AY} + \boldsymbol{BKY}\right) + \boldsymbol{YSY} + \boldsymbol{YQY} + \boldsymbol{YK}^T\boldsymbol{RKY}\right)\boldsymbol{z} \\
&\quad = \boldsymbol{z}^T\left(\mathsf{sym}(\boldsymbol{AY} + \boldsymbol{BW}) + \boldsymbol{YSY} + \boldsymbol{YQY} + \boldsymbol{W}^T\boldsymbol{RW}\right)\boldsymbol{z}
\end{aligned}$$

with $\boldsymbol{W} := \boldsymbol{KY}$.

Substituting $\boldsymbol{P} := \boldsymbol{Y}^{-1}$ in the definition of $\boldsymbol{S}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}})$ we get,

$$\begin{aligned}
\boldsymbol{S}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) &:= \sum_{i=1}^{n_\rho} \frac{\partial \boldsymbol{P}(\boldsymbol{\rho})}{\partial \rho_i} \dot{\rho}_i, \\
&= -\sum_{i=1}^{n_\rho} \boldsymbol{Y}^{-1} \frac{\partial \boldsymbol{Y}(\boldsymbol{\rho})}{\partial \rho_i} \boldsymbol{Y}^{-1} \dot{\rho}_i.
\end{aligned}$$

Therefore,

$$\boldsymbol{YSY} = -\sum_{i=1}^{n_\rho} \frac{\partial \boldsymbol{Y}(\boldsymbol{\rho})}{\partial \rho_i} \dot{\rho}_i := \boldsymbol{X}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}).$$

Therefore, the condition in (3.28) is equivalent to

$$\mathsf{E}\left[\boldsymbol{z}^T\left(\mathsf{sym}(\boldsymbol{AY} + \boldsymbol{BW}) + \boldsymbol{X} + \boldsymbol{YQY} + \boldsymbol{W}^T\boldsymbol{RW}\right)\boldsymbol{z}\right] \leq 0. \tag{3.30}$$

Assuming $\boldsymbol{z}(t,\boldsymbol{\rho})$ is a second order process, we can represent

$$\boldsymbol{z}(t,\boldsymbol{\rho}) := \sum_{i=0}^{\infty} \boldsymbol{z}_i(t)\phi_i(\boldsymbol{\rho}) = \boldsymbol{\Phi}_n^T(\boldsymbol{\rho})\boldsymbol{z}_{pc}(t), \tag{3.31}$$

and get

$$\boldsymbol{z}_{pc}^T \mathsf{E}\left[\boldsymbol{\Phi}_n \left(\mathsf{sym}\left(\boldsymbol{Y}\boldsymbol{A}^T + \boldsymbol{W}^T\boldsymbol{B}^T\right) + \boldsymbol{X} + \boldsymbol{Y}\boldsymbol{Q}\boldsymbol{Y} + \boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\right)\boldsymbol{\Phi}_n^T\right]\boldsymbol{z}_{pc} \leq 0,$$

or

$$\mathsf{E}\left[\mathsf{sym}\left(\boldsymbol{\Phi}_n\boldsymbol{Y}\boldsymbol{A}^T\boldsymbol{\Phi}_n^T + \boldsymbol{\Phi}_n\boldsymbol{W}^T\boldsymbol{B}^T\boldsymbol{\Phi}_n^T\right) + \boldsymbol{\Phi}_n\boldsymbol{X}\boldsymbol{\Phi}_n^T + \boldsymbol{\Phi}_n\boldsymbol{Y}\boldsymbol{Q}\boldsymbol{Y}\boldsymbol{\Phi}_n^T + \boldsymbol{\Phi}_n\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\boldsymbol{\Phi}_n^T\right] \leq 0.$$

Using (3.24) we can write $\boldsymbol{\Phi}_n\boldsymbol{Y} = \mathcal{Y}\boldsymbol{\Phi}_n$, $\boldsymbol{\Phi}_n\boldsymbol{X} = \mathcal{X}\boldsymbol{\Phi}_n$, and $\boldsymbol{\Phi}_n\boldsymbol{W}^T = \mathcal{W}^T\boldsymbol{\Phi}_m$, where $\mathcal{Y} := \boldsymbol{I}_{N+1}\otimes\boldsymbol{Y}(\boldsymbol{\rho})$, $\mathcal{X} := \boldsymbol{I}_{N+1}\otimes\boldsymbol{X}(\boldsymbol{\rho},\dot{\boldsymbol{\rho}})$, $\mathcal{W} := \boldsymbol{I}_{N+1}\otimes\boldsymbol{W}(\boldsymbol{\rho})$, and $\mathsf{sym}\,(\cdot) := (\cdot)+(\cdot)^T$. Substituting them, we get

$$\mathsf{E}\left[\mathsf{sym}\left(\mathcal{Y}\boldsymbol{\Phi}_n\boldsymbol{A}^T\boldsymbol{\Phi}_n^T + \mathcal{W}^T\boldsymbol{\Phi}_m\boldsymbol{B}^T\boldsymbol{\Phi}_n^T + \frac{1}{2}\mathcal{X}\boldsymbol{\Phi}_n\boldsymbol{\Phi}_n^T\right) + \mathcal{Y}\boldsymbol{\Phi}_n\boldsymbol{Q}\boldsymbol{\Phi}_n^T\mathcal{Y} + \mathcal{W}^T\boldsymbol{\Phi}_m\boldsymbol{R}\boldsymbol{\Phi}_m^T\mathcal{W}\right] \leq 0.$$

Since we only treat $\boldsymbol{\rho}$ as random variables, $\mathcal{X}$ can be taken out of the expectation and we arrive the constraint (3.26) in the optimization problem. Notice that (3.26) is affine in $\dot{\boldsymbol{\rho}}$, so to obtain all element in the set $\mathcal{D}_{\dot{\boldsymbol{\rho}}}$ only the vertices of the set are required.

Since $\boldsymbol{x}_0$ is given, with no initial condition uncertainty, the upper bound of the cost function can be written as

$$\mathsf{E}\left[\boldsymbol{x}_0^T\boldsymbol{P}(\boldsymbol{\rho})\boldsymbol{x}_0\right] = \boldsymbol{x}_0^T\mathsf{E}\left[\boldsymbol{P}(\boldsymbol{\rho})\right]\boldsymbol{x}_0 = \boldsymbol{x}_0^T\mathsf{E}\left[\boldsymbol{Y}^{-1}(\boldsymbol{\rho})\right]\boldsymbol{x}_0.$$

If $\boldsymbol{P}^*$ is the optimal solution, then $\boldsymbol{P} \geq \boldsymbol{P}^*$ for any $\boldsymbol{P} > 0$, or in terms of $\boldsymbol{Y} := \boldsymbol{P}^{-1}$, $\boldsymbol{Y} \leq \boldsymbol{Y}^*$ for any $\boldsymbol{Y} > 0$. Therefore, for a given $\boldsymbol{x}_0$, $\min \boldsymbol{x}_0^T\mathsf{E}\left[\boldsymbol{P}(\boldsymbol{\rho})\right]\boldsymbol{x}_0$ is achieved by $\max \mathbf{tr}\ \mathsf{E}\left[\boldsymbol{Y}(\boldsymbol{\rho})\right]$. ■

**Remark.** *Note that we have included infinite terms in the polynomial expansion as expression* *in* (3.31)*, and thus the representation is exact. The theory is presented with the exact, infinite term, polynomial chaos expansion. When the computation is needed in practice, the finite term truncation is obtained.*

The matrix variables $\boldsymbol{Y}(\boldsymbol{\rho})$ and $\boldsymbol{W}(\boldsymbol{\rho})$ in (3.26) are infinite dimensional, since they are functions of $\boldsymbol{\rho}$, which is computationally intractable. For computational tractability, we consider $N^{\text{th}}$ order approximation of $\boldsymbol{Y}(\boldsymbol{\rho})$ and $\boldsymbol{W}(\boldsymbol{\rho})$ using polynomial-chaos matrix-expansion. For the parameterization of $\boldsymbol{Y}(\boldsymbol{\rho}) = \boldsymbol{Y}(\boldsymbol{\rho})^T > 0$, we use the result on sum-of-square (SOS) representation of matrix polynomials [82], given by the following lemma.

**Lemma 19.** *(Lemma 1 in [82]) The polynomial matrix $\boldsymbol{Y}(\boldsymbol{\rho})$ of dimension $n \times n$ is SOS with respect to the monomial basis $\boldsymbol{\Psi}(\boldsymbol{\rho})$ iff there exists a symmetric matrix $\bar{\boldsymbol{Y}}$ such that*

$$\boldsymbol{Y}(\boldsymbol{\rho}) = \left(\boldsymbol{\Psi}(\boldsymbol{\rho}) \otimes \boldsymbol{I}_n\right)^T \bar{\boldsymbol{Y}} \left(\boldsymbol{\Psi}(\boldsymbol{\rho}) \otimes \boldsymbol{I}_n\right) = \boldsymbol{\Psi}_n^T(\boldsymbol{\rho}) \bar{\boldsymbol{Y}} \boldsymbol{\Psi}_n(\boldsymbol{\rho}), \text{ and } \bar{\boldsymbol{Y}} \in \mathbb{R}^{n(N+1)\times n(N+1)} \geq 0. \tag{3.32}$$

*Proof.* See Lemma 1 in [82]. ■

Therefore, $\bar{\boldsymbol{Y}} = \bar{\boldsymbol{Y}}^T \geq 0 \implies \boldsymbol{Y}(\boldsymbol{\rho}) = \boldsymbol{Y}^T(\boldsymbol{\rho}) \geq 0, \ \forall \boldsymbol{\rho}$. However, a positive definite matrix which contains polynomial basis functions is needed here so we modify Lemma 19 as follows.

**Corollary 20.** *The polynomial matrix $\boldsymbol{Y}(\boldsymbol{\rho})$ of dimension $n \times n$ is positive definite with respect to the polynomial basis $\boldsymbol{\Phi}(\boldsymbol{\rho})$ iff there exists a symmetric matrix $\bar{\boldsymbol{Y}}$ such that*

$$\boldsymbol{Y}(\boldsymbol{\rho}) = \boldsymbol{\Phi}_n^T(\boldsymbol{\rho}) \bar{\boldsymbol{Y}} \boldsymbol{\Phi}_n(\boldsymbol{\rho}), \text{ and } \bar{\boldsymbol{Y}} \in \mathbb{R}^{n(N+1)\times n(N+1)} > 0. \tag{3.33}$$

Corollary 20 yields a special structure of parameterization of the positive definite matrix $\boldsymbol{Y}(\boldsymbol{\rho})$ and the following result describes its dimension of the space that would be helpful for the controller synthesis. The following result quantifies the exact degree of freedom in the parameterization of $\boldsymbol{Y}(\boldsymbol{\rho})$.

**Corollary 21.** *$\bar{\boldsymbol{Y}}$ in* (3.32) *is in a linear space of dimension* $n(n+1)(N+1)(N+2)/4$.

*Proof.* Partition $\bar{\boldsymbol{Y}}$ as

$$\bar{\boldsymbol{Y}} := \begin{bmatrix} \bar{\boldsymbol{Y}}_{00} & \cdots & \bar{\boldsymbol{Y}}_{0N} \\ \vdots & & \vdots \\ \bar{\boldsymbol{Y}}_{N0} & \cdots & \bar{\boldsymbol{Y}}_{NN} \end{bmatrix}, \tag{3.34}$$

where $\bar{\boldsymbol{Y}}_{ij} = \bar{\boldsymbol{Y}}_{ji}^T \in \mathbb{R}^{n\times n}$. Therefore,

$$\boldsymbol{Y}(\boldsymbol{\rho}) := \boldsymbol{\Phi}_n^T \bar{\boldsymbol{Y}} \boldsymbol{\Phi}_n = \sum_{ij} \phi_i(\boldsymbol{\rho})\phi_j(\boldsymbol{\rho})\bar{\boldsymbol{Y}}_{ij}. \tag{3.35}$$

But, $\boldsymbol{Y}(\boldsymbol{\rho}) = \boldsymbol{Y}(\boldsymbol{\rho})^T$

$$\implies \bar{\boldsymbol{Y}}_{ij} = \bar{\boldsymbol{Y}}_{ij}^T. \tag{3.36}$$

Combining (3.34) and (3.36) we observe that $\bar{\boldsymbol{Y}}$ admits a linear space of dimension $n(n+1)(N+1)(N+2)/4$. ■

From (3.35), we can write

$$\begin{aligned} \boldsymbol{Y}(\boldsymbol{\rho}) &= \sum_{ij} \phi_i(\boldsymbol{\rho})\phi_j(\boldsymbol{\rho})\bar{\boldsymbol{Y}}_{ij}, & (3.37) \\ &= \left(\boldsymbol{\psi}^T(\boldsymbol{\rho}) \otimes \boldsymbol{I}_n\right) \boldsymbol{V}_{\bar{\boldsymbol{Y}}} = \boldsymbol{\psi}_n^T(\boldsymbol{\rho})\boldsymbol{V}_{\bar{\boldsymbol{Y}}}, & (3.38) \end{aligned}$$

where

$$\boldsymbol{\psi}(\boldsymbol{\rho}) := \begin{bmatrix} \phi_0^2(\boldsymbol{\rho}) \\ 2\phi_1(\boldsymbol{\rho})\phi_0(\boldsymbol{\rho}) \\ \vdots \\ 2\phi_N(\boldsymbol{\rho})\phi_{N-1}(\boldsymbol{\rho}) \\ \phi_N^2(\boldsymbol{\rho}) \end{bmatrix}$$

and

$$\boldsymbol{V}_{\bar{\boldsymbol{Y}}} := \begin{pmatrix} \bar{\boldsymbol{Y}}_{00} \\ \bar{\boldsymbol{Y}}_{10} \\ \vdots \\ \bar{\boldsymbol{Y}}_{N(N-1)} \\ \bar{\boldsymbol{Y}}_{NN} \end{pmatrix} \tag{3.39}$$

Matrix variable $\boldsymbol{W}(\boldsymbol{\rho})$ is parameterized to be linearly dependent on polynomial chaos basis functions $\phi_i(\boldsymbol{\rho})$, i.e.

$$\begin{aligned} \boldsymbol{W}(\boldsymbol{\rho}) &= \sum_{i=0}^{N} \boldsymbol{W}_i \phi_i(\boldsymbol{\rho}), \\ &= \begin{bmatrix} \phi_0(\boldsymbol{\rho})\boldsymbol{I}_m & \cdots & \phi_N(\boldsymbol{\rho})\boldsymbol{I}_m \end{bmatrix} \underbrace{\begin{pmatrix} \boldsymbol{W}_0 \\ \vdots \\ \boldsymbol{W}_N \end{pmatrix}}_{\boldsymbol{V}_{\boldsymbol{W}}}, && (3.40) \\ &= \boldsymbol{\Phi}_m^T(\boldsymbol{\rho})\boldsymbol{V}_{\boldsymbol{W}}. && (3.41) \end{aligned}$$

Substituting (3.38) into $\boldsymbol{X}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}})$ one can obtained its paramaeterization as

$$\boldsymbol{X}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) = -\sum_{i=1}^{n_\rho} \frac{\partial \boldsymbol{\psi}_n^T(\boldsymbol{\rho})}{\partial \rho_i} \dot{\rho}_i \boldsymbol{V}_{\bar{\boldsymbol{Y}}}. \tag{3.42}$$

From the definition of $\mathcal{Y}$

$$\begin{aligned}
\mathcal{Y} &:= \boldsymbol{I}_{N+1} \otimes \boldsymbol{Y}(\boldsymbol{\rho}) \\
&= \boldsymbol{I}_{N+1} \otimes \Big(\boldsymbol{\psi}_n^T(\boldsymbol{\rho})\boldsymbol{V}_{\bar{\boldsymbol{Y}}}\Big) \\
&= \Big(\boldsymbol{I}_{N+1} \otimes\ \boldsymbol{\psi}_n^T(\boldsymbol{\rho})\Big)(\boldsymbol{I}_{N+1} \otimes \boldsymbol{V}_{\bar{\boldsymbol{Y}}}), \\
&= \Big(\boldsymbol{I}_{N+1} \otimes\ \boldsymbol{\psi}_n^T(\boldsymbol{\rho})\Big)\mathcal{V}_{\bar{\boldsymbol{Y}}}. \qquad (3.43)
\end{aligned}$$

Similarly,

$$\begin{aligned}
\mathcal{W} &= \boldsymbol{I}_{N+1} \otimes \left(\boldsymbol{\Phi}_m^T(\boldsymbol{\rho})\boldsymbol{V}_{\boldsymbol{W}}\right), \\
&= \Big(\boldsymbol{I}_{N+1} \otimes\ \boldsymbol{\Phi}_m^T(\boldsymbol{\rho})\Big)(\boldsymbol{I}_{N+1} \otimes \boldsymbol{V}_{\boldsymbol{W}}), \\
&= \Big(\boldsymbol{I}_{N+1} \otimes \boldsymbol{\Phi}_m^T(\boldsymbol{\rho})\Big)\mathcal{V}_{\boldsymbol{W}}. \qquad (3.44)
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{X} &= \boldsymbol{I}_{N+1} \otimes (-\sum_{i=1}^{n_{\boldsymbol{\rho}}} \frac{\partial \boldsymbol{\psi}_n^T(\boldsymbol{\rho})}{\partial \rho_i}\dot{\rho}_i \boldsymbol{V}_{\bar{\boldsymbol{Y}}}) \qquad (3.45) \\
&= -\Big(\boldsymbol{I}_{N+1} \otimes \sum_{i=1}^{n_{\boldsymbol{\rho}}} \frac{\partial \boldsymbol{\psi}_n^T(\boldsymbol{\rho})}{\partial \rho_i}\dot{\rho}_i\Big)(\boldsymbol{I}_{N+1} \otimes \boldsymbol{V}_{\bar{\boldsymbol{Y}}}), \\
&= \Big(\boldsymbol{I}_{N+1} \otimes \sum_{i=1}^{n_{\boldsymbol{\rho}}} \frac{\partial \boldsymbol{\psi}_n^T(\boldsymbol{\rho})}{\partial \rho_i}\dot{\rho}_i\Big)\mathcal{V}_{\bar{\boldsymbol{Y}}}. \qquad (3.46)
\end{aligned}$$

We next present the synthesis algorithm for the particular parameterization considered here.

**Theorem 22.** *Controller gain*

$$\boldsymbol{K}(\boldsymbol{\rho}) = \left(\sum_{i=0}^{N} \boldsymbol{W}_i \phi_i(\boldsymbol{\rho})\right)\left(\boldsymbol{\Phi}_n^T(\boldsymbol{\rho})\bar{\boldsymbol{Y}}\boldsymbol{\Phi}_n(\boldsymbol{\rho})\right)^{-1},$$

*minimizes* (3.22) *if matrices $\bar{\boldsymbol{Y}} = \bar{\boldsymbol{Y}}^T > 0 \in \mathbb{R}^{n(N+1)\times n(N+1)}$ and $\boldsymbol{W}_i \in \mathbb{R}^{m\times n}$, are the solution* *of the following optimization problem*

$$\max_{\bar{\boldsymbol{Y}},\boldsymbol{W}_i} \mathbf{tr} \left( \sum_{i=0}^{N} \mathsf{E}\left[\phi_i^2(\boldsymbol{\rho})\right] \bar{\boldsymbol{Y}}_{ii} \right)$$

*subject to* (3.47)

$$\begin{bmatrix} \mathsf{sym}\left(\mathcal{V}_{\bar{\boldsymbol{Y}}}^T \boldsymbol{M}_1 + \mathcal{V}_{\boldsymbol{W}}^T \boldsymbol{M}_2\right) + \boldsymbol{M}_5 \mathcal{V}_{\bar{\boldsymbol{Y}}} & \mathcal{V}_{\bar{\boldsymbol{Y}}}^T \sqrt{\boldsymbol{M}_3} & \mathcal{V}_{\boldsymbol{W}}^T \sqrt{\boldsymbol{M}_4} \\ \sqrt{\boldsymbol{M}_3}\mathcal{V}_{\bar{\boldsymbol{Y}}} & -\boldsymbol{I} & \boldsymbol{0} \\ \sqrt{\boldsymbol{M}_4}\mathcal{V}_{\boldsymbol{W}} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \leq 0, \tag{3.48}$$

$$\dot{\boldsymbol{\rho}} \in \mathbb{D}_{\dot{\boldsymbol{\rho}}} \tag{3.49}$$

*where* $\mathbb{D}_{\dot{\boldsymbol{\rho}}}$ *is the vertices of the set* $\mathcal{D}_{\dot{\boldsymbol{\rho}}}$*,* $\mathcal{V}_{\bar{\boldsymbol{Y}}} := \boldsymbol{I}_{N+1} \otimes \boldsymbol{V}_{\bar{\boldsymbol{Y}}}, \mathcal{V}_{\boldsymbol{W}} := \boldsymbol{I}_{N+1} \otimes \boldsymbol{V}_{\boldsymbol{W}}$*,* $\boldsymbol{V}_{\bar{\boldsymbol{Y}}}$ *and* $\boldsymbol{V}_{\boldsymbol{W}}$ *are functions of* $\bar{\boldsymbol{Y}}$ *and* $\boldsymbol{W}_i$ *defined in* (3.39) *and* (3.40) *respectively,*

$$\boldsymbol{M}_1 := \mathsf{E}\left[(\boldsymbol{I}_{N+1} \otimes \boldsymbol{\psi}_n)\boldsymbol{\Phi}_n \boldsymbol{A}^T \boldsymbol{\Phi}_n^T\right], \tag{3.50}$$

$$\boldsymbol{M}_2 := \mathsf{E}\left[(\boldsymbol{I}_{N+1} \otimes \boldsymbol{\Phi}_m)\boldsymbol{\Phi}_m \boldsymbol{B}^T \boldsymbol{\Phi}_n^T\right], \tag{3.51}$$

$$\boldsymbol{M}_3 := \mathsf{E}\left[(\boldsymbol{I}_{N+1} \otimes \boldsymbol{\psi}_n)\boldsymbol{\Phi}_n \boldsymbol{Q} \boldsymbol{\Phi}_n^T (\boldsymbol{I}_{N+1} \otimes \boldsymbol{\psi}_n^T)\right], \tag{3.52}$$

$$\boldsymbol{M}_4 := \mathsf{E}\left[(\boldsymbol{I}_{N+1} \otimes \boldsymbol{\Phi}_m)\boldsymbol{\Phi}_m \boldsymbol{R} \boldsymbol{\Phi}_m^T (\boldsymbol{I}_{N+1} \otimes \boldsymbol{\Phi}_m^T)\right], \tag{3.53}$$

$$\boldsymbol{M}_5 := \boldsymbol{I}_{N+1} \otimes \left( \sum_{i=1}^{n_{\boldsymbol{\rho}}} \mathsf{E}\left[ \frac{\partial \boldsymbol{\psi}_n(\boldsymbol{\rho})}{\partial \rho_i} \boldsymbol{\Phi}_n \boldsymbol{\Phi}_n^T \right] \dot{\rho}_i \right) \tag{3.54}$$

*with the principal square roots of the respective matrices* $\sqrt{\boldsymbol{M}_3}$ *and* $\sqrt{\boldsymbol{M}_4}$*.*

*Proof.* Recall from (3.37), $\boldsymbol{Y}(\boldsymbol{\rho}) = \sum_{ij} \phi_i(\boldsymbol{\rho})\phi_j(\boldsymbol{\rho})\bar{\boldsymbol{Y}}_{ij}$. Noting that $\mathsf{E}\left[\phi_i(\boldsymbol{\rho})\phi_j(\boldsymbol{\rho})\right] = 0$, for $i \neq j$, the cost function in (3.25) is then

$$\mathbf{tr}\, \mathsf{E}\left[\boldsymbol{Y}(\boldsymbol{\rho})\right] = \mathbf{tr} \left( \sum_{i=0}^{N} \mathsf{E}\left[\phi_i^2(\boldsymbol{\rho})\right] \bar{\boldsymbol{Y}}_{ii} \right). \tag{3.55}$$

From (3.43) and (3.44), we can substitute $\mathcal{Y}$ and $\mathcal{W}$ in (3.26) to get

$$\mathsf{sym}\left(\mathcal{V}_{\boldsymbol{Y}}^T\boldsymbol{M}_1 + \mathcal{V}_{\boldsymbol{W}}^T\boldsymbol{M}_2\right) + \mathcal{V}_{\boldsymbol{Y}}^T\boldsymbol{M}_5 + \mathcal{V}_{\boldsymbol{Y}}^T\boldsymbol{M}_3\mathcal{V}_{\bar{\boldsymbol{Y}}} + \mathcal{V}_{\boldsymbol{W}}^T\boldsymbol{M}_4\mathcal{V}_{\boldsymbol{W}} \leq 0.$$

Applying Schur complement we get the LMI in (3.48). Since (3.54) is affine in $\dot{\boldsymbol{\rho}}$, only the vertices of $\mathcal{D}_{\dot{\boldsymbol{\rho}}}$ is needed, i.e. $\mathbb{D}_{\dot{\boldsymbol{\rho}}}$. ■

### 3.2.2 Stochastic Collocation Based Formulation

In this section we solve the synthesis problem derived in Theorem 18 in the stochastic collocation framework. In this framework, we can parameterize the matrix variables in (3.30) as

$$\boldsymbol{z}(\boldsymbol{\rho}) = \boldsymbol{L}_n(\boldsymbol{\rho})^T\boldsymbol{z}_{sc}, \tag{3.56}$$

$$\boldsymbol{Y}(\boldsymbol{\rho}) = \boldsymbol{L}_n^T(\boldsymbol{\rho})\tilde{\boldsymbol{Y}}\boldsymbol{L}_n(\boldsymbol{\rho}), \tag{3.57}$$

$$\boldsymbol{W}(\boldsymbol{\rho}) = \boldsymbol{L}_m^T(\boldsymbol{\rho})\boldsymbol{V}_{\tilde{\boldsymbol{W}}}, \tag{3.58}$$

where $\boldsymbol{L}_n = \begin{bmatrix} l_0(\boldsymbol{\rho}) \\ \vdots \\ l_N(\boldsymbol{\rho}) \end{bmatrix} \otimes \boldsymbol{I}_n$, $\tilde{\boldsymbol{Y}} = \begin{bmatrix} \tilde{\boldsymbol{Y}}_{00} & \cdots & \tilde{\boldsymbol{Y}}_{0N} \\ \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{Y}}_{N0} & \cdots & \tilde{\boldsymbol{Y}}_{NN} \end{bmatrix}$, and $\boldsymbol{V}_{\tilde{\boldsymbol{W}}} = \begin{bmatrix} \tilde{\boldsymbol{W}}_0 \\ \vdots \\ \tilde{\boldsymbol{W}}_N \end{bmatrix}$. Using this parameterization, we have the following optimization problem for synthesis.

**Theorem 23.** *Controller gain*

$$\boldsymbol{K}(\boldsymbol{\rho}) = \left(\boldsymbol{L}_m^T(\boldsymbol{\rho})\boldsymbol{V}_{\tilde{\boldsymbol{W}}}\right)\left(\boldsymbol{L}_n^T(\boldsymbol{\rho})\tilde{\boldsymbol{Y}}\boldsymbol{L}_n(\boldsymbol{\rho})\right)^{-1},$$

*minimizes* (3.22) *if* $\exists$ *matrices* $\tilde{\boldsymbol{Y}} = \tilde{\boldsymbol{Y}}^T > 0 \in \mathbb{R}^{n(N+1)\times n(N+1)}$ *and* $\tilde{\boldsymbol{W}}_i \in \mathbb{R}^{m\times n}$*, that solves the* *following optimization problem*

$$
\begin{aligned}
&\max_{\bar{\boldsymbol{Y}},\boldsymbol{W}_i} \mathbf{tr}\left(\sum_{i=0}^{N} \mathsf{E}\left[\boldsymbol{L}_{in}\right]\tilde{\boldsymbol{Y}}_{ii}\right) \\
&\textit{subject to} && (3.59) \\
&\begin{bmatrix} \boldsymbol{M}_{11,i} & \boldsymbol{M}_{12,i} & \boldsymbol{M}_{13,i} \\ \boldsymbol{M}_{12,i}^T & -\boldsymbol{I} & \mathbf{0} \\ \boldsymbol{M}_{13,i}^T & \mathbf{0} & -\boldsymbol{I} \end{bmatrix} \leq 0 \quad \textit{for } i = 0, 1, \cdots, N; && (3.60) \\
&\dot{\boldsymbol{\rho}} \in \mathbb{D}_{\dot{\boldsymbol{\rho}}} && (3.61)
\end{aligned}
$$

*where*

$$
\begin{aligned}
\boldsymbol{M}_{11,i} :=& \mathsf{sym}\left(\tilde{\boldsymbol{Y}}_{ii}^T \mathsf{E}\left[\boldsymbol{L}_{in}\right]\boldsymbol{A}^T(\boldsymbol{\rho}_i) + \tilde{\boldsymbol{W}}_i^T \boldsymbol{B}^T(\boldsymbol{\rho}_i)\right) \\
& - \mathsf{E}\left[\boldsymbol{L}_{in}\right]\sum_{j=1}^{n_{\boldsymbol{\rho}}}\left(\frac{\partial \boldsymbol{L}_{n,i}^T}{\partial \rho_i}\tilde{\boldsymbol{Y}}\boldsymbol{L}_{n,i} + \boldsymbol{L}_{n,i}^T\tilde{\boldsymbol{Y}}\frac{\partial \boldsymbol{L}_{n,i}}{\partial \rho_i}\right)\dot{\rho}_j, && (3.62) \\
\boldsymbol{M}_{12,i} :=& \tilde{\boldsymbol{Y}}_{ii}^T\sqrt{\mathsf{E}\left[\boldsymbol{L}_{in}\right]\boldsymbol{Q}}, && (3.63) \\
\boldsymbol{M}_{12,i} :=& \tilde{\boldsymbol{W}}_i^T\sqrt{\mathsf{E}\left[\boldsymbol{L}_{in}\right]\boldsymbol{R}}, && (3.64)
\end{aligned}
$$

$\mathbb{D}_{\dot{\boldsymbol{\rho}}}$ *is the vertices of the set* $\mathcal{D}_{\dot{\boldsymbol{\rho}}}$*, and*$\boldsymbol{L}_{in} = l_i \otimes \boldsymbol{I}_n$*, and* $\boldsymbol{\rho}_i$ *are the roots of the polynomial chaos basis of degree* $N + 1$*.*

*Proof.* Substituting (3.57) into (3.25) and applying Lemma 16, we have

$$\begin{aligned}
\mathbf{tr}\ \mathsf{E}\left[\boldsymbol{Y}(\boldsymbol{\rho})\right] &= \mathbf{tr}\ \mathsf{E}\left[\boldsymbol{L}_n^T(\boldsymbol{\rho})\tilde{\boldsymbol{Y}}\boldsymbol{L}_n(\boldsymbol{\rho})\right] \\
&= \mathbf{tr}\ (\mathsf{E}\left[\sum_{i=0}^{N}\sum_{j=0}^{N}\boldsymbol{L}_{in}\tilde{\boldsymbol{Y}}_{ij}\boldsymbol{L}_{jn}\right]) \\
&\approx \mathbf{tr}\ (\mathsf{E}\left[\boldsymbol{L}_{0n}\right]\tilde{\boldsymbol{Y}}_{00} + \mathsf{E}\left[\boldsymbol{L}_{1n}\right]\tilde{\boldsymbol{Y}}_{11} + \cdots + \mathsf{E}\left[\boldsymbol{L}_{Nn}\right]\tilde{\boldsymbol{Y}}_{NN}) \\
&= \mathbf{tr}\ (\sum_{i=0}^{N}\mathsf{E}\left[\boldsymbol{L}_{in}\right]\tilde{\boldsymbol{Y}}_{ii}),
\end{aligned}$$

which is the cost function we have to maximize. Then, substituting (3.56)-(3.58) into (3.30) yields

$$\begin{aligned}
\mathsf{E}\Big[\boldsymbol{z}_{sc}^T\boldsymbol{L}_n\Big(\mathsf{sym}\left(\boldsymbol{L}_n^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_n\boldsymbol{A}^T + \ \boldsymbol{V}_{\tilde{\boldsymbol{W}}}^T\boldsymbol{L}_m\boldsymbol{B}^T\right) - \sum_{j=1}^{n_{\boldsymbol{\rho}}}\left(\frac{\partial\boldsymbol{L}_n^T}{\partial\rho_i}\tilde{\boldsymbol{Y}}\boldsymbol{L}_n + \boldsymbol{L}_n^T\tilde{\boldsymbol{Y}}\frac{\partial\boldsymbol{L}_n}{\partial\rho_i}\right)\dot{\rho}_j + \\
\boldsymbol{L}_n^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_n\boldsymbol{Q}\boldsymbol{L}_n^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_n + \boldsymbol{V}_{\tilde{\boldsymbol{W}}}^T\boldsymbol{R}\boldsymbol{L}_m^T\boldsymbol{V}_{\tilde{\boldsymbol{W}}}\Big)\boldsymbol{L}_n^T\boldsymbol{z}_{sc}\Big] \leq 0
\end{aligned}$$

or

$$\begin{aligned}
\mathsf{E}\Big[\boldsymbol{L}_n\Big(\mathsf{sym}\left(\boldsymbol{L}_n^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_n\boldsymbol{A}^T + \ \boldsymbol{V}_{\tilde{\boldsymbol{W}}}^T\boldsymbol{L}_m\boldsymbol{B}^T\right)\boldsymbol{L}_n^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_n\boldsymbol{Q}\boldsymbol{L}_n^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_n + \boldsymbol{V}_{\tilde{\boldsymbol{W}}}^T\boldsymbol{R}\boldsymbol{L}_m^T\boldsymbol{V}_{\tilde{\boldsymbol{W}}}\Big)\boldsymbol{L}_n^T\Big] \\
- \sum_{j=1}^{n_{\boldsymbol{\rho}}}\mathsf{E}\left[\boldsymbol{L}_n\left(\frac{\partial\boldsymbol{L}_n^T}{\partial\rho_i}\tilde{\boldsymbol{Y}}\boldsymbol{L}_n + \boldsymbol{L}_n^T\tilde{\boldsymbol{Y}}\frac{\partial\boldsymbol{L}_n^T}{\partial\rho_i}\right)\boldsymbol{L}_n\right]\dot{\rho}_j \leq 0. \qquad (3.65)
\end{aligned}$$

Applying the Lemma 16, (3.65) can be represented as

$$\mathsf{E}\begin{bmatrix}\boldsymbol{L}_{0n} & & \\ & \ddots & \\ & & \boldsymbol{L}_{Nn}\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{G}_0 & & \\ & \ddots & \\ & & \boldsymbol{G}_N\end{bmatrix} - \begin{bmatrix}\boldsymbol{H}_0 & & \\ & \ddots & \\ & & \boldsymbol{H}_N\end{bmatrix}\right) \leq 0, \qquad (3.66)$$

where

$$\begin{aligned}\boldsymbol{G}_i \approx & \mathsf{sym}\left(\boldsymbol{L}_{n,i}^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_{n,i}\boldsymbol{A}(\rho_i)^T + \boldsymbol{V}_{\tilde{\boldsymbol{W}}}^T\boldsymbol{L}_{m,i}\boldsymbol{B}(\rho_i)^T\right) + \boldsymbol{L}_{n,i}^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_{n,i}\boldsymbol{Q}\boldsymbol{L}_{n,i}^T\tilde{\boldsymbol{Y}}\boldsymbol{L}_{n,i} + \boldsymbol{V}_{\tilde{\boldsymbol{W}}}^T\boldsymbol{R}\boldsymbol{L}_{m,i}^T\boldsymbol{V}_{\tilde{\boldsymbol{W}}}\\ =& \mathsf{sym}\left(\tilde{\boldsymbol{Y}}_{ii}\boldsymbol{A}^T(\boldsymbol{\rho}_i) + \tilde{\boldsymbol{W}}_i^T\boldsymbol{B}^T(\boldsymbol{\rho}_i)\right) + \tilde{\boldsymbol{Y}}_{ii}\boldsymbol{Q}\tilde{\boldsymbol{Y}}_{ii} + \tilde{\boldsymbol{W}}_i^T\boldsymbol{R}\tilde{\boldsymbol{W}}_i,\end{aligned}$$

and

$$\boldsymbol{H}_i \approx \sum_{j=1}^{n_\rho}\left(\frac{\partial \boldsymbol{L}_{n,i}^T}{\partial \rho_i}\tilde{\boldsymbol{Y}}\boldsymbol{L}_{n,i} + \boldsymbol{L}_{n,i}^T\tilde{\boldsymbol{Y}}\frac{\partial \boldsymbol{L}_{n,i}}{\partial \rho_i}\right)\dot{\rho}_j$$

We use the notations $\boldsymbol{L}_{n,i} := \boldsymbol{L}_n(\boldsymbol{\rho}_i)$, $\boldsymbol{L}_{m,i} := \boldsymbol{L}_m(\boldsymbol{\rho}_i)$ and $\frac{\partial \boldsymbol{L}_{n,i}^T}{\partial \rho_i} := \frac{\partial \boldsymbol{L}_n^T}{\partial \rho_i}(\rho_i)$to simplify the above expressions. Since (3.66) is in a diagonal form, it can be separated into $N+1$ constraints.

$$\begin{aligned}\mathsf{E}\left[\boldsymbol{L}_{in}\right]\Bigg[\mathsf{sym}\left(\tilde{\boldsymbol{Y}}_{ii}\boldsymbol{A}^T(\boldsymbol{\rho}_i) + \tilde{\boldsymbol{W}}_i^T\boldsymbol{B}^T(\boldsymbol{\rho}_i)\right) - \sum_{j=1}^{n_\rho}\left(\frac{\partial \boldsymbol{L}_{n,i}^T}{\partial \rho_i}\ \tilde{\boldsymbol{Y}}\boldsymbol{L}_{n,i} + \boldsymbol{L}_{n,i}^T\tilde{\boldsymbol{Y}}\frac{\partial \boldsymbol{L}_{n,i}}{\partial \rho_i}\right)\dot{\rho}_j\\ +\tilde{\boldsymbol{Y}}_{ii}\boldsymbol{Q}\tilde{\boldsymbol{Y}}_{ii} + \tilde{\boldsymbol{W}}_i^T\boldsymbol{R}\tilde{\boldsymbol{W}}_i\Bigg] \leq 0\end{aligned} \tag{3.67}$$

for $i = 0, \cdots, N$. Applying Schur complement to (3.67) we obtain $N+1$ final parameter-dependent LMIs as (3.60). Since (3.67) is affine in $\dot{\boldsymbol{\rho}}$, only the vertices of $\mathcal{D}_{\dot{\boldsymbol{\rho}}}$ is needed, i.e. $\mathbb{D}_{\dot{\boldsymbol{\rho}}}$. ■

It is noteworthy, that in (3.62) the rate term is coupled with all the elements in $\tilde{\boldsymbol{Y}}$. In a special case, when $\boldsymbol{Y}$ is not parameter dependent and consequently $\dot{\boldsymbol{\rho}}$ doesn't appear in $\dot{V}(\boldsymbol{x})$, the LMIs become independent to each other.

Comparing two optimization problem in the frameworks between Galerkin projection and Stochastic collocation one can see the benefit of stochastic collocation. While (3.48) requires much computation effort for calculating expected value of each element in matrices, LMIs in (3.60) only needs to be calculated by substituting sample points into the matrices. This difference becomes more with increasing parameter dimension and approximation order.

### 3.2.3 Stability Concern Due to Finite Term Polynomial Chaos Expansion

Theorem 18 presents the optimization problem for synthesis assuming infinite term polynomial chaos expansion of $\boldsymbol{x}(t,\boldsymbol{\rho})$. There are no approximations in that problem formulation. However, we solve this problem using finite terms in the expansion, for both Galerkin projection and stochastic collocation framework. The problem formulations in Theorem 22 and 23 are based on finite term expansion of $\boldsymbol{x}(t,\boldsymbol{\rho}) \approx \hat{\boldsymbol{x}}(t,\boldsymbol{\rho}) := \sum_{i=0}^{N} \boldsymbol{x}_i(t)\phi_i(\boldsymbol{\rho})$. Therefore, optimal control of $\hat{\boldsymbol{x}}(t,\boldsymbol{\rho})$ does not necessarily imply optimal control of $\boldsymbol{x}(t,\boldsymbol{\rho})$. In fact, we cannot conclude $\lim_{t\to\infty} \mathsf{E}\left[\|\hat{\boldsymbol{x}}(t,\boldsymbol{\rho})\|_2^2\right] \to 0 \implies \lim_{t\to\infty} \mathsf{E}\left[\|\boldsymbol{x}(t,\boldsymbol{\rho})\|_2^2\right] \to 0$. That is, we cannot guarantee exponential mean square stability (EMS) of $\boldsymbol{x}(t,\boldsymbol{\rho})$ from the EMS of $\hat{\boldsymbol{x}}(t,\boldsymbol{\rho})$.

To circumvent this problem, we guarantee stability of $\boldsymbol{x}(t,\boldsymbol{\rho})$ in the *worst-case* sense by imposing the following additional constraints,

$$\mathsf{sym}\left(\boldsymbol{A}(\boldsymbol{\rho}_{\text{wc}})\boldsymbol{Y}(\boldsymbol{\rho}_{\text{wc}}) + \boldsymbol{B}(\boldsymbol{\rho}_{\text{wc}})\boldsymbol{W}(\boldsymbol{\rho}_{\text{wc}})\right) < 0, \tag{3.68}$$

where $\boldsymbol{\rho}_{\text{wc}}$ represents the worst-case values from $\mathcal{D}_{\boldsymbol{\rho}}$. See [83] and [84] for more detail. Therefore, the results presented in this chapter can be interpreted as synthesis of parameter dependent gain $\boldsymbol{K}(\boldsymbol{\rho})$ that stabilizes the system in (3.1) in the worst-case sense and optimizes the performance, using Theorem 22 and 23, based on the first $N$ modes of $\boldsymbol{x}(t,\boldsymbol{\rho})$.

Another approach has been developed recently by Lucia et. al. [85]. In this technique, errors in the first and second moment between the approximated system and original system are analytically computed and the bounds on errors are used to design the feedback control, so that the EMS of the original system is guaranteed even with truncation. Without considering the worst-case conditions, this approach has a potential benefit that the system can be stabilized in a less conservative sense.

We will consider this approach in the LPV formulation, in our future work.

# 4. DIRECT INTERPOLATION WITH SPARSE GRID

In the Chapter 2, several controller syntheses are presented, where all optimization problems contain various parameter-dependent LMIs establishing either sufficient or necessary conditions for stability and performance of LPV systems. We denote that solving the optimization problems presented in the Chapter 2 is still a challenge due to the nonlinearities in the system matrices and infinite dimension of the unknown matrices if there exist. The common approaches to handle with the problem is either to make the problem polytopic or utilize gridding technique. However, polytopic systems render the controller design significantly conservative and the gridding approach requires tremendous computation effort if a high confidence design is needed. See the Section 1.1 for further discussion of the polytopic design and the gridding method.

This chapter is focusing on the gridding method, which requires proper amount of sample points and those locations so that the parameter-dependent LMIs can be satisfied for all elements in the parameter sets. Generally speaking, the more sample points chosen from the set, the higher chance to ensure conditions for the whole set. However, there is no theoretical analysis that can guarantee the conditions with nonlinear terms, but we can give an exam of conditions for a sample set as finer as possible or test the control system via simulations. Recently, Bandeira et. al. [46] and Araujo et. al. [47] has proposed a new technique based on Haar wavelet transformation for the gridding method that guarantees the stability of the LPV system, which inspires the method we develop in the chapter.

The control syntheses presented in this chapter are based on sparse grid technique or so called Smolyak algorithm, which produces an approximation of the optimization problem and requires much less samples than the conventional gridding methodology. The approach proposed in this chapter solves the infinite dimensional LPV problem which contains all sample points in one large size optimization problem. More detail will be described below.

## 4.1 Polynomial Interpolation

### 4.1.1 Univariate Interpolation

Starting from the simplest case, we consider a univariate case, i.e. a continuous differentiable function of a one-dimensional variable $f(x) : [-1, 1] \to \mathbb{R}$. Using (3.13) and selecting $N$ sample points, the approximation of $f(x)$ can be formulated as

$$\mathcal{U}(f) = \sum_{i=1}^{N} f(x_i) l_i(x),$$

where $x_i$ is the user chosen points between -1 and 1, $f(x_i)$ is the function value at the point $x_i$, and $l_i(x) : \mathbb{R} \to \mathbb{R}$ is the Lagrange polynomial defined in (3.13). Although the set $[-1, 1]$ is assumed for the function $f(x)$, any compact set can be linearly transformed into it, and thus $[-1, 1]$ can be considered as a general case. According to Lebesgue theorem [86, 87, 88], the error of the approximation $\mathcal{U}(f)$ is bounded in terms of infinity norm by

$$\|f - f^*\|_\infty \leq \|f - \mathcal{U}(f)\|_\infty \leq (\Lambda + 1)\|f - f^*\|_\infty, \tag{4.1}$$

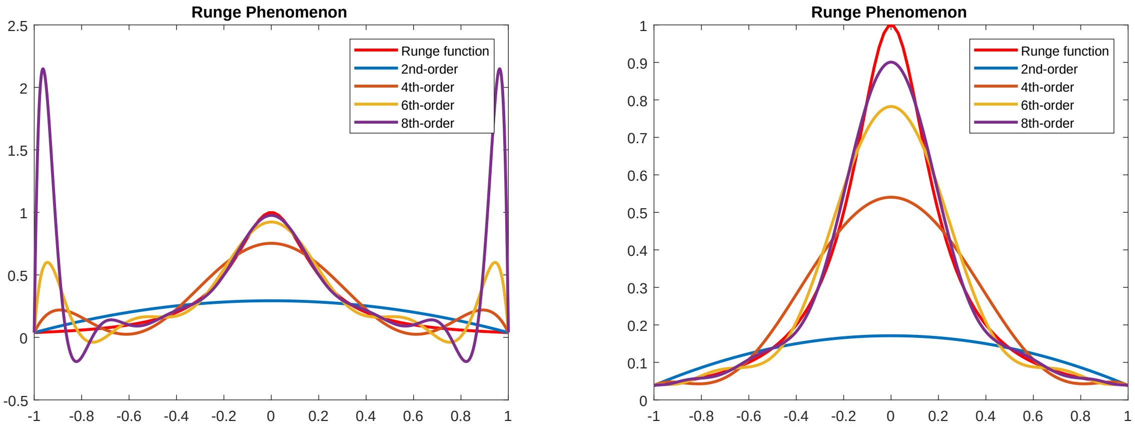
where $f^*$ is the best approximation of $f$ and the Lebesgue constant $\Lambda$ is determined by

$$\Lambda = \max_{x \in [-1,1]} \sum_{i=1}^{N} |l_i(x)|.$$

As $\Lambda$ is turning smaller, the upper bound of the approximation error goes smaller.

It is known that the accuracy is highly depending on the choices of the interpolation sample points. In particular, using Chebyshev nodes

$$x_j = -\cos\frac{\pi(j-1)}{N-1}, \qquad j = 1, \cdots, N \tag{4.2}$$



(a) Interpolation with equidistant points.

(b) Interpolation with Chebyshev nodes.

Figure 4.1: Lagrange interpolation of Runge function shows the divergence for higher order approximation.

which are determined by the roots or extrema of the Chebyshev polynomial, is a better option than using equidistant points, although both types are not optimal in the sense of error approximation. Furthermore, Using Chebyshev nodes avoids Runge's phenomenon, which shows the divergence of the polynomial interpolation near the end of the interpolation points as the order of the polynomials increases. This can be seen in Fig. 4.1, which shows the interpolation of the Runge function $f(x) = \frac{1}{1+25x^2}$. As the order of the Lagrange polynomial increase, the approximation with equidistant points is closer to the Runge function in the middle but diverging near the end points $x = -1$ and $x = 1$, while the interpolation with Chebyshev nodes efficiently stabilizes the approximation near the edges. This is because of the characteristic of the Chebychev nodes that are provided more when approaching to the boundaries.

### 4.1.2 Tensor Product

Extending the interpolation to a more general case, we consider a multivariate function $f(\boldsymbol{x}) : [-1, 1]^d \to \mathbb{R}$. The most intuitive way to interpolate a multivariate function is to utilize the univariate interpolation and extend to multivariate case, that is, a tensor product of the univariate case.

Specifically, $N$ Chebyshev nodes are determined in (4.2) for each dimension which contains $N^d$ sample points, then the tensor product is formulated as

$$\mathcal{I}(f) := \left(\mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_d}\right)(f) = \sum_{j_1=1}^{N} \cdots \sum_{j_d=1}^{N} f\left(x_{j_1}^{i_1}, \cdots, x_{j_d}^{i_d}\right) \cdot \left(l_{j_1}^{i_1} \otimes \cdots \otimes l_{j_d}^{i_d}\right), \tag{4.3}$$

where $i_k$ indicates k-th dimension of the variable set $\mathcal{X}^{i_k} \subset [-1, 1]$, $x_{m_k}^{i_k} \in \mathcal{X}^{i_k}$ indicates m-th sample point for k-th dimension of the variable set , and $l_{j_m}^{i_k}$ is m-th Lagrange polynomial for k-th dimension of the variable. The tensor product (4.3) is easy to formulate due to its straightforward logic and the numerical programming is thus very accessible. However, the number of required sample points becomes a nightmare when the dimension $d$ increases. From (4.3), it is easy to see the total number of nodes is $N^d$, which can exponentially grow with increasing $d$. This is known the *curse of dimensionality*.

## 4.2 Smolyak Algorithm

To avoid the curse of dimensionality, this section presents a sparse grid approach by Smaolyak algorithm, which was first introduced in [89] and is a linear combination of product formulation. With the algorithm, much less samples need to be considered but the interpolation property for univariate case is still preserved for multivariate case.

Following the logic in [90], we define

$$\Delta^i = \mathcal{U}^i - \mathcal{U}^{i-1}$$

with $\mathcal{U}^0 = 0$ for $i \in \{i_1, \cdots, i_d\}$, denote $|\boldsymbol{i}| = i_1 + \cdots + i_d$ for $\boldsymbol{i} = (i_1, \cdots, i_d)\mathbb{Z}^d$. Then, the Smolyak algorithm is given by

$$\mathcal{A}(k, d) = \sum_{|\boldsymbol{i}| \leq k+d} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right) \tag{4.4}$$

for $k \geq 0$, where $k$ stands for the "level" of the approximation. Identically, (4.4) can be rewritten as

$$\mathcal{A}(k,d) = \sum_{k+1 \leq |\boldsymbol{i}| \leq k+d} (-1)^{d+k-|\boldsymbol{i}|} \cdot \begin{pmatrix} d-1 \\ d+k-|\boldsymbol{i}| \end{pmatrix} \cdot \left(\mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_d}\right), \tag{4.5}$$

which is the most common expression in the literature. To compute $\mathcal{A}(k,d)$, we only need to calculate function values at the sparse grid defined as

$$\mathcal{H}(k,d) = \bigcup_{k+1 \leq |\boldsymbol{i}| \leq k+d} \left(\mathcal{X}^{i_1} \times \cdots \times \mathcal{X}^{i_d}\right).$$

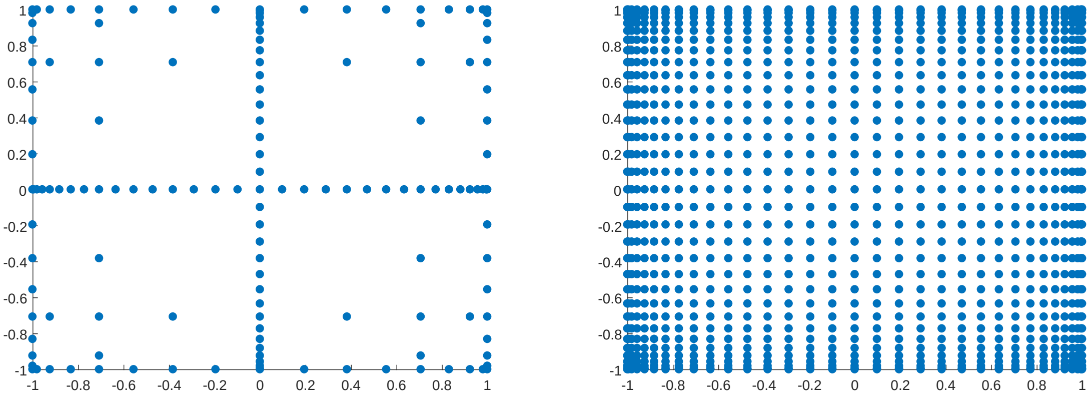
We follow the choice made in [90] sine it has been investigated in many research effort [91, 92, 93]. Thus, the chosen grid points are based on the Chebyshev nodes

$$x_j^i = -\cos\frac{\pi(j-1)}{m_i - 1}, \qquad\qquad j = 1, \cdots, m_i \tag{4.6}$$

where $m_i$ is the number of nodes used for the univariate interpolation $\mathcal{U}^i$. It is noted that the number of nodes used for each dimension is no longer assumed to be equal to each other, which is different than tensor product presented before. To obtain nested sets of grid points, i.e. $\mathcal{X}^i \subset \mathcal{X}^{i+1}$, we choose

$$m_1 = 1 \text{ and } m_i = 2^{i-1} + 1 \text{ for } i > 1. \tag{4.7}$$

This also implies nested sets of grid for different level $k$, i.e. $\mathcal{H}(k,d) \subset \mathcal{H}(k+1,d)$. The nested set is an essential property so that the amount of the samples can be efficiently reduces. Besides,



(a) Smolyak Sparse Grid

(b) Tensor Product

Figure 4.2: Comparison between sparse grid and tensor product in terms of sample points for the dimension of $d = 2$.

the total number of nodes can be calculated by [94]

$$\eta = \sum_{k+1 \le |\boldsymbol{i}| \le k+d} \prod_{n=1}^{d} r(i_n), \text{ where } r(i) := \begin{cases} 1 & \text{if } i = 1 \\ 2 & \text{if } i = 2 \\ 2^{i-2} & \text{if } i > 2 \end{cases} \cdot \tag{4.8}$$

Calculating the total number from (4.8) yields Table 4.1, which compares the number of sample points between sparse grid and tensor product grid and shows a significant improvement in terms of the size of sample points. In particular, Fig. 4.2 shows the location of each node of tensor product and sparse grid with the level $k = 4$ for the dimension $d = 2$, which can visually realize the advantage of the sparse grid.

Regarding the error analysis of the Smolyak algorithm, Barthelmann et. al. [90] has proposed the result of the upper bound for Smolyak algorithm with Chebyshev nodes as follows by using the well-known Jackson estimate.

**Theorem 24.** *For the space $\mathcal{F}_d^c = \{f : [-1,1]^d \to \mathbb{R} | D^\alpha f$ continuous if $\alpha_i \le c$ for all $i\}$ we*

| $d$ | $k$ | Sparse Grid | Tensor Product ($(2^k+1)^d$ for $k>1$) |
|---|---|---|---|
| 2 | 1 | 5 | 9 |
|  | 2 | 13 | 25 |
|  | 3 | 29 | 81 |
|  | 4 | 65 | 289 |
| 10 | 1 | 21 | 59049 |
|  | 2 | 66 | 9765625 |
|  | 3 | 286 | $3.5 \times 10^9$ |
| 20 | 1 | 41 | $3.5 \times 10^9$ |
|  | 2 | 841 | $9.5 \times 10^{13}$ |
| 50 | 1 | 101 | $7.2 \times 10^{23}$ |
|  | 2 | 5101 | $8.9 \times 10^{34}$ |

Table 4.1: Comparison of total number of sample points between sparse grid and tensor product.

*obtain*

$$\|\mathcal{I}_d - \mathcal{A}(k,d)\|_\infty \leq C_{d,c} \cdot M^{-c} \cdot (\log M)^{(c+2)(d-1)+1},$$

*where $\mathcal{I}_d$ is the identity operator in a $d$-dimensional space, $M$ is the total number of nodes that are used by $\mathcal{A}(k,d)$, and $C_{d,c}$ denote constants that depend on $d$ and $c$.*

Although the explicit expression of the constant $C_{d,c}$ is not shown in [90], the convergence of the error is guaranteed as the amount of samples is increasing to the infinity.

Numerically, although the Smolyak algorithm notably diminishes the calculation of the function value at the grid points, many univariate interpolations $\mathcal{U}^i$ still needs to be used repeatedly according to (4.5), where the linear combination of many repeat elements are implemented. Therefore, such inefficient and expensive structure has been improved by Judd et. al. [95] who proposed a more efficient implementation of the Smolyak algorithm to avoid the repetitions of univariate interpolations. The main idea is to replace the conventional nested sets with equivalent dis-joint sets, which are used for both Chebyshev nodes and Smolyak basis functions, and hence the convectional formulation (4.5) can be changed to a canonical Lagrange interpolation.

We again consider a multivariate function $f(\boldsymbol{x}) : [-1,1]^d \rightarrow \mathbb{R}$ and define an approximation of the function $f(\boldsymbol{x})$ as

$$\hat{f}(\boldsymbol{x}) := \sum_{i=1}^{M} b_i \Psi_i(\boldsymbol{x}), \tag{4.9}$$

where $\Psi_i(\boldsymbol{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$ is a $d$-dimensional basis function with associated coefficient $b_i$. In this thesis, we opt to use Chebyshev polynomials the same as in [95]. After selecting $M$ grid points according to the rule stated in (4.6) and (4.7), we calculate the function value $f(\boldsymbol{x})$ at each node and formulate a linear system

$$\begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{f}(\boldsymbol{x}_M) \end{bmatrix} = \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_M \end{bmatrix}, \tag{4.10}$$

which makes the approximation function $\hat{f}(\boldsymbol{x})$ equal to actual function $f(\boldsymbol{x})$ at each grid point. Provided that the matrix consist of basis functions has full rank, the coefficient $b_i$ is obtained by

$$\begin{bmatrix} b_1 \\ \vdots \\ b_M \end{bmatrix} = \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{f}(\boldsymbol{x}_M) \end{bmatrix}. \tag{4.11}$$

Substituting (4.11) into (4.9) yields a complete approximation. Comparing with (4.5), this formulation is more compact and easy to implement in a systematic manner.

This work can also be extended to a generalized approximation, i.e. interpolations of vectors or matrices. Thus, here we consider a matrix $\boldsymbol{F}(\boldsymbol{x}) : [-11]^d \rightarrow \mathbb{R}^{n \times m}$, where each element in the matrix is a function of $\boldsymbol{x}$. Then, an approximation of the matrix can be defined as

$$\hat{\boldsymbol{F}}(\boldsymbol{x}) := \sum_{i=1}^{M} \boldsymbol{B}_i \Psi_i(\boldsymbol{x}) = \Psi_n^T(\boldsymbol{x}) \boldsymbol{B}_{SG}, \tag{4.12}$$

where $\Psi_n(\boldsymbol{x}) = [\psi_1, \cdots, \psi_M]^T \otimes \boldsymbol{I}_n$, and $\boldsymbol{B}_{SG} = [\boldsymbol{B}_1, \cdots \boldsymbol{B}_M]^T$ with $\boldsymbol{B}_i \in \mathbb{R}^{n\times m}$. Similarly, a linear system is formulated such that $\hat{\boldsymbol{F}}(\boldsymbol{x})$ is equivalent to $\boldsymbol{F}(\boldsymbol{x})$ at Smolyak grid points, that is

$$\begin{bmatrix} \boldsymbol{F}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{F}(\boldsymbol{x}_M) \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{F}}(\boldsymbol{x}_1) \\ \vdots \\ \hat{\boldsymbol{F}}(\boldsymbol{x}_M) \end{bmatrix} = \begin{bmatrix} \Psi_1(\boldsymbol{x}_1)\boldsymbol{I}_n & \cdots & \Psi_M(\boldsymbol{x}_1)\boldsymbol{I}_n \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M)\boldsymbol{I}_n & \cdots & \Psi_M(\boldsymbol{x}_M)\boldsymbol{I}_n \end{bmatrix} \boldsymbol{B}_{SG}, \tag{4.13}$$

and the coefficient matrices are obtained from

$$\begin{aligned} \boldsymbol{B}_{SG} &= \begin{bmatrix} \Psi_1(\boldsymbol{x}_1)\boldsymbol{I}_n & \cdots & \Psi_M(\boldsymbol{x}_1)\boldsymbol{I}_n \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M)\boldsymbol{I}_n & \cdots & \Psi_M(\boldsymbol{x}_M)\boldsymbol{I}_n \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{F}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{F}(\boldsymbol{x}_M) \end{bmatrix} \\ &= \left( \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix}^{-1} \otimes \boldsymbol{I}_n \right) \begin{bmatrix} \boldsymbol{F}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{F}(\boldsymbol{x}_M) \end{bmatrix}. \end{aligned} \tag{4.14}$$

Instead of directly taking derivative of the big matrix in (4.13), one can simply use the same inverse matrix to calculate the coefficient matrice $B_{SG}$. In addition, since the inverse matrix is not depending on the function value $\boldsymbol{F}(\boldsymbol{x}_i)$, we only need to compute the inverse matrix once after the grid points are chosen.

## 4.3 Controller Synthesis

This section presents two types of LPV controllers that are developed in the Chapter 2 - linear quadratic regulator and $H_2$ control. Recall that the optimization problems for these problems are built based on some parameter-dependent LMIs, where the unknown matrices are infinite-dimensional. Therefore, the Smolyak algorithm is going to be used to approximate those matrices and the coefficient matrices are solved through a large size problem that contains various parameter-independent LMIs.

### 4.3.1 Linear Quadratic Regulator

From the Theorem 12, our aim is to parameterize $\boldsymbol{Y}(\boldsymbol{\rho}) : \mathcal{D}_{\boldsymbol{\rho}} \to \mathbb{R}^{n\times n}$ and $\boldsymbol{W}(\boldsymbol{\rho}) : \mathcal{D}_{\boldsymbol{\rho}} \to \mathbb{R}^{m\times n}$ and determine the associated coefficients. We thus let

$$\boldsymbol{Y}(\boldsymbol{\rho}) = \Psi_n^T(\boldsymbol{\rho})\boldsymbol{Y}_{SG} \tag{4.15}$$

and

$$\boldsymbol{W}(\boldsymbol{\rho}) = \Psi_m^T(\boldsymbol{\rho})\boldsymbol{W}_{SG}. \tag{4.16}$$

and substitute the above parameterization into (2.40) and (2.41), then we arrive the following optimization problem

$$\max_{\boldsymbol{Y}_{SG},\boldsymbol{W}_{SG},\boldsymbol{Z}} \quad \mathrm{tr}(\boldsymbol{Z}) \tag{4.17}$$

subject to

$$\begin{bmatrix} \mathsf{sym}\Big(\boldsymbol{A}((\boldsymbol{\rho}_i))\Psi_n^T(\boldsymbol{\rho}_i)\boldsymbol{Y}_{SG}+\boldsymbol{B}(\boldsymbol{\rho}_i)\Psi_m^T(\boldsymbol{\rho}_i)\boldsymbol{W}_{SG}\Big)-\sum_j^{n_{\boldsymbol{\rho}}} \frac{\partial \Psi_n^T(\boldsymbol{\rho}_i)}{\partial \boldsymbol{\rho}_j}\boldsymbol{Y}_{SG}\dot{\boldsymbol{\rho}}_j & \Psi_n^T(\boldsymbol{\rho}_i)\boldsymbol{Y}_{SG} & \boldsymbol{W}_{SG}^T\Psi_m(\boldsymbol{\rho}_i) \\ \Psi_n^T(\boldsymbol{\rho}_i)\boldsymbol{Y}_{SG} & -\boldsymbol{Q}^{-1} & \mathbf{0} \\ \Psi_m^T(\boldsymbol{\rho}_i)\boldsymbol{W}_{SG} & \mathbf{0} & -\boldsymbol{R}^{-1} \end{bmatrix} < 0 \tag{4.18}$$

$$\Psi_n^T(\boldsymbol{\rho}_i)\boldsymbol{Y}_{SG} - \boldsymbol{Z} > 0 \tag{4.19}$$

$$\boldsymbol{Z} > 0 \tag{4.20}$$

$$\dot{\boldsymbol{\rho}}(t) \in \mathbb{D}_{\dot{\boldsymbol{\rho}}},$$

for $i = 1, \cdots, M$, where $\boldsymbol{\rho}_i$ is the $i$-th grid point determined by the rule stated in (4.6) and (4.7). Substituted with all the grid points, the above optimization includes $M^{2^{n_{\boldsymbol{\rho}}}+1} + 1$ parameter-independent LMIs, so this can be simply solved via general LMI solvers, e.g. CVX, YALMIP.

Then, the solution to the optimization problem is taken to formulate the coefficients of (4.15) and (4.16).

However, since the LMIs in (4.18) and (4.19) are highly decoupled for $i = 1, \cdots, M$, the computational cost is expensive, which becomes nontrivial if the parameter set is high dimensional. Hence, we reformulate the optimization in the form of

$$\max_{\substack{\boldsymbol{Y}_1,\cdots,\boldsymbol{Y}_M \\ \dot{\boldsymbol{Y}}_1,\cdots,\dot{\boldsymbol{Y}}_M \\ \boldsymbol{W}_1,\cdots,\boldsymbol{W}_M \\ \boldsymbol{Z}}} \quad \mathrm{tr}(\boldsymbol{Z}) \tag{4.21}$$

subject to

$$\begin{bmatrix} \mathsf{sym}\left(\boldsymbol{A}((\boldsymbol{\rho}_i))\boldsymbol{Y}_i + \boldsymbol{B}(\boldsymbol{\rho}_i)\boldsymbol{W}_i\right) - \sum_j^{n_\rho} \dot{\boldsymbol{Y}}_j \dot{\boldsymbol{\rho}}_j & \boldsymbol{Y}_i & \boldsymbol{W}_i^T \\ \boldsymbol{Y}_i & -\boldsymbol{Q}^{-1} & \boldsymbol{0} \\ \boldsymbol{W}_i & \boldsymbol{0} & -\boldsymbol{R}^{-1} \end{bmatrix} < 0 \tag{4.22}$$

$$\boldsymbol{Y}_i - \boldsymbol{Z} > 0 \tag{4.23}$$

$$\dot{\boldsymbol{Y}}_j = \left( \begin{bmatrix} \frac{\partial \Psi_1(\boldsymbol{x}_1)}{\partial \rho_j} & \cdots & \frac{\partial \Psi_M(\boldsymbol{x}_1)}{\partial \rho_j} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Psi_1(\boldsymbol{x}_M)}{\partial \rho_j} & \cdots & \frac{\partial \Psi_M(\boldsymbol{x}_M)}{\partial \rho_j} \end{bmatrix} \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix}^{-1} \right) \begin{bmatrix} \boldsymbol{Y}_1 \\ \vdots \\ \boldsymbol{Y}_M \end{bmatrix} \text{ for } j = 1, \cdots, n_{\boldsymbol{\rho}} \tag{4.24}$$

$$\boldsymbol{Z} > 0 \tag{4.25}$$

$$\dot{\boldsymbol{\rho}}(t) \in \mathbb{D}_{\dot{\boldsymbol{\rho}}},$$

for $i = 1, \cdots, M$, where $\boldsymbol{Y}_i$, $\dot{\boldsymbol{Y}}_i$ and $\boldsymbol{W}_i$ are parameter-independent matrices that are determined by solving the optimization. It is obvious to see that (4.22) and (4.23) turn into decoupled constraints for different sample points. Besides, the equality constraints in (4.24) describe the relationship between $\boldsymbol{Y}(\boldsymbol{\rho})$ and $\dot{\boldsymbol{Y}}(\boldsymbol{\rho})$, which are vital due to the assumption in the optimization that $\boldsymbol{Y}_i$ and $\dot{\boldsymbol{Y}}_i$ are separately defined and considered as independent variables. Using (4.14) the coefficient matrices of $\boldsymbol{Y}(\boldsymbol{\rho})$, $\boldsymbol{W}(\boldsymbol{\rho})$ are obtained by

$$\boldsymbol{Y}_{SG} = \left( \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix}^{-1} \otimes \boldsymbol{I}_n \right) \begin{bmatrix} \boldsymbol{Y}_1 \\ \vdots \\ \boldsymbol{Y}_M \end{bmatrix}, \tag{4.26}$$

and

$$\boldsymbol{W}_{SG} = \left( \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix}^{-1} \otimes \boldsymbol{I}_m \right) \begin{bmatrix} \boldsymbol{W}_1 \\ \vdots \\ \boldsymbol{W}_M \end{bmatrix}. \tag{4.27}$$

One should notice that the controller synthesized through the above optimization does not guarantee the stability and performance for all the elements in the parameter set, since the problem is formulated by the interpolations that produce approximation error and hence violate the constraints. Therefore, some stress is introduced to the LMIs in (4.22) and (4.23) so that the LMIs keep away from the infeasible region. In particular, we add extra constants to the LMIs in the above optimization problem and arrive

$$\max_{\substack{\boldsymbol{Y}_1,\cdots,\boldsymbol{Y}_M \\ \dot{\boldsymbol{Y}}_1,\cdots,\dot{\boldsymbol{Y}}_{n_{\boldsymbol{\rho}}} \\ \boldsymbol{W}_1,\cdots,\boldsymbol{W}_M \\ \boldsymbol{Z}}} \quad \mathrm{tr}(\boldsymbol{Z}) \tag{4.28}$$

subject to

$$\begin{bmatrix} \mathsf{sym}\left(\boldsymbol{A}((\boldsymbol{\rho}_i))\boldsymbol{Y}_i + \boldsymbol{B}(\boldsymbol{\rho}_i)\boldsymbol{W}_i\right) - \sum_j^{n_{\boldsymbol{\rho}}} \dot{\boldsymbol{Y}}_j \dot{\boldsymbol{\rho}}_j & \boldsymbol{Y}_i & \boldsymbol{W}_i^T \\ \boldsymbol{Y}_i & -\boldsymbol{Q}^{-1} & \boldsymbol{0} \\ \boldsymbol{W}_i & \boldsymbol{0} & -\boldsymbol{R}^{-1} \end{bmatrix} < \alpha_i \boldsymbol{I}_{2n+m} \tag{4.29}$$

$$\boldsymbol{Y}_i - \boldsymbol{Z} > \beta_i \boldsymbol{I}_n \tag{4.30}$$

$$\dot{\boldsymbol{Y}}_j = \left( \begin{bmatrix} \frac{\partial \Psi_1(\boldsymbol{x}_1)}{\partial \rho_j} & \cdots & \frac{\partial \Psi_M(\boldsymbol{x}_1)}{\partial \rho_j} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Psi_1(\boldsymbol{x}_M)}{\partial \rho_j} & \cdots & \frac{\partial \Psi_M(\boldsymbol{x}_M)}{\partial \rho_j} \end{bmatrix} \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix}^{-1} \right) \begin{bmatrix} \boldsymbol{Y}_1 \\ \vdots \\ \boldsymbol{Y}_M \end{bmatrix} \tag{4.31}$$

$$\boldsymbol{Z} > 0 \tag{4.32}$$

$$\dot{\boldsymbol{\rho}}(t) \in \mathbb{D}_{\dot{\boldsymbol{\rho}}},$$

for $j = 1, \cdots, n_{\boldsymbol{\rho}}$ and $i = 1, \cdots, M$, such that the maximum approximation error does not affect the feasibility. There is no analytic approach to choose $\alpha_i > 0$ and $\beta_i > 0$ in the current state of art, so we introduce an ad-hoc algorithm as follows.

1. Pick the level $k$ of the Smolyak algorithm and the associated grid points according to the rule in (4.6) and (4.7).

2. Pick positive small values for $\alpha_i$ and $\beta_i$ from $i = 1, \cdots, M$.

3. Solve the problem (4.28)-(4.32) and check the feasibility for finer grids.

4. If the controller is infeasible, increase the values of $k$ or $\alpha_i$ and $\beta_i$ and resolve the problem until the solution is feasible for the finer grids.

The optimization contains $M^{2^{n_{\boldsymbol{\rho}}}+1} + n_{\boldsymbol{\rho}} + 1$, so the amount of the LMIs increases exponentially as the number of grid point grows up. Besides, The feasible region of the optimization problem is shrinking as $\alpha_i$ and $\beta_i$ increase, and the conservatism thus is somehow introduced. Therefore, one should check the performance index even if the solution is feasible because it is impractical to obtain a solution that has too much conservative performance.

To reduce the conservatism of relaxation ,we can keep increasing the order of the expansion and sample points according to the algorithm. However, the computational cost will be growing up exponentially as well. To resolve this problem, we can separate the order of the expansion and the number of the sample points. In particular, we can decrease the expansion to lower order $N$ and increase the amount of sample points to $M$ which is greater than $N$. By doing this, although the number of samples will cost the computation, we reduce the complexity of the expansion. As a result, the optimization in (4.28)-(4.32) can solved more efficiently. Observing (4.31), the inverse matrix is invalid due to rank deficiency, so a least-square solution is obtained.

### 4.3.2 $H_2$ Control

Based on the Theorem 13, we reformulate the problem with the same logic as previous subsection and arrive the following optimization for control synthesis.

$$\min_{\substack{\boldsymbol{X},\boldsymbol{W}\\ \boldsymbol{Z}_1,\cdots,\boldsymbol{Z}_M}} \quad \operatorname{tr}(\boldsymbol{W}) \tag{4.33}$$

$$\text{subject to} \quad \mathsf{sym}\left(\boldsymbol{A}(\boldsymbol{\rho}_i)\boldsymbol{X} + \boldsymbol{B}_u(\boldsymbol{\rho}_i)\boldsymbol{Z}_i\right) + \boldsymbol{B}_w(\boldsymbol{\rho}_i)\boldsymbol{B}_w(\boldsymbol{\rho}_i)^T < \alpha_i, \tag{4.34}$$

$$\begin{bmatrix} \boldsymbol{W} & \boldsymbol{C}_z(\boldsymbol{\rho}_i)\boldsymbol{X} + \boldsymbol{D}_u(\boldsymbol{\rho}_i)\boldsymbol{Z}_i \\ (\bullet)^T & \boldsymbol{X} \end{bmatrix} > \beta_i, \tag{4.35}$$

for $i = 1, \cdots, M$. The above optimization (4.33) - (4.35) only contains $2 * M$ LMIs since $\boldsymbol{X}$ is not depending on the parameter and no rate term is needed in(4.34). As the amount of the samples raises, the size of the problem linearly increases. Similar to (4.26) and (4.27), the coefficient of

$\boldsymbol{Z}(\boldsymbol{\rho})$ is obtained by

$$\boldsymbol{Z}_{SG} = \left( \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix}^{-1} \otimes \boldsymbol{I}_m \right) \begin{bmatrix} \boldsymbol{Z}_1 \\ \vdots \\ \boldsymbol{Z}_M \end{bmatrix}. \tag{4.36}$$

Besides, the same algorithm stated before can be implemented to solve the problem.

Regarding the Theorem 14, we parameterize all the unknown matrices

$$\boldsymbol{X}(\boldsymbol{\rho}) = \Psi_n^T(\boldsymbol{\rho})\boldsymbol{X}_{SG} \tag{4.37}$$

$$\boldsymbol{Z}(\boldsymbol{\rho}) = \Psi_m^T(\boldsymbol{\rho})\boldsymbol{z}_{SG} \tag{4.38}$$

$$\boldsymbol{W}(\boldsymbol{\rho}) = \Psi_{n_{\boldsymbol{z}}}^T(\boldsymbol{\rho})\boldsymbol{W}_{SG}, \tag{4.39}$$

and formulate an optimization problem as follows.

$$\min_{\substack{\boldsymbol{X}_1,\cdots,\boldsymbol{X}_M\\ \dot{\boldsymbol{X}}_1,\cdots,\dot{\boldsymbol{X}}_{n_{\boldsymbol{\rho}}}\\ \boldsymbol{W}_1,\cdots,\boldsymbol{W}_M\\ \boldsymbol{Z}_1,\cdots,\boldsymbol{Z}_M}} \quad \mathrm{tr}(\boldsymbol{Y}) \tag{4.40}$$

$$\text{subject to} \quad \mathsf{sym}\left(\boldsymbol{A}(\boldsymbol{\rho}_i)\boldsymbol{X}_i + \boldsymbol{B}_u(\boldsymbol{\rho}_i)\boldsymbol{Z}_i\right) + \boldsymbol{B}_w(\boldsymbol{\rho}_i)\boldsymbol{B}_w(\boldsymbol{\rho}_i)^T - \sum_{j=1}^{n_{\boldsymbol{\rho}}} \dot{\boldsymbol{X}}_j\dot{\rho}_j < \alpha_i, \tag{4.41}$$

$$\begin{bmatrix} \boldsymbol{W}_i & \boldsymbol{C}_z(\boldsymbol{\rho}_i)\boldsymbol{X}_i + \boldsymbol{D}_u(\boldsymbol{\rho}_i)\boldsymbol{Z}_i \\ (\bullet)^T & \boldsymbol{X}_i \end{bmatrix} > \beta_i, \tag{4.42}$$

$$\boldsymbol{W}_i - \boldsymbol{Y} \leq \gamma_i \tag{4.43}$$

$$\dot{\boldsymbol{X}}_j = \left( \begin{bmatrix} \frac{\partial\Psi_1(\boldsymbol{x}_1)}{\partial\rho_j} & \cdots & \frac{\partial\Psi_M(\boldsymbol{x}_1)}{\partial\rho_j} \\ \vdots & \ddots & \vdots \\ \frac{\partial\Psi_1(\boldsymbol{x}_M)}{\partial\rho_j} & \cdots & \frac{\partial\Psi_M(\boldsymbol{x}_M)}{\partial\rho_j} \end{bmatrix} \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix}^{-1} \right) \begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_M \end{bmatrix} \tag{4.44}$$

$$\dot{\boldsymbol{\rho}} \in \mathbb{D}_{\dot{\boldsymbol{\rho}}},$$

for $j = 1, \cdots, n_{\boldsymbol{\rho}}$ and $i = 1, \cdots, M$, which contains $M^{2^{n_{\boldsymbol{\rho}}}+2} + n_{\boldsymbol{\rho}}$ constraints. Accordingly, the coefficients of the associated matrices are determined by

$$\boldsymbol{X}_{SG} = \left( \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix}^{-1} \otimes \boldsymbol{I}_n \right) \begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_M \end{bmatrix},$$

$$\boldsymbol{Z}_{SG} = \left( \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix}^{-1} \otimes \boldsymbol{I}_m \right) \begin{bmatrix} \boldsymbol{Z}_1 \\ \vdots \\ \boldsymbol{Z}_M \end{bmatrix},$$

and

$$\boldsymbol{W}_{SG} = \left( \begin{bmatrix} \Psi_1(\boldsymbol{x}_1) & \cdots & \Psi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{x}_M) & \cdots & \Psi_M(\boldsymbol{x}_M) \end{bmatrix}^{-1} \otimes \boldsymbol{I}_n \right) \begin{bmatrix} \boldsymbol{W}_1 \\ \vdots \\ \boldsymbol{W}_M \end{bmatrix},$$

and using the proposed algorithm yields a controller that satisfies Theorem 14.

Observing (4.40) to(4.40), we can expect the size of the problem will grow up exponentially as the number of the grid points increases. Although the curse of dimensionality is inevitable, the methodology proposed in this chapter still renders solution to the parameter-dependent optimization with significantly less samples comparing with the conventional gridding approach. Furthermore, by choosing appropriate stress constants to the LMIs the controller may result in conservatism, but this can be resolved by introducing high level of the Smolyak algorithm.

# 5. MODELING OF TENSEGRITY SYSTEMS IN AN LPV FRAMEWORK

A tensegrity system is an arrangement of axially-loaded elements (no element bends, even though the overall structure bends), that we loosely characterize as a network of bars and cables. The bars take compressive axial loads and the cables handle tensile loads. Since failure due to axial stresses happens at higher loads than at bending, a tensegrity structure has a higher strength-to-weight ratio. Famous architect Buckminster Fuller in the 60's coined the term tensegrity, combining the words tensile and integrity. Since then, tensegrity principles have found applications in diverse domains. Tensegrity systems have been widely adopted in architecture. Donald E. Ingber [96] explained the behavior of cells by modeling them as tensegrity structures. He further showed that tensegrity structures exist at all detectable scales of the human body. Tensegrity icosahedrons are used to model biologic organisms from viruses to vertebrates, their cells, systems, and subsystems. Biotensegrity [97, 98, 99, 100] is quite an active area of research. Beyond architecture and biology, tensegrity principles are gaining popularity in robotics. NASA is considering a new terrestrial robot design based on tensegrity principles [101]. Tensegrity structures, through use of pre-stresses in the bars and cables, can also achieve controlled stiffness in the structure, which makes it attractive in applications such as soft-robotics [102], robotic locomotion [103, 104], and prosthetics [105]. In essence, tensegrity principles can be applied in the design of any structure where mass is premium, a high strength-to-weight ratio is critical, and structural stiffness needs to be tailored in both space and time. These include several applications from various engineering sectors such as aerospace (morphing airframes), energy (wind turbine blades, off-shore structures) as well as biomedical engineering (stents, minimally invasive surgical tools) and many more. Clearly, a framework is required that can efficiently model the dynamics of tensegrity structures directly from the topology of bars and cables.

The dynamics of tensegrity systems is governed by multi-body dynamics, given by a set of ordinary differential equations. This chapter develops a Lagrangian formulation for deriving these differential equations directly from the given topology of members (bars and strings), and their mass and geometric properties. Three key features of classical tensegrity systems are a) actuations only occur via cables, b) bar-to-bar connections are pin joints, and c) the bars do not spin about their longitudinal axis. These properties are exploited to simplify the equations of motion. However, the Lagrangian framework presented here is general enough to allow modeling of general multi-body systems with actuated joints. Our work of dynamics developement in this chapter is similar to a recent result from Goyal and Skalton [106], who have developed equations of motion for tensegrity systems by using Newton's second law.

The demand for more accurate simulating tools for multi-body dynamics is being challenged quite positively by the open-source community. Physics engines such as Bullet[107] and the Open Dynamics Engine (ODE)[108] have become common in robotics applications. NASA's Tensegrity Robotics Toolkit (NTRT)[109] is based on the Bullet engine. They rely on non-minimal coordinate descriptions, while other popular engines, e.g. Simscape Multibody[110], MuJoCo[111], DART[112] and Simbody[113] favor using generalized coordinates for describing the kinematics of bodies. This is because they mostly focus on robotics applications, where the configuration space is naturally reduced in the presence of joints and other constraints[114]. However, we have opted to use the Cartesian coordinate system to describe the motion of bodies, most notably, for two reasons. Skelton observed[115] that in three dimensions, a minimal coordinates approach is prone to singularities developed in the mass matrix, and therefore, the dynamics necessitates an excess coordinates description. Additionally, non-minimal descriptions of vector kinematics allows us to write elegant differential-algebraic equations (DAE), free of trigonometric terms.

To fully express a rigid body motion in Cartesian coordinates, equations describing constraints are written at the acceleration level and augmented to the equations of motion to develop a mass-descriptor form of a set of index-1 DAEs. Since only acceleration level constraints are tackled in the equations, position and velocity level constraints are violated due to errors from numerical integration. Numerous advances have been made in the past few decades addressing this very issue. A prominent method is that of generalized coordinates partitioning [48, 116] in which, utilizing Gauss-Jordan reduction, independent variables are identified and integrated numerically while dependent variables are preserved through the constraint equations. Baumgarte[117], on the other hand, instead of bypassing the problem, introduced two extra terms to the constraint equations so that the violations can be stabilized in the sense of Lyapunov. This method has been studied in different frameworks, such as in adaptive mechanisms [118], optimal sense [119], and digital control theory [120]. Stabilization allows for greater computational speed whereas coordinate partitioning is known for its superior error control characteristics, and methods that combine these two techniques [121, 122] to tap into these advantages have been developed as well.

However, parameter selection in the Baumgarte technique is a challenging task [123, 124], as systems implemented with the wrong feedback parameters have been found to become unstable. Therefore, other methods were looked into, the most common being one in which constraint violation is eliminated directly by adding appropriate correction terms to the generalized coordinates after each numerical integration. Using geometric and energy conservation constraints, Yoon et al. chose corrected positions (constrained through geometry) and velocities (constrained through energy) to be linear in the Jacobian of the constraints[125]. Yu and Chen developed an algorithm to obtain the corrected terms with the constraints at position and velocity level (both constrained through geometry) by using the Moore-Penrose inverse[126]. Citing inconsistency of units and dimensions in generalized coordinates, Blajer added an inverse of the mass matrix to the corrections of [125] as a weight matrix [127]. However, Zhang et al. compared the above two formulations in benchmark examples showing that the violation of constraints performed in the same order [128].

Furthermore, compared with the Baumgarte technique, the applied direct correction method performs more efficiently in the context of constraint violations at the position and velocity level[129, 130, 128]. However, the extent of inaccuracy in the motion, which can be determined from the violations of the energy constraints is still unclear [131, 127, 125]. Therefore, inspired by [129] and [126], one of the contributions of this thesis is to present a novel methodology that attains explicit elimination of not only position and velocity constraints, i.e. holonomic constraints, but also energy variations, i.e. nonholonomic constraints. The Lagrangian formulation lends itself favorably to an equilibrium analysis of the motion and any corresponding violations in energy conservation. Instead of considering corrected terms of position and velocity separately, we formulate a set of equations linear in these variables with energy constraints and solve the variables simultaneously in the sense of minimal norm.

This chapter is going to introduce a mathematical model of multibody dynamics in the LPV framework. The chapter describes the LPV formulation and simulations of multibody motions in much greater detail: the nomenclature used in developing the equations, the Lagrangian method for deriving the governing DAEs in the presence of constraints, an elaborate description of the holonomic constraint equations, the direct correction method deployed to ensure that these constraints are not violated at any given time, the proposed energy correction algorithm to nullify energy gain/loss occurring numerically, the coordinate partitioning method to remove the redundant states, the LPV model transformation. The results for several examples are compared with those from Simscape Multibody (MATLAB's multi-body package) and presented at the end to discuss the validity of the formulation and the benefits of the approaches proposed in the chapter. The results in this chapter will be used in Chapter refchap:example as one of the numerical examples of LPV control design.

## 5.1 Derivation of Tensegrity Dynamics

### 5.1.1 Nomenclature

The notations used in the derivation of the multibody dynamics are defined as follows.

1. Let $\boldsymbol{n}_i \in \mathbb{R}^{3\times 1}$ be the position of the $i^{\text{th}}$ node.

2. Let $\boldsymbol{N} \in \mathbb{R}^{3\times n}$ be the nodal matrix defined by

$$\boldsymbol{N} := \begin{bmatrix} \boldsymbol{n}_1 & \boldsymbol{n}_2 & \cdots & \boldsymbol{n}_n \end{bmatrix},$$

   where $n$ is the number of nodes in the tensegrity system.

3. Let $\boldsymbol{C} \in \mathbb{R}^{m\times n}$ be the connectivity matrix that defines the tensegrity system, where $m$ members are defined by connecting $n$ nodes. Specifically, if the $k^{\text{th}}$ member is defined by connecting nodes $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$, then

$$C(k,i) = -1,\ C(k,j) = 1,\ \text{ and } C(k,\cdot) = 0 \text{ otherwise.}$$

   Moreover, we can partition the $m$ members to bars and strings, resulting in a partitioned connectivity matrix

$$\boldsymbol{C} := \begin{bmatrix} \boldsymbol{C}_b \\ \boldsymbol{C}_s \end{bmatrix},$$

   where $\boldsymbol{C}_b \in \mathbb{R}^{n_b\times n}$ defines the $n_b$ bar connections and $\boldsymbol{C}_s \in \mathbb{R}^{n_s\times n}$ defines the $n_s$ string connections.

   Observing the connectivity matrix $\boldsymbol{C}_b$, we derive a matrix $\boldsymbol{L}_{p_m} \in \mathbb{R}^{n_{p_m}\times n}$ describing locations of $n_{p_m}$ point masses. These masses are placed at nodes where only strings connect. Specifically, if the $k^{\text{th}}$ point mass is positioned at the node $\boldsymbol{n}_i$, then

$$L_{p_m}(k,i) = 1,\ \text{and } L_{p_m}(k,\cdot) = 0 \text{ otherwise.}$$

4. The bars, strings and point masses are then defined as

$$\boldsymbol{B} := \boldsymbol{N}\boldsymbol{C}_b^T \in \mathbb{R}^{3\times n_b}, \boldsymbol{S} := \boldsymbol{N}\boldsymbol{C}_s^T \in \mathbb{R}^{3\times n_s},\ \text{and } \boldsymbol{P} := \boldsymbol{N}\boldsymbol{L}_{p_m}^T \in \mathbb{R}^{3\times n_{p_m}}.$$

   The $k^{\text{th}}$ column of $\boldsymbol{B}$ represents the $k^{\text{th}}$ bar, denoted by $\boldsymbol{b}_k$. Similarly, the $k^{\text{th}}$ column of $\boldsymbol{S}$

represents the $k^{\text{th}}$ string, denoted by $\boldsymbol{s}_k$, and the $k^{\text{th}}$ column of $\boldsymbol{P}$ represents the $k^{\text{th}}$ point mass, denoted by $\boldsymbol{p}_k$. Let $\boldsymbol{\theta}_k$, $\boldsymbol{\psi}_k$, and $\boldsymbol{\phi}_k$ be vectors in $\mathbb{R}^{n_b}$, $\mathbb{R}^{n_s}$, and $\mathbb{R}^{n_{pm}}$ respectively with the $k^{\text{th}}$ elements equal to one and the rest zero. Therefore, we can compactly write

$$\left.\begin{aligned}
\boldsymbol{b}_k &:= \boldsymbol{N}\boldsymbol{C}_b^T\boldsymbol{\theta}_k = \big((\boldsymbol{\theta}_k^T\boldsymbol{C}_b)\otimes\boldsymbol{I}_3\big)\,\mathbf{vec}(\boldsymbol{N}) = \boldsymbol{X}_k\boldsymbol{q},\\
\bar{\boldsymbol{b}}_k &:= \frac{1}{2}\,\big((\boldsymbol{\theta}_k^T|\boldsymbol{C}_b|)\otimes\boldsymbol{I}_3\big)\,\mathbf{vec}(\boldsymbol{N}) = \bar{\boldsymbol{X}}_k\boldsymbol{q},\\
\boldsymbol{s}_k &:= \boldsymbol{N}\boldsymbol{C}_s^T\boldsymbol{\psi}_k = \big((\boldsymbol{\psi}_k^T\boldsymbol{C}_s)\otimes\boldsymbol{I}_3\big)\,\mathbf{vec}(\boldsymbol{N}) = \boldsymbol{Y}_k\boldsymbol{q},\\
\boldsymbol{p}_k &:= \boldsymbol{N}\boldsymbol{L}_{p_m}^T\boldsymbol{\phi}_k = \big((\boldsymbol{\phi}_k^T\boldsymbol{L}_{p_m})\otimes\boldsymbol{I}_3\big)\,\mathbf{vec}(\boldsymbol{N}) = \boldsymbol{P}_k\boldsymbol{q},
\end{aligned}\right\} \tag{5.1}$$

where

$$\boldsymbol{X}_k := \big((\boldsymbol{\theta}_k^T\boldsymbol{C}_b)\otimes\boldsymbol{I}_3\big)\,, \tag{5.2}$$

$$\bar{\boldsymbol{X}}_k := \frac{1}{2}\,\big((\boldsymbol{\theta}_k^T|\boldsymbol{C}_b|)\otimes\boldsymbol{I}_3\big)\,, \tag{5.3}$$

$$\boldsymbol{Y}_k := \big((\boldsymbol{\psi}_k^T\boldsymbol{C}_s)\otimes\boldsymbol{I}_3\big)\,, \tag{5.4}$$

$$\boldsymbol{P}_k := \big((\boldsymbol{\phi}_k^T\boldsymbol{L}_{p_m})\otimes\boldsymbol{I}_3\big)\,, \tag{5.5}$$

and $\boldsymbol{q} := \mathbf{vec}(\boldsymbol{N})$ represents the Cartesian coordinates.

5. Let $\boldsymbol{F}\in\mathbb{R}^{3\times n}$ be the non-conservative force matrix defined by

$$\boldsymbol{F} := \begin{bmatrix}\boldsymbol{f}_1 & \boldsymbol{f}_2 & \cdots & \boldsymbol{f}_n\end{bmatrix},$$

where $\boldsymbol{f}_i\in\mathbb{R}^3$ is the total force acting on the $i^{\text{th}}$ node, and accordingly, the force matrix can be vectorized as $\boldsymbol{f} := \mathbf{vec}\,(\boldsymbol{F})\in\mathbb{R}^{3n}$. Here we assume a general condition where all the nodes have external forces acting on them. In practice, all nodes may not be loaded. We can set those $f_i$ to zero in the above expression. These external forces can be used to model disturbances and other loads acting on the tensegrity structure.

### 5.1.2 Kinematics

Consider the motion of $k^{\text{th}}$ bar defined by nodes $\boldsymbol{b}_k := \boldsymbol{n}_{j_k} - \boldsymbol{n}_{i_k}$. The center of mass of the bar is given by

$$\bar{\boldsymbol{b}}_k := \frac{\boldsymbol{n}_{j_k} + \boldsymbol{n}_{i_k}}{2}, \tag{5.6}$$

and its velocity is given by

$$\dot{\bar{\boldsymbol{b}}}_k := \frac{\dot{\boldsymbol{N}}_{j_k} + \dot{\boldsymbol{N}}_{i_k}}{2}. \tag{5.7}$$

To determine the angular velocity of the bar we first relate the velocities of $\boldsymbol{n}_{j_k}$ and $\boldsymbol{n}_{i_k}$ using

$$\dot{\boldsymbol{n}}_{j_k} = \dot{\boldsymbol{n}}_{i_k} + \boldsymbol{\omega}_k \times \boldsymbol{b}_k,$$

or

$$\begin{aligned} \dot{\boldsymbol{b}}_k &:= \dot{\boldsymbol{n}}_{j_k} - \dot{\boldsymbol{n}}_{i_k} \qquad (5.8) \\ &= \boldsymbol{\omega}_k \times \boldsymbol{b}_k. \qquad (5.9) \end{aligned}$$

Taking cross product with $\boldsymbol{b}_k$ on both sides we get

$$\boldsymbol{b}_k \times \dot{\boldsymbol{b}}_k = \boldsymbol{b}_k \times (\boldsymbol{\omega}_k \times \boldsymbol{b}_k).$$

Using the result from triple cross product

$$\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{b}(\boldsymbol{a} \cdot \boldsymbol{c}) - \boldsymbol{c}(\boldsymbol{a} \cdot \boldsymbol{b}),$$

we get

$$\boldsymbol{b}_k \times \dot{\boldsymbol{b}}_k = \boldsymbol{\omega}_k(\boldsymbol{b}_k \cdot \boldsymbol{b}_k) - \boldsymbol{b}_k(\boldsymbol{\omega}_k \cdot \boldsymbol{b}_k). \tag{5.10}$$

For tensegrity systems, $\boldsymbol{\omega}_k \cdot \boldsymbol{b}_k = 0$, i.e. the bar does not spin about its body axis. This is an important difference between tensegrity systems and general multi-body systems.

Therefore, for tensegrity systems, we can write the expression for angular velocity

$$\boldsymbol{\omega}_k = \frac{\boldsymbol{b}_k \times \dot{\boldsymbol{b}}_k}{\boldsymbol{b}_k^T \boldsymbol{b}_k}. \tag{5.11}$$

Noting that $\boldsymbol{b}_k^T \boldsymbol{b}_k = L_k^2$, where $L_k$ is the length of the bar and is a constant, we can write

$$\boldsymbol{\omega}_k = \frac{\boldsymbol{b}_k \times \dot{\boldsymbol{b}}_k}{L_k^2}. \tag{5.12}$$

Let the body axis be defined by $(\hat{b}_k,\ \hat{b}_2,\ \hat{b}_3)$. We can then write the angular velocity in terms of the body axis of the bar as $\boldsymbol{\omega}_k := \omega_2 \hat{b}_2 + \omega_3 \hat{b}_3$ where $\omega_2$, $\omega_3$ are respective components.

Assuming, the bar to be a cylinder with radius $r_k$ and length $L_k$, the moment of inertia of the rod in this body-fixed principal frame is

$$\boldsymbol{I}_k^{\mathrm{b}_k} := \mathbf{diag} \begin{bmatrix} \frac{m_{\boldsymbol{b}_k} r^2}{2} & \frac{m_{\boldsymbol{b}_k}}{12}(3r_k^2 + l_{\boldsymbol{b}_k}^2) & \frac{m_{\boldsymbol{b}_k}}{12}(3r_k^2 + L_k^2) \end{bmatrix}.$$

The angular momentum $h_k$ of the bar is therefore

$$\boldsymbol{h}_k := \boldsymbol{I}_k^{\mathrm{b}_k} \begin{bmatrix} 0 & \omega_2 & \omega_3 \end{bmatrix}^T \tag{5.13}$$

$$= \frac{m_{\boldsymbol{b}_k}}{12}(3r_k^2 + l_{\boldsymbol{b}_k}^2)(\omega_2 \hat{\boldsymbol{b}}_2 \; + \omega_3 \hat{\boldsymbol{b}}_3) \tag{5.14}$$

$$= \frac{m_{\boldsymbol{b}_k}}{12}(3r_k^2 + l_{\boldsymbol{b}_k}^2)\; \boldsymbol{\omega}_k, \tag{5.15}$$

$$= \frac{(3r_k^2 + l_{\boldsymbol{b}_k}^2)m_{\boldsymbol{b}_k}}{12 l_{\boldsymbol{b}_k}^2}\, \boldsymbol{b}_k \times \dot{\boldsymbol{b}}_k. \tag{5.16}$$

If $\boldsymbol{r}_k$ can be ignored, then

$$\boldsymbol{h}_k \approx \frac{m_{\boldsymbol{b}_k}}{12}\, \boldsymbol{b}_k \times \dot{\boldsymbol{b}}_k. \tag{5.17}$$

Often, hollow cylinders are used. In that case, we can substitute the appropriate inertia matrix in the expression for angular momentum.

The inertial position coordinates of $k^{\text{th}}$ point mass are given by:

$$\boldsymbol{p}_k := \boldsymbol{n}_{\boldsymbol{i_k}} \tag{5.18}$$

and its velocity given by:

$$\dot{\boldsymbol{p}}_k := \dot{\boldsymbol{n}}_{i_k} \tag{5.19}$$

### 5.1.3 Dynamics Using Lagrangian Approach

Let $\mathcal{L} := T - V$ be the Lagrangian, defined over coordinates $\boldsymbol{q}$, with components $q_i$. The equations of motion are then given by

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} - \boldsymbol{\lambda}^T(t)\frac{\partial \boldsymbol{R}(\boldsymbol{q})}{\partial q_i} = \boldsymbol{f}^T\frac{\partial \boldsymbol{q}}{\partial q_i}$$

where $\boldsymbol{R}(\boldsymbol{q}) : \mathbb{R}^{3n} \mapsto \mathbb{R}^m = \boldsymbol{0}$ depict ideal constraints that satisfy the principle of D'Alembert, first stated by Lagrange [132]. On the right, $\boldsymbol{f}$ is the non conservative force acting on the system such as externally applied forces, damper forces or disturbances. From the definition of the coordinate $\boldsymbol{q}$, one can notice that $\frac{\partial \boldsymbol{q}}{\partial q_i}$ is the $i^{\text{th}}$ column of an identity matrix $\boldsymbol{I}_{3n}$. We can therefore write the equation of motion as

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} - \boldsymbol{\lambda}^T(t)\frac{\partial \boldsymbol{R}(\boldsymbol{q})}{\partial q_i} = f_i,$$

where $f_i$ is the $i^{\text{th}}$ element of $\boldsymbol{f}$.

Substituting $\mathcal{L} := T - V$, we get the equations of motion

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) + \frac{\partial}{\partial q_i}\left(V - \boldsymbol{\lambda}^T \boldsymbol{R}(\boldsymbol{q})\right) = f_i, \tag{5.20}$$

for $i = 1, \cdots, 3n$; or in terms of $\boldsymbol{q}$ as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\boldsymbol{q}}}\right) + \frac{\partial}{\partial \boldsymbol{q}}\left(V - \boldsymbol{\lambda}^T \boldsymbol{R}(\boldsymbol{q})\right) = \boldsymbol{f}^T. \tag{5.21}$$

#### *5.1.3.1 Total Kinetic Energy*

Total kinetic energy of the system is

$$T := \sum_{k=1}^{n_b}\left(\frac{1}{2}m_{b_k}\dot{\bar{\boldsymbol{b}}}_k \cdot \dot{\bar{\boldsymbol{b}}}_k + \frac{1}{2}\boldsymbol{h}_k \cdot \boldsymbol{\omega}_k\right) + \sum_{k=1}^{n_{pm}}\left(\frac{1}{2}m_{p_k}\dot{\boldsymbol{p}}_{\boldsymbol{k}} \cdot \dot{\boldsymbol{p}}_{\boldsymbol{k}}\right)$$

The kinetic energy of the $k^{\text{th}}$ bar is

$$T_{b_k} = \frac{1}{2}\left( m_{b_k} \dot{\bar{\boldsymbol{b}}}_k^T \dot{\bar{\boldsymbol{b}}}_k + \frac{I_{b_k}}{L_k^4} (\boldsymbol{b}_k \times \dot{\boldsymbol{b}}_k) \cdot (\boldsymbol{b}_k \times \dot{\boldsymbol{b}}_k) \right),$$

where $I_{b_k} := \frac{m_{b_k}}{12}(3r_k^2 + l_{\boldsymbol{b}_k}^2)$. Simplifying

$$(\boldsymbol{b}_k \times \dot{\boldsymbol{b}}_k) \cdot (\boldsymbol{b}_k \times \dot{\boldsymbol{b}}_k) = (\boldsymbol{b}_k \cdot \boldsymbol{b}_k)(\dot{\boldsymbol{b}}_k \cdot \dot{\boldsymbol{b}}_k) - (\boldsymbol{b}_k \cdot \dot{\boldsymbol{b}}_k)(\dot{\boldsymbol{b}}_k \cdot \boldsymbol{b}_k) = l_{b_k}^2 (\dot{\boldsymbol{b}}_k \cdot \dot{\boldsymbol{b}}_k),$$

we get

$$T_{b_k} = \frac{1}{2}\left( m_{b_k} \dot{\bar{\boldsymbol{b}}}_k \cdot \dot{\bar{\boldsymbol{b}}}_k + \frac{I_{b_k}}{l_{\boldsymbol{b}_k}^2} \dot{\boldsymbol{b}}_k \cdot \dot{\boldsymbol{b}}_k \right).$$

Using (5.1), we can write $T_{b_k}$ in terms of $\dot{\boldsymbol{q}}$ as

$$T_{b_k} = \frac{1}{2} \dot{\boldsymbol{q}}^T \underbrace{\left[ m_{b_k} \bar{\boldsymbol{X}}_k^T \bar{\boldsymbol{X}}_k + \frac{I_{b_k}}{l_{\boldsymbol{b}_k}^2} \boldsymbol{X}_k^T \boldsymbol{X}_k \right]}_{:=\boldsymbol{M}_{b_k}} \dot{\boldsymbol{q}} = \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{M}_{b_k} \dot{\boldsymbol{q}}.$$

The kinetic energy of the $k^{\text{th}}$ point mass is

$$T_{p_k} = \frac{1}{2} \left( m_{p_k} \dot{\boldsymbol{p}}_{\boldsymbol{k}} \cdot \dot{\boldsymbol{p}}_{\boldsymbol{k}} \right)$$

Using (5.1), we can write $T_{p_k}$, in terms of $\dot{\boldsymbol{q}}$ as

$$T_{p_k} = \frac{1}{2} \dot{\boldsymbol{q}}^T \underbrace{\left[ m_{p_k} \boldsymbol{P}_k^T \boldsymbol{P}_k \right]}_{:=\boldsymbol{M}_{p_k}} \dot{\boldsymbol{q}} = \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{M}_{p_k} \dot{\boldsymbol{q}}$$

$$
\begin{aligned}
\implies T &= \sum_{k=1}^{n_b} T_{b_k} + \sum_{k=1}^{n_{pm}} T_{p_k} \\
&= \frac{1}{2}\dot{\boldsymbol{q}}^T \underbrace{\left(\sum_{k=1}^{n_b} \boldsymbol{M}_{b_k}\right)}_{:=\boldsymbol{M_b}} \dot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^T \underbrace{\left(\sum_{k=1}^{n_{pm}} \boldsymbol{M}_{p_k}\right)}_{:=\boldsymbol{M_p}} \dot{\boldsymbol{q}} \\
&= \frac{1}{2}\dot{\boldsymbol{q}}^T \underbrace{(\boldsymbol{M_b} + \boldsymbol{M_p})}_{:=\boldsymbol{M}} \dot{\boldsymbol{q}} \\
&= \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}\dot{\boldsymbol{q}}. \qquad (5.22)
\end{aligned}
$$

#### *5.1.3.2 Gravity Potential Energy*

Total gravitational potential energy of the system is

$$
V_g := -\sum_{k=1}^{n_b} m_{b_k}(\boldsymbol{g}\cdot\bar{\boldsymbol{b}}_k) - \sum_{k=1}^{n_{pm}} m_{p_k}(\boldsymbol{g}\cdot\boldsymbol{p}_k) = \underbrace{-\boldsymbol{g}^T\left(\sum_{k=1}^{n_b} m_{b_k}\bar{\boldsymbol{X}}_k + \sum_{k=1}^{n_{pm}} m_{p_k}\boldsymbol{P}_k\right)}_{:=\boldsymbol{G}^T}\boldsymbol{q} = -\boldsymbol{G}^T\boldsymbol{q}, \qquad (5.23)
$$

where $\boldsymbol{g} := \begin{bmatrix} 0 & 0 & -9.806 \end{bmatrix}^T$ is the gravity vector.

#### *5.1.3.3 Potential Energy of Strings Modeled as Springs*

We can model the strings as springs. In this case, the spring energy is

$$
V_s := \frac{1}{2}\sum_{k=1}^{n_s} K_k\left(\|\boldsymbol{s}_k\| - L_k\right)^2, \qquad (5.24)
$$

adds to the potential energy of the system. In this case, $l_{s_k}$ is the natural length of the spring and $K_k$ is the spring constant. In this formulation, we have to be mindful about $\|\boldsymbol{s}_k\| - l_{s_k} \geq 0$, because the strings can only exert tensile force (unidirectional), unlike regular springs. Force density $\sigma_k$, is defined as

$$\sigma_k := K_k \left(1 - \frac{l_{s_k}}{\|\boldsymbol{s}_k\|}\right), \tag{5.25}$$

which is the control variable. In the implementation, if the condition $\|\boldsymbol{s}_k\| - l_{s_k} \geq 0$ is violated for any string at any point in time, the corresponding force density is set to zero at that instant.

The spring energy in terms of $\sigma_k$ can be written as

$$\begin{aligned}
V_s : &= \frac{1}{2}\sum_{k=1}^{n_s}\left(\frac{\sigma_k^2}{K_k}\right)\|\boldsymbol{s}_k\|^2 \\
&= \frac{1}{2}\sum_{k=1}^{n_s}\left(\frac{\sigma_k^2}{K_k}\right)\boldsymbol{s}_k^T\boldsymbol{s}_k \\
&= \frac{1}{2}\boldsymbol{q}^T\left(\sum_{k=1}^{n_s}\frac{\sigma_k^2}{K_k}\boldsymbol{Y}_k^T\boldsymbol{Y}_k\right)\boldsymbol{q} \\
&= \frac{1}{2}\boldsymbol{q}^T\left(\underbrace{\begin{bmatrix}\boldsymbol{Y}_1^T\boldsymbol{Y}_1 & \cdots & \boldsymbol{Y}_{n_s}^T\boldsymbol{Y}_{n_s}\end{bmatrix}}_{:=\boldsymbol{Y}}\left(\frac{\boldsymbol{\sigma}^2}{\boldsymbol{K}}\otimes\boldsymbol{I}_{3n}\right)\right)\boldsymbol{q}, \\
&= \frac{1}{2}\boldsymbol{q}^T\boldsymbol{Y}\left(\frac{\boldsymbol{\sigma}^2}{\boldsymbol{K}}\otimes\boldsymbol{I}_{3n}\right)\boldsymbol{q}
\end{aligned} \tag{5.26}$$

where $\boldsymbol{\sigma}^2 := \begin{bmatrix}\sigma_1^2 & \cdots & \sigma_{n_s}^2\end{bmatrix}^T$, $\boldsymbol{K} := \begin{bmatrix}K_1 & \cdots & K_{n_s}\end{bmatrix}$.

#### *5.1.3.4 Damper force*

We assume a damper force between two nodes where the string/spring exists and the force is proportional to the changing rate of the string/spring length. Thus the $k^{\text{th}}$ damper can be modeled as

$$\boldsymbol{f}_{d,k} = -c\frac{d\|\boldsymbol{s}_k\|}{dt}\frac{\boldsymbol{s}_k}{\|\boldsymbol{s}_k\|}, \tag{5.27}$$

$$= -c\frac{\boldsymbol{s}_k^T\dot{\boldsymbol{s}}_k}{\|\boldsymbol{s}_k\|}\frac{\boldsymbol{s}_k}{\|\boldsymbol{s}_k\|}, \tag{5.28}$$

$$= -c\frac{(\dot{\boldsymbol{s}}_k^T\boldsymbol{s}_k)\boldsymbol{s}_k}{\boldsymbol{s}_k^T\boldsymbol{s}_k} \tag{5.29}$$

where $c$ is the damping coefficient and the direction of the force is always parallel to the string (spring). One should notice that the damper force disappears whenever the string is slack, that is $\boldsymbol{f}_{d,k} = 0$ if $\|\boldsymbol{s}_k\| - l_{s_k} \leq 0$, but the damper force always exists in the spring. To represent the total damper force acting on a node, one can utilize (5.4) and obtain $\boldsymbol{f}_d = \sum_{k=1}^{n_s}\boldsymbol{Y}_k^T\boldsymbol{f}_{d,k}$. Considering damper force as one of the members in external force, it can be added to $\boldsymbol{f}$ in (5.21).

#### *5.1.3.5 Equations of Motion*

We are now ready to derive the equations of motion. From (5.22), we have

$$\frac{d}{dt}\left(\frac{\partial T}{\partial\dot{\boldsymbol{q}}}\right) = \ddot{\boldsymbol{q}}^T\boldsymbol{M},$$

from (5.23), we have

$$\frac{\partial V_g}{\partial\boldsymbol{q}} = -\boldsymbol{G}^T,$$

and finally from (5.26), we have

$$\begin{aligned}
\frac{\partial V_s}{\partial \boldsymbol{q}} &= \frac{\partial}{\partial \boldsymbol{q}} \left( \frac{1}{2} \sum_{k=1}^{n_s} \frac{\sigma_k^2}{K_k} \boldsymbol{s}_k^T \boldsymbol{s}_k \right) \\
&= \frac{1}{2} \sum_{k=1}^{n_s} \left( \frac{\partial}{\partial \boldsymbol{q}} \left( \frac{\sigma_k^2}{K_k} \right) \boldsymbol{s}_k^T \boldsymbol{s}_k + 2 \frac{{\sigma_k}^2}{K_k} \boldsymbol{s}_k^T \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{q}} \right) \\
&= \frac{1}{2} \sum_{k=1}^{n_s} \left( 2 l_{s_k} \sigma_k \frac{\boldsymbol{s}_k^T}{\|\boldsymbol{s}_k\|^3} \boldsymbol{Y}_k \|\boldsymbol{s}_k\|^2 + 2 \frac{{\sigma_k}^2}{K_k} \boldsymbol{s}_k^T \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{q}} \right) \\
&= \sum_{k=1}^{n_s} \left( l_{s_k} \sigma_k \frac{\boldsymbol{s}_k^T}{\|\boldsymbol{s}_k\|} \boldsymbol{Y}_k + \frac{{\sigma_k}^2}{K_k} \boldsymbol{s}_k^T \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{q}} \right) \\
&= \sum_{k=1}^{n_s} \sigma_k \left( l_{s_k} \frac{\boldsymbol{s}_k^T}{\|\boldsymbol{s}_k\|} + \left( 1 - \frac{l_{s_k}}{\|\boldsymbol{s}_k\|} \right) \boldsymbol{s}_k^T \right) \boldsymbol{Y}_k \\
&= \sum_{k=1}^{n_s} \sigma_k \boldsymbol{q}^T \boldsymbol{Y}_k^T \boldsymbol{Y}_k \\
&= \boldsymbol{q}^T \boldsymbol{Y} \left( \boldsymbol{\sigma} \otimes \boldsymbol{I}_{3n} \right)
\end{aligned} \tag{5.30}$$

Therefore, the equations of motion are given by

$$\ddot{\boldsymbol{q}}^T \boldsymbol{M} - \boldsymbol{G}^T + \boldsymbol{q}^T \boldsymbol{Y} (\boldsymbol{\sigma} \otimes \boldsymbol{I}_{3n}) - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}} = \boldsymbol{f}^T,$$

or with transpose

$$\boldsymbol{M} \ddot{\boldsymbol{q}} - \left( \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}} \right)^T \boldsymbol{\lambda} = -(\boldsymbol{\sigma}^T \otimes \boldsymbol{I}_{3n}) \boldsymbol{Y}^T \boldsymbol{q} + \boldsymbol{G} + \boldsymbol{f}. \tag{5.31}$$

We next look at the constraint equation $\boldsymbol{R}(\boldsymbol{q}) = \mathbf{0}$, and compute

$$\begin{aligned}
\frac{d^2\boldsymbol{R}(\boldsymbol{q})}{dt^2} &= \frac{d}{dt}\left(\frac{d\boldsymbol{R}(\boldsymbol{q})}{dt}\right), \\
&= \frac{d}{dt}\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}\right), \\
&= \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}\right)\ddot{\boldsymbol{q}} + \begin{bmatrix} \dot{\boldsymbol{q}}^T \left(\frac{\partial^2 R_1}{\partial \boldsymbol{q}^2}\right)\dot{\boldsymbol{q}} \\ \vdots \\ \dot{\boldsymbol{q}}^T \left(\frac{\partial^2 R_m}{\partial \boldsymbol{q}^2}\right)\dot{\boldsymbol{q}} \end{bmatrix},
\end{aligned}$$

where $\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}\right)$ is a Jacobian of $\boldsymbol{R}(\boldsymbol{q})$ and $\left(\frac{\partial^2 R_i}{\partial \boldsymbol{q}^2}\right)$ is the Hessian of $R_i(\boldsymbol{q})$.

Therefore, $\frac{d^2\boldsymbol{R}(\boldsymbol{q})}{dt^2} = 0$ implies

$$-\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}\right)\ddot{\boldsymbol{q}} = \begin{bmatrix} \dot{\boldsymbol{q}}^T \left(\frac{\partial^2 R_1}{\partial \boldsymbol{q}^2}\right)\dot{\boldsymbol{q}} \\ \vdots \\ \dot{\boldsymbol{q}}^T \left(\frac{\partial^2 R_m}{\partial \boldsymbol{q}^2}\right)\dot{\boldsymbol{q}} \end{bmatrix}. \tag{5.32}$$

Combining (5.31) and (5.32), we get the final equation

$$\begin{bmatrix} \boldsymbol{M} & -\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}\right)^T \\ -\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}\right) & \mathbf{0} \end{bmatrix} \begin{pmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{bmatrix} -(\boldsymbol{\sigma}^T \otimes \boldsymbol{I}_{3n})\boldsymbol{Y}^T\boldsymbol{q} + \boldsymbol{G} + \boldsymbol{f} \\ \dot{\boldsymbol{q}}^T \left(\frac{\partial^2 R_1}{\partial \boldsymbol{q}^2}\right)\dot{\boldsymbol{q}} \\ \vdots \\ \dot{\boldsymbol{q}}^T \left(\frac{\partial^2 R_m}{\partial \boldsymbol{q}^2}\right)\dot{\boldsymbol{q}} \end{bmatrix}. \tag{5.33}$$

Defining,

$$
\begin{aligned}
\boldsymbol{R}_{\boldsymbol{q}} &:= \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}, \\
\boldsymbol{\xi}_1 &:= -(\boldsymbol{\sigma}^T \otimes \boldsymbol{I}_{3n})\boldsymbol{Y}^T\boldsymbol{q} + \boldsymbol{G} + \boldsymbol{f}, \\
\boldsymbol{\xi}_2 &:= \begin{bmatrix} \dot{\boldsymbol{q}}^T \left(\frac{\partial^2 R_1}{\partial \boldsymbol{q}^2}\right) \dot{\boldsymbol{q}} \\ \vdots \\ \dot{\boldsymbol{q}}^T \left(\frac{\partial^2 R_m}{\partial \boldsymbol{q}^2}\right) \dot{\boldsymbol{q}} \end{bmatrix},
\end{aligned}
$$

we can analytically express $\ddot{\boldsymbol{q}}$ and $\boldsymbol{\lambda}$ as

$$
\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}\left[\boldsymbol{\xi}_1 - \boldsymbol{R}_{\boldsymbol{q}}^T\left(\boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{M}^{-1}\boldsymbol{R}_{\boldsymbol{q}}^T\right)^{-1}\left(\boldsymbol{\xi}_2 + \boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{M}^{-1}\boldsymbol{\xi}_1\right)\right] = \boldsymbol{\xi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\sigma}, \boldsymbol{f}), \tag{5.34}
$$

$$
\boldsymbol{\lambda} = -\left(\boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{M}^{-1}\boldsymbol{R}_{\boldsymbol{q}}^T\right)^{-1}\left(\boldsymbol{\xi}_2 + \boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{M}^{-1}\boldsymbol{\xi}_1\right). \tag{5.35}
$$

In this formulation, numerical difficulties may occur when solving the above equations of motion. Here we assume that the mass matrix $\boldsymbol{M}$ is invertible since the kinetic energy is always positive. Small inertia can also cause numerical ill conditioning. In addition, redundant constraints can also cause singularity in $\boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{M}^{-1}\boldsymbol{R}_{\boldsymbol{q}}^T$.

#### *5.1.3.6 Ideal Constraints*

Ideal constraints, as stated earlier, are those that satisfy D'Alembert's principle. In the current derivation, we only consider holonomic constraints, that is, they reduce the dimension of the space of accessible configurations, but do not restrict motion and paths within the reduced dimension [133]. Mathematically, the constraint equations can be expressed as $\boldsymbol{R}(\boldsymbol{q}) = \boldsymbol{0}$, where $\boldsymbol{q}$ is a function of time. Commonly constraints will include bar-length constraints that are quadratic in $\boldsymbol{q}$, and boundary conditions on $\boldsymbol{q}$ that will be linear in $\boldsymbol{q}$. Bar length constraints are of the type $\boldsymbol{b}_k^T\boldsymbol{b}_k - l_{\boldsymbol{b}_k}^2 = 0$, which in terms of $\boldsymbol{q}$ are $\boldsymbol{q}^T\boldsymbol{X}_k^T\boldsymbol{X}_k\boldsymbol{q} - l_{\boldsymbol{b}_k}^2 = 0$.

Therefore, for these two cases

$$\boldsymbol{R}(\boldsymbol{q}) := \begin{bmatrix} \boldsymbol{A}\boldsymbol{q} - \boldsymbol{b} \\ \boldsymbol{q}^T \boldsymbol{X}_1^T \boldsymbol{X}_1 \boldsymbol{q} - l_{b_1}^2 \\ \vdots \\ \boldsymbol{q}^T \boldsymbol{X}_{n_b}^T \boldsymbol{X}_{n_b} \boldsymbol{q} - l_{b_{n_b}}^2 \end{bmatrix} = 0. \tag{5.36}$$

Therefore,

$$\boldsymbol{R}_{\boldsymbol{q}} := \left( \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}} \right) = \begin{bmatrix} \boldsymbol{A} \\ 2\boldsymbol{q}^T \boldsymbol{X}_1^T \boldsymbol{X}_1 \\ \vdots \\ 2\boldsymbol{q}^T \boldsymbol{X}_{n_b}^T \boldsymbol{X}_{n_b} \end{bmatrix}, \tag{5.37}$$

and

$$\begin{bmatrix} \dot{\boldsymbol{q}}^T \left( \frac{\partial^2 R_1}{\partial \boldsymbol{q}^2} \right) \dot{\boldsymbol{q}} \\ \vdots \\ \dot{\boldsymbol{q}}^T \left( \frac{\partial^2 R_m}{\partial \boldsymbol{q}^2} \right) \dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ 2\dot{\boldsymbol{q}}^T \boldsymbol{X}_1^T \boldsymbol{X}_1 \dot{\boldsymbol{q}} \\ \vdots \\ 2\dot{\boldsymbol{q}}^T \boldsymbol{X}_{n_b}^T \boldsymbol{X}_{n_b} \dot{\boldsymbol{q}} \end{bmatrix}.$$

## 5.2 LPV Model Transformation

This section develops the tensegrity dynamical system from equations of motion to a standard quasi-LPV system. There are two crucial techniques we use for the transformation. One is to utilize coordinate partitioning to remove the redundant states and the other one is to define virtual scheduling parameter such that the nonlinear dynamics is transformed to an LPV model.

Expending and rearranging (5.34) yields

$$\ddot{\boldsymbol{q}} = \hat{\boldsymbol{M}}\hat{\boldsymbol{Y}}\boldsymbol{u} - \boldsymbol{M}^{-1}\boldsymbol{R_q}(\boldsymbol{R_q}\boldsymbol{M}^{-1}\boldsymbol{R_q^T})^{-1}\boldsymbol{\xi}_2\dot{\boldsymbol{q}} - \hat{\boldsymbol{M}}(\boldsymbol{G}+\boldsymbol{f}), \tag{5.38}$$

where $\boldsymbol{u} = \boldsymbol{\sigma}$ is a control vector,

$$\hat{\boldsymbol{M}} = \left[\boldsymbol{M}^{-1}\boldsymbol{R_q^T}(\boldsymbol{R_q}\boldsymbol{M}^{-1}\boldsymbol{R_q^T})^{-1}\boldsymbol{R_q}\boldsymbol{M}^{-1} - \boldsymbol{M}^{-1}\right]$$

and $\hat{\boldsymbol{Y}} \in \mathbb{R}^{3n\times n_u}$ such that **vec** $\left(\hat{\boldsymbol{Y}}\right) = \boldsymbol{Y}^T\boldsymbol{q}$, which is a rearrangement of the elements of column vector $\boldsymbol{Y}^T\boldsymbol{q}$ into a matrix of dimension $3n \times n_u$. The rearrangement of the equation implies that (5.38) is affine in the control variable $\boldsymbol{u}$ and $\boldsymbol{G}+\boldsymbol{f}$. This is comparable to the standard LPV model stated in (2.44), where $\boldsymbol{G}+\boldsymbol{f}$ is considered as exogenous disturbances $\boldsymbol{w}$.

Since the controller designed in previous chapters is not valid for a control system that is represented in a non-minimum set of coordinates, e.g. Cartesian coordinates , used in this chapter, the model reduction is necessary to achieve such that the model is described in a minimum set of coordinate. Here we utilize the coordinate partitioning method [48] to (5.38) so that the chosen independent coordinates is used to formulate a minimal realization in a state space form.

A a set of independent coordinates $\boldsymbol{q}_i$ is determined by Gaussian elimination such that

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_d \\ \boldsymbol{q}_i \end{bmatrix}, \tag{5.39}$$

where $\boldsymbol{q}_d$ is a dependent coordinates vector. In particular, the Jacobian matrix (5.37) is transformed to the echelon form, where the states corresponding to the full rank matrix with dimensionality of $m\times m$ is dependent coordinates $\boldsymbol{q}_d$ and the rest of states are called independent coordinates $\boldsymbol{q}_i$. One should notice that the dimension of $\boldsymbol{q}_i$ is the same as the degree of freedom of the system, since it is the minimum coordinates to describe the dynamics. Moreover, the original set of coordinates may not be stack as (5.39), so some of the matrices describing the bars and cables are affected but the main structure of the dynamical model does not change. This can be corrected by adjusting the matrices in (5.2)-(5.5).

Taking a time derivative of the constraints, i.e.

$$\dot{\boldsymbol{R}} = \boldsymbol{R_q}\dot{\boldsymbol{q}} = \boldsymbol{R}_{\boldsymbol{q}_d}\dot{\boldsymbol{q}}_d + \boldsymbol{R}_{\boldsymbol{q}_i}\dot{\boldsymbol{q}}_i = 0,$$

we can express the dependent coordinates in terms of the independent coordinates as

$$\dot{\boldsymbol{q}}_d = -\boldsymbol{R}_{\boldsymbol{q}_d}^{-1}\boldsymbol{R}_{\boldsymbol{q}_i}\dot{\boldsymbol{q}}_i,$$

where $\boldsymbol{R}_{\boldsymbol{q}_d}$ has full rank if there is no redundant constraint, or we can state

$$\dot{\boldsymbol{q}} = \begin{bmatrix} -\boldsymbol{R}_{\boldsymbol{q}_d}^{-1}\boldsymbol{R}_{\boldsymbol{q}_i} \\ \boldsymbol{I} \end{bmatrix} \dot{\boldsymbol{q}}_i \tag{5.40}$$

Substituting (5.40) into (5.38) we get

$$\ddot{\boldsymbol{q}} = \hat{\boldsymbol{M}}\hat{\boldsymbol{Y}}\boldsymbol{u} - \boldsymbol{M}^{-1}\boldsymbol{R_q}(\boldsymbol{R_q}\boldsymbol{M}^{-1}\boldsymbol{R}_{\boldsymbol{q}}^T)^{-1}\boldsymbol{\xi}_2 \begin{bmatrix} -\boldsymbol{R}_{\boldsymbol{q}_d}^{-1}\boldsymbol{R}_{\boldsymbol{q}_i} \\ \boldsymbol{I} \end{bmatrix} \dot{\boldsymbol{q}}_i - \hat{\boldsymbol{M}}(\boldsymbol{G} + \boldsymbol{f}). \tag{5.41}$$

Since the first and second term on the right hand side of (5.41) is affine in $\boldsymbol{u}$ and $\dot{\boldsymbol{q}}_i$ respectively, introducing the parameters $\boldsymbol{\rho} = \boldsymbol{x}$ yields the system in an quasi-LPV form, i.e.

$$\begin{bmatrix} \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} 0 & \begin{bmatrix} -\boldsymbol{R}_{\boldsymbol{q}_d}^{-1}\boldsymbol{R}_{\boldsymbol{q}_i} \\ \boldsymbol{I} \end{bmatrix} \\ 0 & \boldsymbol{A}_{22}(\boldsymbol{\rho}) \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \dot{\boldsymbol{q}}_i \end{bmatrix} + \begin{bmatrix} 0 \\ \hat{\boldsymbol{M}}\hat{\boldsymbol{Y}} \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} 0 \\ \hat{\boldsymbol{M}} \end{bmatrix} \boldsymbol{w}, \tag{5.42}$$

where $\boldsymbol{A}_{22} = -\boldsymbol{M}^{-1}\boldsymbol{R_q}(\boldsymbol{R_q}\boldsymbol{M}^{-1}\boldsymbol{R}_{\boldsymbol{q}}^T)^{-1}\boldsymbol{\xi}_2 \begin{bmatrix} -\boldsymbol{R}_{\boldsymbol{q}_d}^{-1}\boldsymbol{R}_{\boldsymbol{q}_i} \\ \boldsymbol{I} \end{bmatrix}$ and $\boldsymbol{w} = \boldsymbol{G} + \boldsymbol{f}$ is considered as disturbances that contain the gravity field and external forces. It is noted that (5.42) is a mapping from the independent coordinates to non-minimum coordinates which contain both independent and dependent coordinates. Therefore, we can extract the needed terms and associated matrices out to get a state-space in terms of independent coordinates only.

Therefore, defining $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q}_i \\ \dot{\boldsymbol{q}}_i \end{bmatrix}$ we arrive a final quasi-LPV form of

$$\begin{aligned} \dot{\boldsymbol{x}} = & \begin{bmatrix} 0 & \boldsymbol{I} \\ 0 & \bar{\boldsymbol{A}}_{22}(\rho) \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ \boldsymbol{B}_2^u(\boldsymbol{\rho}) \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} 0 \\ \boldsymbol{B}_2^w(\boldsymbol{\rho}) \end{bmatrix} \boldsymbol{w} \\ = & \boldsymbol{A}(\boldsymbol{\rho})\boldsymbol{x} + \boldsymbol{B}_u(\boldsymbol{\rho})\boldsymbol{u} + \boldsymbol{B}_w(\boldsymbol{\rho})\boldsymbol{w}, \end{aligned} \tag{5.43}$$

where $\bar{\boldsymbol{A}}_{22} \in \mathbb{R}^{3n-m \times 3n-m}$ is the last $3n - m$ rows of $\boldsymbol{A}_{22}$, $\boldsymbol{B}_2^u \in \mathbb{R}^{3n-m \times n_s}$ is the last $3n - m$ rows of $\hat{\boldsymbol{M}}\hat{\boldsymbol{Y}}$ and $\boldsymbol{B}_2^w \in \mathbb{R}^{3n-m \times n_w}$ is the last $3n - m$ rows of $\hat{\boldsymbol{M}}$.

The above equation depicting the motion of the system is in the minimal realization. One should notice that the minimal realization is for controller design only and the simulation of structure dynamics is described by the original equations of motion (5.33) in terms of non-minimum set of coordinates. Furthermore, the system matrices in (5.43) consists of highly nonlinear terms, e.g. polynomials and polynomial fractions, and there is no trigonometric function in the matrices. This is a benefit of non-minimum coordinates based formulation that reduces the complexity of the model.

### 5.3 DAE Correction

The holonomic constraints are converted to differential equations by differentiating them twice. This results in constraints on acceleration, which are satisfied exactly. However, the position and velocity constraints get violated due to errors in numerical integration. In addition to the constraints, numerical errors also violate energy conservation. For this reason, inspired by the direct correction approach in [129] and [126], we use the idea of constraint variations and derive a system of linear equations to correct for errors in numerical integration. To account for energy conservation, we also include variation in the total system energy in the formulation.

Considering the vector of coordinates and its time derivative that need to be corrected for the original constraints,

$$\boldsymbol{q}^c = \boldsymbol{q}^u + \delta\boldsymbol{q}, \tag{5.44}$$

where $\boldsymbol{q}^u$ denotes the uncorrected position, obtained from numerical integration, $\boldsymbol{q}^c$ the corrected position, and $\delta\boldsymbol{q}$ is the correction required to satisfy the constraint. Therefore

$$\boldsymbol{R}(\boldsymbol{q}^c) = \boldsymbol{R}(\boldsymbol{q}^u + \delta\boldsymbol{q}) = \boldsymbol{R}(\boldsymbol{q}^u) + \boldsymbol{R_q}\delta\boldsymbol{q} = \boldsymbol{0}, \tag{5.45}$$

where $\boldsymbol{R_q}$ is a Jacobian matrix defined in (5.37).

Similarly, the time derivative of the holonomic constraint should satisfy

$$\frac{d\boldsymbol{R}}{dt} = \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{q}}\dot{\boldsymbol{q}} = \boldsymbol{0}. \tag{5.46}$$

With

$$\dot{\boldsymbol{q}}^c = \dot{\boldsymbol{q}}^u + \delta\dot{\boldsymbol{q}}, \tag{5.47}$$

we get

$$\left[\boldsymbol{R}_q(\boldsymbol{q}^u) + \frac{\partial\boldsymbol{R_q}}{\partial\boldsymbol{q}}\delta\boldsymbol{q}\right](\dot{\boldsymbol{q}}^u + \delta\dot{\boldsymbol{q}}) = \boldsymbol{0}, \tag{5.48}$$

where $\frac{\partial \boldsymbol{R}_{\boldsymbol{q}}}{\partial \boldsymbol{q}}$ is a third-order tensor and can be represented in a matrix form

$$\frac{\partial \boldsymbol{R}_{\boldsymbol{q}}}{\partial \boldsymbol{q}} \delta \boldsymbol{q} = \begin{bmatrix} \frac{\partial \boldsymbol{R}_{\boldsymbol{q}}}{\partial q_1} \delta \boldsymbol{q} & \cdots & \frac{\partial \boldsymbol{R}_{\boldsymbol{q}}}{\partial q_{3n}} \delta \boldsymbol{q} \end{bmatrix}.$$

Ignoring higher order terms in (5.48), we get

$$\boldsymbol{R}_{\boldsymbol{q}} \dot{\boldsymbol{q}}^u + \boldsymbol{R}_{\boldsymbol{q}} \delta \dot{\boldsymbol{q}} + \left( \frac{\partial \boldsymbol{R}_{\boldsymbol{q}}}{\partial \boldsymbol{q}} \delta \boldsymbol{q} \right) \dot{\boldsymbol{q}}^u = \boldsymbol{0}, \tag{5.49}$$

where the third term can be reformulated as

$$\begin{aligned} \left( \frac{\partial \boldsymbol{R}_{\boldsymbol{q}}}{\partial \boldsymbol{q}} \delta \boldsymbol{q} \right) \dot{\boldsymbol{q}}^u &= \begin{bmatrix} \frac{\partial \boldsymbol{R}_{\boldsymbol{q}}}{\partial q_1} \delta \boldsymbol{q} & \cdots & \frac{\partial \boldsymbol{R}_{\boldsymbol{q}}}{\partial q_{3n}} \delta \boldsymbol{q} \end{bmatrix} \dot{\boldsymbol{q}}^u, \\ &= \sum_{i=1}^{3n} \frac{\partial \boldsymbol{R}_{\boldsymbol{q}}}{\partial q_i} \dot{q}_i^u \delta \boldsymbol{q}, \\ &= \left( \sum_{i=1}^{3n} \frac{\partial \boldsymbol{R}_{\boldsymbol{q}}}{\partial q_i} \dot{\boldsymbol{q}}^u \boldsymbol{\alpha}_i \right) \delta \boldsymbol{q}, \\ &= \boldsymbol{Q} \delta \boldsymbol{q}, \end{aligned}$$

where $\boldsymbol{Q} := \left( \sum_{i=1}^{3n} \frac{\partial \boldsymbol{R}_{\boldsymbol{q}}}{\partial q_i} \dot{\boldsymbol{q}}^u \boldsymbol{\alpha}_i \right)$, and $\boldsymbol{\alpha}_i$ is the $i^{\text{th}}$ column of the identity matrix $\boldsymbol{I}_{3n}$.

Then (5.49) becomes

$$\boldsymbol{R}_{\boldsymbol{q}} \dot{\boldsymbol{q}}^u + \boldsymbol{R}_{\boldsymbol{q}} \delta \dot{\boldsymbol{q}} + \boldsymbol{Q} \delta \boldsymbol{q} = \boldsymbol{0}. \tag{5.50}$$

Combining (5.45) and (5.50), we obtain the following system of linear equations

$$\begin{bmatrix} \boldsymbol{R}_{\boldsymbol{q}} & \boldsymbol{0} \\ \boldsymbol{Q} & \boldsymbol{R}_{\boldsymbol{q}} \end{bmatrix} \begin{bmatrix} \delta \boldsymbol{q} \\ \delta \dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{R} \\ -\boldsymbol{R}_{\boldsymbol{q}} \dot{\boldsymbol{q}}^u \end{bmatrix}. \tag{5.51}$$

Since the matrix in (5.51) has fewer rows than columns ($2m < 6n$), it doesn't have full column rank and there exist infinite solutions. In [129], the author applied Moore-Penrose inverse to minimize the 2-norm of the solution. However, the corrections in $\delta \boldsymbol{q}$ and $\delta \dot{\boldsymbol{q}}$ from such a formulation, modifies the potential and kinetic energy of the system and violates the conservation of mechanical energy. In this work, we extend the work in [129], by explicitly constraining the energy change, due to $\delta \boldsymbol{q}$ and $\delta \dot{\boldsymbol{q}}$, to be zero. This will result in an additional linear equation in $\delta \boldsymbol{q}$ and $\delta \dot{\boldsymbol{q}}$. The derivation of that constraint equation is as follows. We consider a general formulation, where the work done by external forces are accounted for.

The total energy of the system is defined as

$$E(\boldsymbol{q}, \dot{\boldsymbol{q}}) := T(\dot{\boldsymbol{q}}) + V_s(\boldsymbol{q}) + V_g(\boldsymbol{q}), \tag{5.52}$$

and energy conservation states that the total energy at any time $t$ is the sum of the total energy at initial time and the work done by non conservative forces, i.e.

$$E(\boldsymbol{q}, \dot{\boldsymbol{q}}) = E(\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0) + \int_C \boldsymbol{f} \cdot d\boldsymbol{q}, \tag{5.53}$$

where $(\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0)$ is the initial condition, $\boldsymbol{f}$ is the external force, which can be either state or time dependent, and the integration is done over path $C$ connecting $\boldsymbol{q}_0$ to $\boldsymbol{q}$.

Let the work done by force $\boldsymbol{f}$ be $W_{\boldsymbol{f}}$, i.e.

$$\begin{aligned} W_{\boldsymbol{f}} &= \int_C \boldsymbol{f} \cdot d\boldsymbol{q}, \\ &= \int_{t_0}^{t} \boldsymbol{f}^T \frac{d\boldsymbol{q}(\tau)}{d\tau} d\tau. \end{aligned} \tag{5.54}$$

Here we treat $W_{\boldsymbol{f}}$ as an additional state variable, and augment the state-dynamics in (5.34), with

$$\dot{W}_{\boldsymbol{f}} = \boldsymbol{f}^T \dot{\boldsymbol{q}(t)}. \tag{5.55}$$

Integration of (5.55) results in the time evolution of $W_{\boldsymbol{f}}$, which will also incur errors due to numerical integration, and hence must be corrected like $\boldsymbol{q}$, and $\dot{\boldsymbol{q}}$. Similar to the correction for $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$, we consider

$$W_{\boldsymbol{f}}^c = W_{\boldsymbol{f}}^u + \delta W_{\boldsymbol{f}}. \tag{5.56}$$

Due to numerical errors in integration of dynamics and (5.54), (5.53) will not be satisfied. Therefore, the corrections $\delta\boldsymbol{q}$, $\delta\dot{\boldsymbol{q}}$, and $\delta W_{\boldsymbol{f}}$ must be such that (5.53) is satisfied with the corrected quantities $\boldsymbol{q}^c$, $\dot{\boldsymbol{q}}^c$, and $W_{\boldsymbol{f}^c}$, i.e,

$$E(\boldsymbol{q}^c, \dot{\boldsymbol{q}}^c) = E(\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0) + W_{\boldsymbol{f}}^c. \tag{5.57}$$

Substituting $\boldsymbol{q}^c$, $\dot{\boldsymbol{q}}^c$, in $T(\dot{\boldsymbol{q}}^c)$, $V_g(\boldsymbol{q}^c)$, $V_s(\boldsymbol{q}^c)$, and retaining linear terms only, we get

$$T(\dot{\boldsymbol{q}}^c) = T(\dot{\boldsymbol{q}}^u + \delta\dot{\boldsymbol{q}}) \approx T(\dot{\boldsymbol{q}}^u) + \left.\frac{\partial T}{\partial \dot{\boldsymbol{q}}}\right|_{\dot{\boldsymbol{q}}^u} \delta\dot{\boldsymbol{q}} \tag{5.58}$$

$$V_g(\boldsymbol{q}^c) = V_g(\boldsymbol{q}^u + \delta\boldsymbol{q}) \approx V_g(\boldsymbol{q}^u) + \left.\frac{\partial V_g}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}^u} \delta\boldsymbol{q}, \tag{5.59}$$

$$V_s(\boldsymbol{q}^c) = V_s(\boldsymbol{q}^u + \delta\boldsymbol{q}) \approx V_s(\boldsymbol{q}^u) + \left.\frac{\partial V_s}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}^u} \delta\boldsymbol{q}. \tag{5.60}$$

Therefore, (5.57) becomes

$$\begin{bmatrix} \left( \left.\frac{\partial V_g}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}^u} + \left.\frac{\partial V_s}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}^u} \right) & \left.\frac{\partial T}{\partial \dot{\boldsymbol{q}}}\right|_{\dot{\boldsymbol{q}}^u} & -1 \end{bmatrix} \begin{pmatrix} \delta\boldsymbol{q} \\ \delta\dot{\boldsymbol{q}} \\ \delta W_{\boldsymbol{f}} \end{pmatrix} = E(\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0) - E(\boldsymbol{q}^u, \dot{\boldsymbol{q}}^u) + W_{\boldsymbol{f}}^u. \tag{5.61}$$

Combining (5.51) and (5.61), we arrive at a final linear system of equations

$$\underbrace{\begin{bmatrix} \boldsymbol{R_q} & \mathbf{0} & 0 \\ \boldsymbol{Q} & \boldsymbol{R_q} & 0 \\ \left(\frac{\partial V_g}{\partial \boldsymbol{q}}\Big|_{\boldsymbol{q}^u} + \frac{\partial V_s}{\partial \boldsymbol{q}}\Big|_{\boldsymbol{q}^u}\right) & \frac{\partial T}{\partial \dot{q}}\Big|_{\dot{\boldsymbol{q}}^u} & -1 \end{bmatrix}}_{:=\boldsymbol{A}_c} \begin{bmatrix}\delta\boldsymbol{q}\\ \delta\dot{\boldsymbol{q}} \\ \delta W_{\boldsymbol{f}}\end{bmatrix} = \underbrace{\begin{bmatrix} -\boldsymbol{R} \\ -\boldsymbol{R_q}\dot{\boldsymbol{q}}^u \\ E(\boldsymbol{q}_0,\dot{\boldsymbol{q}}_0) - E(\boldsymbol{q}^u,\dot{\boldsymbol{q}}^u) + W_{\boldsymbol{f}}^u\end{bmatrix}}_{:=\boldsymbol{b}_c}, \tag{5.62}$$

and the minimum norm corrections $\delta\boldsymbol{q}$, $\delta\dot{\boldsymbol{q}}$, and $\delta W_{\boldsymbol{f}}$ are determined using pseudoinverse of $\boldsymbol{A}_c$, i.e.

$$\begin{bmatrix}\delta\boldsymbol{q}\\ \delta\dot{\boldsymbol{q}} \\ \delta W_{\boldsymbol{f}}\end{bmatrix} = \boldsymbol{A}_c^T(\boldsymbol{A}_c\boldsymbol{A}_c^T)^{-1}\boldsymbol{b}_c. \tag{5.63}$$

These corrections are done after every integration in each time step once the 2-norm of the constraints violations or the energy violation is greater than a given threshold $\gamma$. Further, since the constraints and energy equation are approximated through linearization, the solution to (5.62) does not fully satisfy the nonlinear equations (5.36) and (5.53) depending on the error coming from the integration. So, an iteration method presented in Algorithm 1 is used to ensure the performance of the corrections.

**Algorithm 1:** An iteration method to minimize the constraints and energy violations.

```
input  : uncorrected terms $\boldsymbol{q}^u$, $\dot{\boldsymbol{q}}^u$ and $W_f^u$
output: corrected terms $\boldsymbol{q}^c$, $\dot{\boldsymbol{q}}^c$ and $W_f^c$
while $\|\boldsymbol{R}\|_2 > \gamma$ or $E - E_0 - W_f > \gamma$ do
    implement (5.63);
    update $\boldsymbol{q}^c$, $\dot{\boldsymbol{q}}^c$ and $W_f^c$ by (5.44),(5.47), (5.56);
    update $\|\boldsymbol{R}\|_2$, $E$ and $W_f$ with corrected term;
end
```

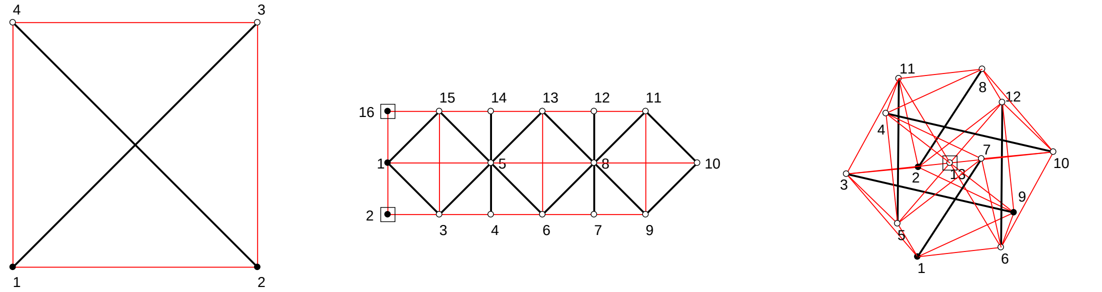
## 5.4 Example

In this section, we model a simple tensegrity structure in 2 dimensions as an example to demonstrate the accuracy of the constraints and motion trajectories using the proposed approach. We also compare our results with those obtained using a commercial tool, i.e. Simscape [110]. In addition, a robotic arm and a ball based on tensegrity structures as two examples are presented to show the efficiency of the method applied to models with higher complexity. In particular, corrected numerical integration is utilized by Matlab to these different models. The equation of motion (5.33) is integrated based on the Dormand-Prince method [134, 135] with relative and absolute tolerances of $10^{-10}$, both constraint correction and energy correction turned on, and the given threshold $\gamma = 10^{-10}$.

First of all, a 2D tensegrity structure built using 2 bars and 4 springs shown in Figure 5.1a is used, where the nodes at the bottom are fixed and the left and right springs are pre-stressed at $90\%$ of the rest length of the springs. Secondly, a robotic arm built from 3 sets of squares is shown in Figure 5.1b, where strings made of nylon are prestressed so that the structure is in equilibrium under gravity. An external force of a time-dependent sinusoidal function is applied vertically to the tip of the arm. Thirdly, a 3D ball with a payload is shown in Figure 5.1c, where 6 bars and 32 strings are used. Here too, strings are prestressed so that the structure can be in equilibrium under gravity. A time-dependent external force of a sinusoidal function is given to the top 3 nodes in different directions, i.e. along x,y,z axes respectively (in order of numbering).

Figure A.1a shows the discrepancies between the motion trajectories obtained using the proposed approach and the minimum realization, where we consider the latter as the benchmark since the equations of motion are derived using generalized coordinates that preserve the geometric constraints and the relative and absolute tolerance of numerical integration is $2.2 \times 10^{-14*}$ and $10^{-14}$ respectively, while Figure A.1b shows the differences between Simscape and minimum realiza-

[*] This is the minimum value of relative tolerance that can be chosen in Matlab.



(a) A 2-bar structure in 2 dimensions.
(b) A robotic arm in 2 dimensions.
(c) A ball in 3 dimensions.

Figure 5.1: Structures of the examples where red lines indicate strings, black lines indicate rigid bars, squares indicate point masses, black dots are fixed nodes, white dots are free nodes, and numbers are node notations.

| Model<br>Attribute | T-Bar | Arm | Ball |
|---|---|---|---|
| Gravity $(m/s^2)$ | 0 | -9.806 | -9.806 |
| Bars: Length $(m)$ | 5 | 1 | 1 |
| Bars: Radius $(m)$ | 0.05 | 0.01 | 0.01 |
| Bars: Density $(kg/m^3)$ | 500 | 1300 | 1300 |
| Springs: Stiffness $(N/m)$ | 100 | - | - |
| Springs: Rest Length Percentage (% of initial) | 90 (vertical only) | - | - |
| Strings: Young's Modulus $(GPa)$ | - | 2 | 2 |
| Strings: Radius $(m)$ | - | 0.001 | 0.001 |
| External Force $(N)$ | 0 | $300sin(t)$ | $300sin(t)$ |

Table 5.1: User-defined properties of the 3 models

tion, where Simscape is with the same numerical method and tolerance as minimum realization. Comparing the figures, we observe that our proposed method produces the motion $10^7$ times closer to the benchmark than Simscape, which indicates a significant improvement in accuracy. Figures A.2a, A.2b and A.3 present the magnitude of constraint violations in bar length and total energy, which shows that the proposed method of constraint correction reduces the violation of the energy to around $10^{-11}$ and keeps the bar length constraint violations at about $10^{-12}$ simultaneously. One can observe that the bar length violations in the simulation produced using Simcape are of a smaller magnitude. This is because it utilizes generalized coordinates, thereby automatically satisfying the bar length constraints. The differences in motion seem to be increasing in time, but in fact, simulating for very long durations would show them to be within bounds, as can also be said by observing the total energy variation. Since the energy is always stable from Figure A.3, the motion must be stable and therefore, the differences are all bounded. The attached video *TEST_TBAR.mp4* demonstrates the motions of the 2-bar structure with 3 approaches in real time.

It is important to note here that, when simulating tensegrity systems with non-minimum coordinates at machine-level precision tolerance settings (at the expense of speed), the user might be tempted to do away with correction altogether. However, the solution, while still being accurate in its motion trajectory for short time lengths, drifts away from the constraint space and consequently, tends to become inaccurate if simulated for long durations. As Yoon[136] points out, it is a necessary condition for accurate simulation that both geometric and energy constraints be satisfied during integration. Hence, it would be advisable to keep the correction algorithm turned on at all times. Figure A.4a shows how accurately the non-minimum formulation performs without the need for correction at the tolerance settings of $10^{-14}$, and the constraint and energy violations as presented in Figure A.4b, A.4c, and A.4d present the extent of the associated constraint drift. Figure A.5 shows the plot for computation times for the T-bar example, simulated at different tolerance settings ranging from $10^{-6}$ to $10^{-14}$. This goes to show that if computation speeds are a higher priority than accuracy, it would be much more prudent to perform simulations at lower tolerance settings like $10^{-9}$ or $10^{-10}$ with correction turned on than to do it at $10^{-14}$ without any correction at all.

To investigate the impact of the energy preservation scheme of the T-bar, we simulated the T-bar example at 2 different tolerance settings ($10^{-6}$ and $10^{-10}$), and at 3 different rest lengths($50\%$, $70\%$, and $90\%$). We found a considerably stronger effect at a higher tolerance setting than at a lower one, for the same rest length $50\%$, as indicated by the order of magnitude of motion errors in Figures A.7a and A.6a respectively. Figures A.6c, A.6d, A.7c and A.7d demonstrate the consistency of the direct constraint correction scheme in stabilizing geometric constraint violations below a specified norm bound, despite a large difference in the order of magnitude of tolerance. In Figure A.8a and A.8b, the T-bar example is simulated at a tolerance of $10^{-10}$ for rest lengths of $50\%$, $70\%$, and $90\%$. Evidently, larger deformations in the $50\%$ case bring energy correction into play more effectively.

For the example of the robotic arm, Figure A.9a shows the motion of node 5, node 8 and node 10 for 20 seconds. Since we've simulated the structure with nylon strings (Young's modulus: 2 GPa), the structure appears to be chattering intermittently. Figures A.10 and A.11 show the constraint violations of bar length and energy. The order of the violations testifies to the stability of the constraints for problems involving intricate geometries. One can observe that the bar length constraints of bars #3, #6, #11 amd #14 are violated more than others in the observed time period and nodes #4,#7,#12,#14 in Figure A.9b are vibrating in higher frequencies, which implies a positive correlation between constraint variations and motion frequencies. Figure A.12a depicts the motion of the 3-dimensional ball which is in accordance with the high stiffness of the strings. Preserving the order of constraint violations as observed in the second example, Figure A.13 and A.12b demonstrate the ability of the implemented correction method to maintain stability of the constraints despite an increase in complexities associated with 3 dimensions. The videos *TEST_ARM.mp4* and *TEST_BALL.mp4* capture the real-time motion of the arm and the ball respectively.

# 6. NUMERICAL EXAMPLE

This chapter will present two different applications of LPV dynamical system control. The first example is a low dimensional autopilot missile system developed in [3], where we apply linear parameter varying quadratic regulator to the system. Two approaches presented in Chapter 3, i.e. in stochastic framework, and Chapter 4, i.e. direct interpolation, are used and compared together with conventional gridding-based LPV control and linear control in terms of performance and the trajectories. The other application is relatively high dimensional tensegrity structures proposed in terms of the dynamical model developed in Chapter 5, where we apply $H_2$ control to a robotic arm. Similarly, the result of direct interpolation with Smolyak algorithm will be shown.

## 6.1 Autopilot Missile System

Consider an autopilot design for a nonlinear missile model [3] using the results presented in the Chapter 3 and 4 and benchmark it with existing techniques. The dynamics of the missile model is given by

$$\dot{\alpha} = K_\alpha M C_n(\alpha, \delta, M)\cos(\alpha) + q, \tag{6.1}$$

$$\dot{q} = K_q M^2 C_m(\alpha, \delta, M), \tag{6.2}$$

where

$$C_n(\alpha, \delta, M) = \alpha\left[a_n|\alpha|^2 + b_n|\alpha| + c_n\left(2 - \frac{M}{3}\right)\right] + d_n\delta,$$
$$C_m(\alpha, \delta, M) = \alpha\left[a_m|\alpha|^2 + b_m|\alpha| + c_m\left(-7 + \frac{8M}{3}\right)\right] + d_m\delta,$$

are the aerodynamic coefficients, $\alpha$ is angle of attack in degrees, $q$ is pitch rate in degrees per second, $\delta$ that denotes control variable is tail deflection angle in degrees, and $M$ is Mach number.

| Parameter | Value | Unit | note |
|---|---|---|---|
| $K_\alpha$ | $0.7\frac{P_0 S}{m v_s}$ | | |
| $K_q$ | $0.7\frac{P_0 S d}{\boldsymbol{I}_y}$ | | |
| $K_z$ | $0.7\frac{P_0 S}{m}$ | | |
| $A_x$ | $0.7\frac{P_0 S C_a}{m}$ | | |
| $P_0$ | 973.3 | $\frac{lbs}{ft^2}$ | static pressure at 20000 $ft$ |
| $S$ | 0.44 | $ft^2$ | surface area |
| $m$ | 13.98 | $slug$ | mass |
| $v_s$ | 1036.4 | $\frac{ft}{s}$ | speed of sound at 20000 $ft$ |
| $d$ | 0.75 | $ft$ | diameter |
| $\boldsymbol{I}_y$ | 182.5 | $slug \cdot ft^2$ | pitch moment of inertia |
| $C_a$ | $-0.3$ | | drag coefficient |
| $\zeta$ | 0.7 | | actuator damping ratio |
| $\omega_a$ | 150 | $\frac{rad}{s}$ | actuator undamped natural frequency |
| $a_n$ | 0.000103 | $deg^{-3}$ | |
| $b_n$ | $-0.00945$ | $deg^{-2}$ | |
| $c_n$ | $-0.1696$ | $deg^{-1}$ | |
| $d_n$ | $-0.034$ | $deg^{-1}$ | |
| $a_m$ | 0.000215 | $deg^{-3}$ | |
| $b_m$ | $-0.0195$ | $deg^{-2}$ | |
| $c_m$ | 0.051 | $deg^{-1}$ | |
| $d_m$ | $-0.206$ | $deg^{-1}$ | |

Table 6.1: Coefficients of missile model. [3]

The system parameters are defined in Table 6.1. The objective is to design a full state feedback controller $\boldsymbol{K}(\boldsymbol{\rho})$ that stabilizes the missile system such that $-20deg \leq \alpha \leq 20deg$ and $|\dot{\alpha}| \leq 10deg/sec$ while minimizing the cost-to-go function with

$$\boldsymbol{Q} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}, \quad \boldsymbol{R} = 1.$$

### 6.1.1 Univariate Case

We first consider $M = 2.5$ to be a constant for simplicity, and transform the nonlinear dynamics to a quasi-LPV system by introducing $\rho := \alpha$,

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} K_\alpha M \left[a_n|\boldsymbol{\rho}|^2 + b_n|\boldsymbol{\rho}| + c_n\left(2 - \frac{M}{3}\right)\right]\cos(\boldsymbol{\rho}) & 1 \\ K_q M^2 \left[a_m|\boldsymbol{\rho}|^2 + b_m|\boldsymbol{\rho}| + c_m\left(-7 + \frac{8M}{3}\right)\right] & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} K_\alpha M d_n \cos\boldsymbol{\rho} \\ K_q M^2 d_m \end{bmatrix} \delta. \tag{6.3}$$

$$= \boldsymbol{A}(\rho)\boldsymbol{x} + \boldsymbol{B}(\rho)\delta \tag{6.4}$$

5 types of control syntheses are presented as follows

- **Globally Optimal Control**: We design classic LQR for the dynamics (6.4) at every instant time and feedback to the missile system. Therefore, the control gain $\boldsymbol{K}_{\text{LTI}}$ is designed in real time. This is considered as a benchmark design.

- **Polynomial Chaos**: The parameter $\boldsymbol{\rho}$ is considered as random variable uniformly distributed over $[-20, 20]$, so we define $\boldsymbol{\rho} \in \mathcal{U}_{[-20,20]}$. From Theorem 22, the controller $\boldsymbol{K}_{\text{PC}}$ is obtained with different order polynomial chaos expansions.

- **Stochastic Collocation**: Similar with $\boldsymbol{K}_{\text{PC}}$, the controller $\boldsymbol{K}_{\text{SC}}$ is obtained with different polynomial chaos expansions by solving the optimization problem in Theorem 23.

- **Conventional Gridding Method**: According to the LPV system (6.4), we set $\boldsymbol{Y}_{\text{LPV}}(\rho) := \boldsymbol{Y}_0 + \sum_{i=1}^3 f_i(\rho)\boldsymbol{Y}_i > 0, \boldsymbol{Y}_i = \boldsymbol{Y}_i^T$, and $\boldsymbol{W}_{\text{LPV}}(\rho) := \boldsymbol{W}_0 + \sum_{i=1}^3 f_i(\rho)\boldsymbol{W}_i$, and the controller $\boldsymbol{K}_{\text{LPV}}(\rho) = \boldsymbol{W}_{\text{LPV}}(\rho)\boldsymbol{Y}_{\text{LPV}}^{-1}(\rho)$ is obtained by solving the optimization problem below with $N$ sample points equidistant in $[-20, 20]$, where $f_i(\rho)$ are user defined functions,

| Controller | Synthesis Time (sec) | $J$ |
|---|---|---|
| $\boldsymbol{K}_{\text{LTI}}$ |  | 103.8635 |
| $\boldsymbol{K}_{\text{PC}}$ ($3^{rd}$ order PC) | 17.9557 | 104.0577 |
| $\boldsymbol{K}_{\text{PC}}$ ($4^{th}$ order PC) | 133.0689 | 104.0129 |
| $\boldsymbol{K}_{\text{PC}}$ ($5^{th}$ order PC) | 176.9519 | 103.9172 |
| $\boldsymbol{K}_{\text{SC}}$ ($5^{th}$ order SC) | 0.6864 | 108.9585 |
| $\boldsymbol{K}_{\text{SC}}$ ($9^{th}$ order SC) | 1.3416 | 107.3185 |
| $\boldsymbol{K}_{\text{SC}}$ ($12^{th}$ order SC) | 1.7628 | 104.1266 |
| $\boldsymbol{K}_{\text{LPV}}$ (10 samples) | 1.9812 | 118.5069 |
| $\boldsymbol{K}_{\text{LPV}}$ (100 samples) | 11.8561 | 118.7867 |
| $\boldsymbol{K}_{\text{LPV}}$ (500 samples) | 55.9108 | 118.5196 |
| $\boldsymbol{K}_{\text{SG}}$ ($k = 3\ \alpha_i = 0.15$) | 0.4281 | 281.2174 |
| $\boldsymbol{K}_{\text{SG}}$ ($k = 6\ \alpha_i = 0.08$) | 0.5956 | 231.8533 |
| $\boldsymbol{K}_{\text{SG}}$ ($k = 9\ \alpha_i = 0.1$) | 9.0174 | 240.6945 |

Table 6.2: Comparison of controller performances and synthesis times.

i.e. $f_1 := \rho$, $f_2 := \rho^2$ and $f_3 := \cos\rho$ in this example.

$$\max_{\boldsymbol{Y}_0,\boldsymbol{Y}_1,\boldsymbol{W}_0,\boldsymbol{W}_1} \ \mathbf{tr}\ (\boldsymbol{Z}_{LPV}),\ \text{subject to}$$

$$\begin{bmatrix} \mathsf{sym}\left(\boldsymbol{Y}_{\text{LPV}}(\rho_k)\boldsymbol{A}^T(\rho_k) + \boldsymbol{W}^T_{\text{LPV}}(\rho_k)\boldsymbol{B}^T(\rho_k)\right) & \boldsymbol{Y}_{\text{LPV}}(\rho_k) & \boldsymbol{W}^T_{\text{LPV}}(\rho_k) \\ \boldsymbol{Y}_{\text{LPV}}(\rho_k) & -\boldsymbol{Q}^{-1} & \mathbf{0} \\ \boldsymbol{W}_{\text{LPV}}(\rho_k) & \mathbf{0} & -\boldsymbol{R}^{-1} \end{bmatrix} \leq 0,$$

$$\boldsymbol{Y}_{LPV}(\rho_k) - \boldsymbol{Z}_{LPV} \geq 0,$$

for $k = 1, \cdots, N$ and $\boldsymbol{Z}_{LPV} > 0$.

- **Direct Interpolation**: The controller $\boldsymbol{K}_{SG} = \boldsymbol{W}_{SG}(\boldsymbol{\rho})\boldsymbol{Y}^{-1}_{SG}(\boldsymbol{\rho})$ is obtained by solving the optimization problem stated in (4.28)-(4.32) and implementing the algorithm in the Section 4.3.1. The coefficients of $\boldsymbol{Y}_{SG}$ and $\boldsymbol{W}_{SG}(\boldsymbol{\rho})$ are determined by (4.26) and (4.27) respectively, and then we can formulate the interpolation matrices from (4.15) and (4.16).

Table 6.2 compares the value of cost function computed from (2.33) and computation time in the respective optimization problem. Undoubtedly, $\boldsymbol{K}_{\text{LTI}}$ has the lowest cost function and it is obvious to see that $\boldsymbol{K}_{\text{PC}}$, which gives a cost function close to $\boldsymbol{K}_{\text{LTI}}$, dominates the others, but $\boldsymbol{K}_{\text{SC}}$ has similar performance with much less computational cost.

However, by checking the negative definiteness of (2.40) with the solution $\boldsymbol{Y}_{\text{PC}}(\rho)$ and $\boldsymbol{W}_{\text{PC}}(\rho)$, or $\boldsymbol{Y}_{\text{SC}}(\rho)$ and $\boldsymbol{W}_{\text{SC}}(\rho)$, for a dense grid, we found that they do not really satisfy the sufficient condition. This is due to the difference of the problem setting. The polynomial chaos expansion and stochastic collocation are used in the stochastic framework, where we opt to satisfy the expectation condition (3.21), while the original LPV problem is deterministic and without expected values. In other words, even though the controller designed in the stochastic framework yields a better performance in this example, it is not guaranteed to have such good result for other case due to the lack of analytical relationship between (3.21) and (2.35). After all, (3.21) and (2.35) are only sufficient conditions that are not necessarily satisfied to get a controller having good performance.

Regarding the controllers based on conventional gridding method and sparse grid interpolation, Table 6.2 shows that they are both computationally efficient. This is because the example system is a lower dimensional problem, where the dimensionality issue of the gridding method is imperceptible. If the dimension of the parameter set is much greater than 1, it is almost impossible to implement gridding approach. Although the gridding method performs better than sparse grid based interpolation with respect to the cost function $J$ as shown in Table 6.2, the feasibility of the condition (2.40) may not satisfy. Fig 6.1 shows, for $\dot{\rho} = 10$, the amount of checking points, where (2.40) is infeasible, in the number of samples used for gridding method, which implies that the feasibility region is growing up as we increase the amount of sample points. However, this is again impractical for high dimensional problem.

On the other hand, we checked the infeasible region of (2.40) with Smolyak algorithm based approach and the conditions are all satisfied even for low level $k$ with less sample points. However, comparing with other types of controller, the sparse grid based control does not show any advantage in terms of the cost function index $J$. We infer that the sparse grid based control does not render

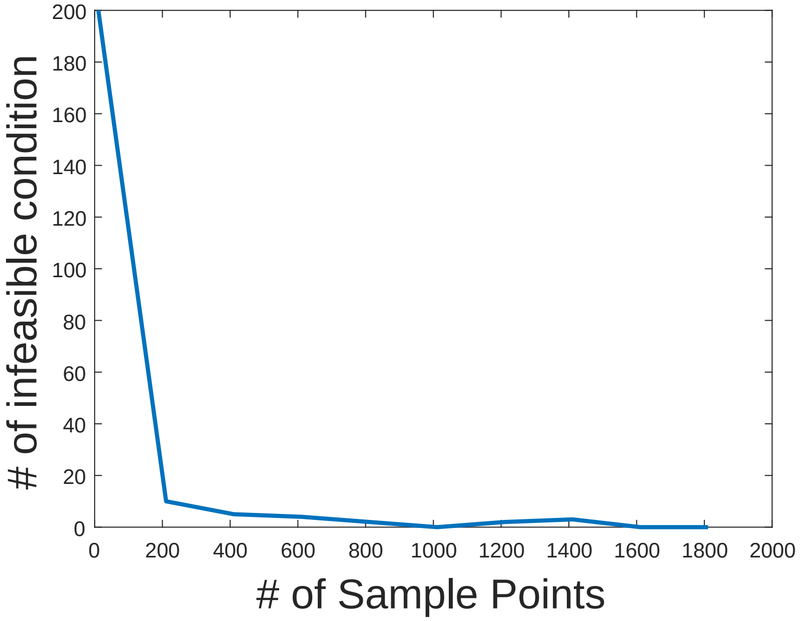


Figure 6.1: # of infeasible condition of (2.40),which is measured in 2000 points between -10 and 10, versus # of sample points for controller synthesis.

better performance than others but it easily guarantees the sufficient conditions for the stability and performance.

Fig.(B.1) to Fig.(B.4) plot the trajectories of states and control for different orders of their own types of controllers. The plots look similar among different orders which accord with the cost index $J$ in the Table 6.2. Fig.(B.5) compares the trajectories with different types of controller, where the state trajectories does not have much difference but the value of control effort for each controller does agree with the performance trend shown in the Table 6.2. Finally, Fig.(B.6) shows the trajectories of the rate of angle of attack implying that the required bounds are achieved.

### 6.1.2 Multivariate Case

We extend the previous example to a multivariate LPV system by letting the Mach number varying between 2 and 4 and $|\dot{M}| \leq 1$. Therefore, we let $\boldsymbol{\rho} := [\alpha, M]^T$ and the same LPV model as (6.4) is used. In order to have a reasonably realistic Mach trajectory, we set the Mach by

$$\dot{M}(t) = \frac{1}{v_s}\left[-|\eta(t)|\sin(|\alpha(t)|) + A_x M^2(t)\cos(\alpha(t))\right]$$

with $M(0) = 3$, where

$$\eta = K_z \boldsymbol{M}^2 \left[a_n\alpha^2 + b_n|\alpha| + c_n(2 - \frac{M}{3})\right]\cos(\alpha) + K_\alpha M d_n \cos(\alpha)\delta$$

Similarly, we design 5 types of controller accordingly as follows.

- **Linear Control**: The design is exactly the same as shown in the previous section since it is independent of parameters.

- **Polynomial Chaos**: The parameter $\rho$ is considered as a set of random variables uniformly distributed over $[-20, 20] \times [2, 4]$, so we define $\boldsymbol{\rho} \in \mathcal{U}_{[-20,20]\times[2,4]}$ From the Theorem 22, the controller $\boldsymbol{K}_{\text{pc}}$ is obtained with different order polynomial chaos expansions.

- **Stochastic Collocation**: Similar with $\boldsymbol{K}_{\text{PC}}$, the controller $\boldsymbol{K}_{\text{SC}}$ is obtained with different polynomial chaos expansions by solving the optimization problem in Theorem 23.

- **Conventional Gridding Method**: According to the LPV system (6.4), we set $\boldsymbol{Y}_{\text{LPV}}(\boldsymbol{\rho}) := \boldsymbol{Y}_0 + \sum_{i=1}^{5} f_i(\rho)\boldsymbol{Y}_i > 0, \boldsymbol{Y}_i = \boldsymbol{Y}_i^T$, and $\boldsymbol{W}_{\text{LPV}}(\rho) := \boldsymbol{W}_0 + \sum_{i=1}^{5} f_i(\rho)\boldsymbol{W}_i$, and the controller $\boldsymbol{K}_{\text{LPV}}(\rho) = \boldsymbol{W}_{\text{LPV}}(\rho)\boldsymbol{Y}_{\text{LPV}}^{-1}(\rho)$ is obtained by solving the same optimization problem in the previous section with $N^2$ grid points in the sample space $[-20, 20] \times [2, 4]$, where $f_i(\rho)$ are user defined functions, i.e. $f_1 := M\rho$, $f_2 := M\rho^2$, $f_3 := M\cos\rho$, $f_4 := M$ and $f_5 := M^2$ in this example.

- **Direct Interpolation**: The method to obtain $\boldsymbol{K}_{\text{SG}}$ is the same as the previous section, where the grid points are chosen based on the Smolyak algorithm. We set the level of sparse grid to $k = 8$ for 1537 grid points and the level $k = 3$ for 29 basis functions. The constants in the constraints are $\alpha_i = 0.15$.

| Controller | synthesis Time | $J$ |
|---|---|---|
| $\boldsymbol{K}_{\text{LTI}}$ |  | 114.2306 |
| $\boldsymbol{K}_{\text{pc}}$ (3rd order) | 6043.4 | 236.8949 |
| $\boldsymbol{K}_{\text{sc}}$ (3rd order) | 0.3350 | 122.0233 |
| $\boldsymbol{K}_{\text{sc}}$ (5th order) | 5.1995 | 116.7951 |
| $\boldsymbol{K}_{\text{sc}}$ (7th order) | 97.2259 | 115.7800 |
| $\boldsymbol{K}_{\text{LPV}}$ (10^2 samples) | 19.0789 | 208.5167 |
| $\boldsymbol{K}_{\text{LPV}}$ (20^2 samples) | 86.0034 | 213.5322 |
| $\boldsymbol{K}_{\text{LPV}}$ (30^2 samples) | 212.6450 | 217.5059 |
| $\boldsymbol{K}_{\text{SG}}$ ($k = 8$) | 17.5620 | 203.2555 |

Table 6.3: Comparison of controller performances and synthesis times for multivariate case.

In the 2-dimension case, the Table 6.3 evidently shows that the computational cost of polynomial chaos expansion is significantly increasing even it is only 3rd order expansion. This can be realized by analyzing the dimension of the LMI in (3.48), the PC based approach is hence not suitable for multivariate problems. The stochastic collocation, on the other hand, shows its advantage on the computational efficiency and its performance close to the benchmark design. Since the system in this example has explicit and simple expressions, one can easily find basis functions for gridding method. However, it is still an open question about the choices of basis functions, so the performance of the gridding method is varying and unpredictable. Regarding the sparse grid based controller, its performance is not remarkable in terms of the cost function $J$ in the Table 6.3 because of the choice of basis function and the constant $\alpha_i$ introduced in the algorithm, but the synthesis time is speedy.

With respect to the feasibility of the constraint for the parameter sets, Fig.(B.7) to Fig.(B.9) show the feasibility region of each controller, where only $\boldsymbol{K}_{\text{SG}}$ satisfies the conditions for all the elements in the parameter set. Therefore, $\boldsymbol{K}_{\text{SG}}$ has less region to find an optimal solution while other controllers are synthesized by searching an optimal solution in a larger region,which contains both infeasible and feasible region. Again, since the conditions (2.40) are only sufficient, a controller that does not satisfies the constraints for all the elements in the parameter sets may still stabilize the system and even performs better, although it is not guaranteed.

In fact, the constants $\alpha_i$ and $\beta_i$ in the constraint can be applied to conventional gridding method as well, so that the $\boldsymbol{K}_{\text{LPV}}$ can satisfy the conditions for whole parameter space and the conservatism is introduced. Fig.(B.10) shows the feasibility region where the amount of sample points is the same as we use in Fig.(B.8).

Fig.(B.11) and Fig.(B.12) plot the states and control trajectories for stochastic collocation based and gridding based control respectively, which show very similar result for different orders or the number of sample points. Fig.(B.13) compares all controllers together and it shows the sparse grid based control yields faster stabilization but large control effort so that the cost function $J$ is high among these controllers as shown in the Table 6.3. Finally, Fig.(B.14) and Fig.(B.15) show the trajectories of the rate of angle of attack and Mach number implying that the required bound are achieved.

## 6.2 Tensegrity Structure

This section will demonstrate two applications based on tensegrity structures achieving some desired locomotion under disturbances, so intuitively an $H_2$ control will be utilized to the control system. The mathematical model and the associated simulation used in this section was developed in Chapter 5, where a minimum set of coordinates used. In this section, we only focus on sparse grid based method, since the dimensionality is too high to implement either stochastic frameworks or conventional gridding method.

### 6.2.1 Robotic Arm

Fig.(6.2) shows the configuration of a 2 dimensional tensegrity structure, where node 1 is assigned to be the origin $(0, 0)$, and the objective is to let the tip of the arm track two desired trajectories, i.e. two circles intersecting with each other,where the disturbance and noise rejections are achieved.

The model is of the form

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{A}(\boldsymbol{\rho})\boldsymbol{x} + \boldsymbol{B}_{\boldsymbol{u}}(\boldsymbol{\rho})\boldsymbol{u} + \boldsymbol{B}_{\boldsymbol{w}}(\boldsymbol{\rho})\boldsymbol{w} \\
\boldsymbol{y} &= \boldsymbol{C}_{\boldsymbol{y}}\boldsymbol{x} + \boldsymbol{n} \\
\boldsymbol{z} &= \boldsymbol{C}_{\boldsymbol{z}}\boldsymbol{x} + \boldsymbol{D}_{\boldsymbol{u}}(\boldsymbol{\rho})\boldsymbol{u},
\end{aligned}
$$

where $\boldsymbol{\rho} := \boldsymbol{x}$, $\boldsymbol{x} = [\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i]^T$ with a minimum set of coordinates $\boldsymbol{q}_i$, the system matrices are derived in Chapter 5 according to the configuration in Fig.(6.2), $\boldsymbol{u} \in \mathbb{R}^{10}$ is control variables indicating the force density of the cables, $\boldsymbol{y}$ is the measurement, $\boldsymbol{n}$ denoting sensor noise is the zero-mean white noise, $\boldsymbol{w}$ denoting external disturbance is zero-mean white noise as well, and $\boldsymbol{z}$ denoting the desire output contains the states and the control variables. It is obvious to see that the system has 6 degrees of freedom, so $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{x}} \subset \mathbb{R}^{12}$, and we assign a minimum set of coordinates as

$$
\boldsymbol{q}_i = \begin{bmatrix} x_4 & x_6 & y_6 & x_8 & x_{11} & y_{11} \end{bmatrix}.
$$

In particular, the desire output contains

$$
\boldsymbol{z} = \begin{bmatrix} x_4 & 20x_6 & 20y_6 & x_8 & x_{11} & y_{11} & \dot{x}_4 & \dot{x}_6 & \dot{y}_6 & \dot{x}_8 & \dot{x}_{11} & \dot{y}_{11} & \frac{\boldsymbol{u}}{200} \end{bmatrix}^T,
$$

since we opt to keep the nodes as close to the reference trajectories as possible, especially the tip of the structure described by $x_6$ and $y_6$, and relax the control effort. In addition, we consider two accelerometers installed on the node 6 and node 11 to measure the accelerations in both x and y directions, and two rotary encoders attached on the node 11 to measure the positions of the node 4 and node 8 based on the angles. Besides, we assume that all disturbances and sensor noises are uncorrelated and

$$
\mathbf{E}\left[\boldsymbol{w}(t+\tau)\boldsymbol{w}^T(t)\right] = \boldsymbol{Q}\delta(\tau), \quad \mathbf{E}\left[\boldsymbol{n}(t+\tau)\boldsymbol{n}^T(t)\right] = \boldsymbol{R}\delta(\tau)
$$

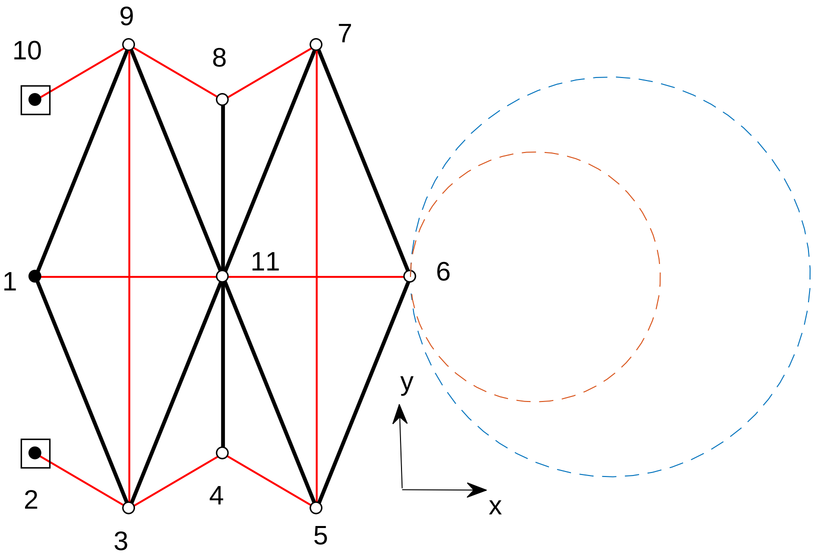


Figure 6.2: The configuration of the system at initial position where black dots denote fixed points, black lines denote rigid bars, red lines denote cables, the numbers index the nodes and the dotted circles are desired trajectories.

where $\boldsymbol{Q} = 18.75\boldsymbol{I}$ and $\boldsymbol{R} = 0.0033\boldsymbol{I}$, because the disturbances are assumed to be uniformly distributed between -7.5 and 7.5 and the noises are uniformly distributed between -0.1 and 0.1.

Based on the control objective, we utilize $H_2$ control to the system, which was developed in the Theorem 13 and 14, and by using extended Kalman filter, which is a famous state estimator for nonlinear systems, we can estimate the full states $\hat{\boldsymbol{x}}$ and apply it to the feedback control. Since the the parameter set is in 12-dimensional, it is impossible to apply the method based on stochastic framework and conventional gridding method for such a high dimensional problem.

Therefore, the Smolyak algorithm introduced in Chapter 4 and revealing a benefit of significant grid points reduction is used to solve the $H_2$ problem. More specifically, the optimization problem (4.33) to (4.34) are solved based on the algorithm in the Section 4.3.1 and the controller $\boldsymbol{K}(\boldsymbol{\rho}) = \boldsymbol{W}(\Psi_m^T(\boldsymbol{\rho})\boldsymbol{z}_{SG})^{-1}$, where $\boldsymbol{z}_{SG}$ is obtained from (4.36).

Regarding the tracking problem, we consider a time based circular trajectory shown in Fig.(6.2), where the mathematical description of the trajectories for node 6 is

$$\boldsymbol{q}_r(t) = \begin{cases} \begin{bmatrix} 2 - 0.5\cos(t) \\ 0.5\sin(t) \end{bmatrix} & \text{for small circle} \\ \begin{bmatrix} 2.3 - 0.8\cos(t) \\ 0.8\sin(t) \end{bmatrix} & \text{for large circle} \end{cases} \tag{6.5}$$

We let the node 6 tracks the small circle first starting from the initial position and then change it to large circle and have a "small-large-small" loop. Accordingly, the trajectories for other nodes can be formulated similar to (6.5) but in different location.

Another concern is the physical limits of the cable which does not sustain compression, so we need to set up a pre-stress condition for the system and ensure the cables always work in tension during the tracking motion. In particular, consider a trajectory $\boldsymbol{x}_r(t)$ that satisfies

$$\dot{\boldsymbol{x}}_r(t) = \boldsymbol{A}(\boldsymbol{\rho}_r)\boldsymbol{x}_r + \boldsymbol{B}(\boldsymbol{\rho}_r)\boldsymbol{u}_r,$$

then we can solve the equation with the pre-stress constraint $\boldsymbol{u}_r \geq \boldsymbol{u}_p$ where $\boldsymbol{u}_p$ is given by 50 in the example. Moreover, the total control effort becomes $\boldsymbol{u} = \boldsymbol{K}(\boldsymbol{\rho})(\hat{\boldsymbol{x}} - \boldsymbol{x}_r) + \boldsymbol{u}_r$, i.e. the reference control for tracking problem plus the $H_2$ control for disturbance and noise rejection.

Fig.(B.16) shows the motion trajectory of the node 6 with the given control $\boldsymbol{u}$ developed above, where the actual output tracks the reference very well except for the motion close to trajectory transition. Fig.(B.17) plots the control effort of the system, which shows that the control does not violate the physical rule of the cable, and the high value of the control in Fig.(B.17) is due to the needs for pulling the arm forward and backward.

We have also checked the feasibility of the constraints stated in the Theorem 13 with the solution to the optimization problem (4.33)- (4.34) and the solution actually does not satisfy the constraints, although the tracking control problem is solved. Again, the constraints are sufficient condition, so it is not necessary to ensure the feasibility. If one can find a solution satisfies all the constraints, it is always guaranteed to stabilize the system and achieve the performance. Increasing the level of Smolyak algorithm may achieve the feasibility for all the element in the parameter set. However, it is not practical for a high-dimensional problem because the extremely large amount of LMIs needs to be solved simultaneously according to (4.8). Observing all kinds of LPV control including the existing controls and the proposed control methods, we remark that sparse grid based methodology is the most suitable one to handle problems with high dimensions.

# 7. CONCLUSION

In this thesis, we first give a brief overview of LPV systems. This includes the reasons on the LPV system development, the benefit of LPV control rather than the gain-scheduling approach. We have survey some notable research about LPV systems including modeling and control design. The goal of modeling is to transform an original nonlinear system to a standard LPV or quasi-LPV model by either approximation methods or function substitutions. Different kinds of LPV control has been reviews and discuss the benefits and the drawbacks of each method, where the gridding approach is the main direction we are focusing on in the thesis. We have also discuss the needs and potential of the dynamical systems described in term of non-minimum set of coordinates and the gridding based LPV control design.

Regarding the stability of the LPV systems, we have introduced quadratic stability and robust stability and developed the associated conditions that needs to be satisfied such that the stability is guaranteed. Other than the stability of the LPV systems, the $H_2$ performance has also been investigated, where a set of LMIs has been formulated so that the $H_2$ is bounded. In particular, parameter-independent or parameter-dependent unknown matrices are solved so that the system is guarantted to be stable and achieve the $H_2$ performance, where parameter-dependent matrices yield less conservatism. We have further investigated the optimal LPV control systems in terms of the cost-to-go function and the $H_2$ norm of the system. More specifically, a linear quadratic regular for LPV systems has been developed such that the cost function is optimized and an $H_2$ control, which guarantees the $H_2$ norm performance of the control system, has also been investigated. Both methods yields a parameter-dependent optimization which is convex if the parameter is fixed. Clearly, there is a tradeoff between problem size and conservatism in the design.

In the thesis, inspired by Fujisaki [55], we presented a new framework for synthesizing LPV controllers using polynomial chaos framework. This framework builds on the probabilistic representation of the scheduling variables. The parameter is treated as a random variable with a given distribution, so we consider the expected value of the cost function instead of the original one, which is the key difference from the conventional LPV control design. We have taken this approach to develop new algorithms for synthesis of linear quadratic regulators. We pursued two approaches: polynomial chaos expansion and stochastic collocation to develop the necessary theoretical framework, which results in a convex optimization problem.

Other than stochastic framework, we have studied the conventional LPV approach, i.e. gridding method, which requires extremely large amount of samples to ensure the feasibility of the constraints for either system analysis or controller design. Due to this drawback, we have developed a series of controller syntheses and an algorithm to solve the parameter-dependent optimization problems. The key mechanism is the sparse grid based on Smolyak algorithm that significantly reduces the number of sample points and increase the efficiency of the controller synthesis. This is one of the main results in the thesis.

In addition to controller design, the equations of motion for tensegrity structures were developed in Cartesian coordinates, i.e. a nonminimum set of coordinates, using Lagrangian mechanics. The equations of motion have further been transformed into a quasi-LPV model by using coordinate partitioning method. With respect to the simulation of constrained dynamics, since states at the position and velocity levels may drift away from the constraint space due to numerical computations, the direct correction approach was used so that the violations in both geometric constraints and total mechanical energy are eliminated. A simple tensegrity structure has been simulated with different parameters to show the improvement of the accuracy by using the proposed approach. Besides, we have shown that the differences between numerical simulation performed with and without correction become more prominent when a structure has larger motion or when the error of numerical integration is increasing. We have also simulated tensegrity structures with significantly more complex geometries to demonstrate the consistency of the proposed method towards constraint stabilization.

To evaluate the proposed method stated in this thesis we have shown examples of a missile system and a tensegrity based robotic arm that are controlled by LPV controllers. In the example of the missile system, we have compared the optimality of each LQR controller, i.e. polynomial chaos expansion, stochastic collocation, convectional gridding method, and Smolayk algorithm. In addition, the feasibility of the constraints from different controllers implies that Smolyak algorithm based control is able to satisfy the sufficient condition that guarantees the stability and performance but the actual cost function is not the optimal one among others. We have discussed some explanations for this results. Another example of tensegrity system is a high dimensional problem, so only sparse grid method is suitable due to the limits of the computation for controller synthesis. Although the feasibility does not hold, the simulation results have shown good performance of a tracking problem.

It is known that the algorithm stated in the Chapter 4 is just ad-hoc. There still exist many open questions in it. For example, there is no theoretical analysis to choose the approximation level and relaxation of the constraints so that the parameter-dependent conditions can hold for sure. A possible solution is to analyze the upper bound of the approximation error and utilize the bound to relax the constraint. This is similar to the ad-hoc algorithm but the choices of the associate relaxation coefficients are determined automatically depending on the level of the Smolyak algorithm. The related research on error analysis of Smolyak algorithm is found in [90, 94]. Another future challenge is the curse of dimensionality. It is known that the computation cost hikes up significantly if the dimension raise too much. Even for Sparse grid method, the number of the grid points will sill be very huge that the solution to the problem becomes impractical due to the limitation of the numerical computation. Therefore, the investigation on this topic will be a benefit for a large size system.

# REFERENCES

[1] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback control theory*. Courier Corporation, 2013.

[2] D. Xiu and G. Karniadakis, "The Wiener–askey polynomial chaos for stochastic differential equations," *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 619–644, 2002.

[3] Fen Wu, A. Packard, and G. Balas, "LPV control design for pitch-axis missile autopilots," *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol. 1, no. December, pp. 188–193, 1995.

[4] R. A. Nichols, R. T. Reichert, and W. J. Rugh, "Gain scheduling for h-infinity controllers: a flight control example," *IEEE Transactions on Control Systems Technology*, vol. 1, pp. 69–79, June 1993.

[5] H. Zhang, X. Zhang, and J. Wang, "Robust gain-scheduling energy-to-peak control of vehicle lateral dynamics stabilisation," *Vehicle System Dynamics*, vol. 52, no. 3, pp. 309–340, 2014.

[6] B. W. Bequette, "Nonlinear control of chemical processes: a review," *Industrial & Engineering Chemistry Research*, vol. 30, no. 7, pp. 1391–1413, 1991.

[7] J. S. Shamma and M. Athans, "Gain scheduling: potential hazards and possible remedies," *IEEE Control Systems Magazine*, vol. 12, pp. 101–107, June 1992.

[8] J. S. Shamma, *Anaylsis and Design of Gain Scheduled Control Systems*. PhD thesis, Massachusetts Institute of Technology, 1988.

[9] R. Tóth, *Modeling and Identification of Linear Parameter-Varying Systems*, vol. 403 of *Lecture Notes in Control and Information Sciences*. Berlin, Heidelberg: Springer Berlin Heidelberg, dec 2010.

[10] A. Marcos and G. Balas, “Linear parameter varying modeling of the Boeing 747-100/200 longitudinal motion,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (Reston, Virigina), pp. 1–11, American Institute of Aeronautics and Astronautics, aug 2001.

[11] R. Tóth and D. Fodor, “Speed sensorless mixed sensitivity linear parameter variant h$\infty$ control of the induction motor,” *Journal of Electrical Engineering*, vol. 6, no. 4, p. 12, 2006.

[12] P. Gáspár, Z. Szabó, and J. Bokor, “A grey-box identification of an LPV vehicle model for observer-based side slip angle estimation,” *Proceedings of the American Control Conference*, pp. 2961–2966, 2007.

[13] J. S. SHAMMA and J. R. CLOUTIER, “Gain-scheduled missile autopilot design using linear parameter varying transformations,” *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 2, pp. 256–263, 1993.

[14] D. J. Leith and W. E. Leithead, “Comments On the Prevalence of Linear Parameter Varying Systems,” tech. rep., Department of Electronic & Electrical Engineering, University of Strathclyde, 1999.

[15] M. Lovera, M. Bergamasco, and F. Casella, *Robust Control and Linear Parameter Varying Approaches*, vol. 437 of *Lecture Notes in Control and Information Sciences*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[16] G. Papageorgiou, *Robust Control System Design : H$\infty$ Loop Shaping and Aerospace Applications*. PhD thesis, University of Cambridge, 1998.

[17] A. Marcos and G. J. Balas, “Development of Linear-Parameter-Varying Models for Aircraft,” *Journal of Guidance, Control, and Dynamics*, vol. 27, pp. 218–228, mar 2004.

[18] J.-Y. Shin and C. Belcastro, “Quasi-Linear Parameter Varying Representation of General Aircraft Dynamics Over Non-Trim Region,” tech. rep., NASA Langley Research Center, Hampton, VA, 2007.

[19] J.-Y. Shin, G. J. Balas, and M. A. Kaya, "Blending Methodology of Linear Parameter Varying Control Synthesis of F-16 Aircraft System," *Journal of Guidance, Control, and Dynamics*, vol. 25, pp. 1040–1048, nov 2002.

[20] J.-Y. Shin, *Worst-case analysis and linear parameter-varying gain-scheduled control of aerospace systems*. PhD thesis, University of Minnesota, 2000.

[21] R. Bhattacharya, G. J. Balas, M. A. Kaya, and A. Packard, "Nonlinear Receding Horizon Control of an F-16 Aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 25, pp. 924–931, sep 2002.

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

[23] G. J. Balas, J. C. Doyle, K. Glover, A. Packard, and R. Smith, "$\mu$-analysis and synthesis toolbox," *MUSYN Inc. and The MathWorks, Natick MA*, 1993.

[24] P. Apkarian and P. Gahinet, "A convex characterization of gain-scheduled H/sub $\infty$/ controllers," *IEEE Transactions on Automatic Control*, vol. 40, pp. 853–864, may 1995.

[25] A. Packard, "Gain scheduling via linear fractional transformations," *Systems & Control Letters*, vol. 22, pp. 79–92, feb 1994.

[26] G. Scorletti and L. E. Ghaoui, "Improved LMI conditions for gain scheduling and related control problems," *International Journal of Robust and Nonlinear Control*, vol. 8, pp. 845–877, aug 1998.

[27] C. Scherer, "A full block S-procedure with applications," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 3, pp. 2602–2607, IEEE, nov 1997.

[28] C. W. Scherer, "Mixed h2/h$\infty$ control for time-varying and linear parametrically-varying systems," *International Journal of Robust and Nonlinear Control*, vol. 6, pp. 929–952, nov 1996.

[29] C. Scherer, “LPV control and full block multipliers,” *Automatica*, vol. 37, pp. 361–375, mar 2001.

[30] P. Gahinet, P. Apkarian, and M. Chilali, “Affine parameter-dependent Lyapunov functions and real parametric uncertainty,” *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 436–442, 1996.

[31] P. Apkarian, P. Gahinet, and G. Becker, “Self-scheduled H$\infty$ control of linear parameter-varying systems: a design example,” *Automatica*, vol. 31, no. 9, pp. 1251–1261, 1995.

[32] J. Daafouz and J. Bernussou, “Parameter dependent lyapunov functions for discrete time systems with time varying parametric uncertainties,” *Systems & control letters*, vol. 43, no. 5, pp. 355–359, 2001.

[33] I. E. Kose, F. Jabbari, and W. E. Schmitendorf, “A direct characterization of l/sub 2/-gain controllers for lpv systems,” *IEEE Transactions on Automatic Control*, vol. 43, pp. 1302–1307, Sep. 1998.

[34] W. Xie, “H2 gain scheduled state feedback for LPV system with new LMI formulation,” *IEE Proceedings - Control Theory and Applications*, vol. 152, pp. 693–697, nov 2005.

[35] F. Wu, X. H. Yang, A. Packard, and G. Becker, “Induced L2-norm control for LPV systems with bounded parameter variation rates,” *International Journal of Robust and Nonlinear Control*, vol. 6, no. 9-10, pp. 983–998, 1996.

[36] G. Becker and a. Packard, “Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback,” *Systems & Control Letters*, vol. 23, no. 3, pp. 205–215, 1994.

[37] F. Wu, “A generalized LPV system analysis and control synthesis framework,” *International Journal of Control*, vol. 74, pp. 745–759, jan 2001.

[38] P. Apkarian and R. J. Adams, “Advanced gain-scheduling techniques for uncertain systems,” *IEEE Transactions on Control Systems Technology*, vol. 6, no. 1, pp. 21–32, 1998.

[39] C. Briat, *Linear Parameter-Varying and Time-Delay Systems*, vol. 3 of *Advances in Delays and Dynamics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

[40] C. Hoffmann and H. Werner, "A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 416–433, 2015.

[41] B. Lu, H. Choi, G. D. Buckner, and K. Tammi, "Linear parameter-varying techniques for control of a magnetic bearing system," *Control Engineering Practice*, vol. 16, pp. 1161–1172, oct 2008.

[42] T. Johansen, I. Petersen, J. Kalkkuhl, and J. Ludemann, "Gain-scheduled wheel slip control in automotive brake systems," *IEEE Transactions on Control Systems Technology*, vol. 11, pp. 799–811, nov 2003.

[43] I. Fialho and G. J. Balas, "Road adaptive active suspension design using linear parameter-varying gain-scheduling," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 43–54, 2002.

[44] Bei Lu, Fen Wu, and SungWan Kim, "Switching LPV control of an F-16 aircraft via controller state reset," *IEEE Transactions on Control Systems Technology*, vol. 14, pp. 267–277, mar 2006.

[45] R. Skelton and M. Izadi, "Tensegrity dynamics," *TAMU Technical Report*, 2017.

[46] P. T. Bandeira, P. C. Pellanda, and L. O. de Araujo, "New Haar-Based Algorithms for Stability Analysis of LPV Systems," *IEEE Control Systems Letters*, vol. 2, pp. 605–610, oct 2018.

[47] L. O. de Araujo, P. C. Pellanda, J. F. Galdino, and A. M. Simoes, "Haar-Based Stability Analysis of LPV Systems," *IEEE Transactions on Automatic Control*, vol. 60, pp. 192–198, jan 2015.

[48] R. Wehage and E. Haug, "Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems," *Journal of mechanical design*, vol. 104, no. 1, pp. 247–255, 1982.

[49] H. K. KHALIL, *Nonlinear Systems*. Prentice Hall, third ed., 2002.

[50] A. M. LYAPUNOV, "The general problem of the stability of motion," *International Journal of Control*, vol. 55, no. 3, pp. 531–534, 1992.

[51] M. Green and D. J. Limebeer, *Linear robust control*. Courier Corporation, 2012.

[52] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Society for Industrial Mathematics, 1994.

[53] J. S. Shamma, "An Overview of LPV Systems," in *Control of Linear Parameter Varying Systems with Applications*, vol. 9781461418, pp. 3–26, Boston, MA: Springer US, 2012.

[54] D. J. Leith and W. E. Leithead, "Survey of gain-scheduling analysis and design," *International Journal of Control*, vol. 73, pp. 1001–1025, jan 2000.

[55] Y. Fujisaki, F. Dabbene, and R. Tempo, "Probabilistic design of LPV control systems," *Automatica*, vol. 39, pp. 1323–1337, aug 2003.

[56] R. Bhattacharya, "Robust state feedback control design with probabilistic system parameters," in *53rd IEEE Conference on Decision and Control*, vol. 2015-Febru, pp. 2828–2833, IEEE, dec 2014.

[57] R. Bhattacharya, "A polynomial chaos framework for designing linear parameter varying control systems," in *2015 American Control Conference (ACC)*, pp. 409–414, July 2015.

[58] N. Wiener, "The homogeneous chaos," *American Journal of Mathematics*, vol. 60, pp. 897–936, Oct. 1938.

[59] V. Volterra, "Lecons sur les Equations Integrales et Integrodifferentielles," *Paris: Gauthier Villars*, 1913.

[60] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach.* New York, NY, USA: Springer-Verlag New York, Inc., 1991.

[61] R. H. Cameron and W. T. Martin, "The Orthogonal Development of Non-Linear Functionals in Series of Fourier-Hermite Functionals," *The Annals of Mathematics*, vol. 48, no. 2, pp. 385–392, 1947.

[62] R. Askey and J. Wilson, "Some Basic Hypergeometric Polynomials that Generalize Jacobi Polynomials," *Memoirs Amer. Math. Soc.*, vol. 319, 1985.

[63] T. Y. Hou, W. Luo, B. Rozovskii, and H.-M. Zhou, "Wiener Chaos Expansions and Numerical Solutions of Randomly Forced Equations of Fluid Mechanics," *J. Comput. Phys.*, vol. 216, no. 2, pp. 687–706, 2006.

[64] D. Xiu and G. E. Karniadakis, "Modeling Uncertainty in Flow Simulations via Generalized Polynomial Chaos," *J. Comput. Phys.*, vol. 187, no. 1, pp. 137–167, 2003.

[65] X. Wan, D. Xiu, and G. E. Karniadakis, "Stochastic solutions for the two-dimensional advection-diffusion equation," *SIAM J. Sci. Comput.*, vol. 26, no. 2, pp. 578–590, 2005.

[66] R. Ghanem and J. Red-Horse, "Propagation of Probabilistic Uncertainty in Complex Physical Systems Using a Stochastic Finite Element Approach," *Phys. D*, vol. 133, no. 1-4, pp. 137–144, 1999.

[67] R. Ghanem, "Ingredients for a General Purpose Stochastic Finite Elements Implementation," *Comput. Methods Appl. Mech. Eng.*, vol. 168, no. 1-4, pp. 19–34, 1999.

[68] F. S. Hover and M. S. Triantafyllou, "Application of polynomial chaos in stability and control," *Automatica*, vol. 42, no. 5, pp. 789–795, 2006.

[69] K. Kim and R. D. Braatz, "Generalized polynomial chaos expansion approaches to approximate stochastic receding horizon control with applications to probabilistic collision checking and avoidance," in *Control Applications (CCA), 2012 IEEE International Conference on*, pp. 350–355, IEEE, 2012.

[70] J. Fisher and R. Bhattacharya, “Linear quadratic regulation of systems with stochastic parameter uncertainties,” *Automatica*, vol. 45, no. 12, pp. 2831–2841, 2009.

[71] R. Bhattacharya and J. Fisher, “Linear receding horizon control with probabilistic system parameters,” in *Robust Control Design*, vol. 7, pp. 627–632, 2012.

[72] P. Dutta and R. Bhattacharya, “Nonlinear Estimation with Polynomial Chaos and Higher Order Moment Updates,” in *2010 American Control Conference, Marriott Waterfront*, (Baltimore, MD, USA), pp. 3142–3147, 2010.

[73] O. P. Le Maître and O. M. Knio, *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. Springer, 2010.

[74] M. Eldred and J. Burkardt, “Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification,” *AIAA paper*, vol. 976, no. 2009, pp. 1–20, 2009.

[75] D. Xiu, “Efficient collocational approach for parametric uncertainty analysis,” *Commun. Comput. Phys*, vol. 2, no. 2, pp. 293–309, 2007.

[76] M. S. Eldred, J. Burkardt, and V. Tech, “Comparison of Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Quantification,” *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, jan 2009.

[77] S. Oladyshkin and W. Nowak, “Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion,” *Reliability Engineering & System Safety*, vol. 106, pp. 179–190, 2012.

[78] J. A. Paulson, E. A. Buehler, and A. Mesbah, “Arbitrary polynomial chaos for uncertainty propagation of correlated random variables in dynamic systems,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3548–3553, 2017.

[79] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge university press, 2012.

[80] F. B. Hildebrand, *Introduction to numerical analysis*. Courier Corporation, 1987.

[81] R. Bhattacharya, “Robust state feedback control design with probabilistic system parameters,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 2828–2833, IEEE, 2014.

[82] C. W. Scherer and C. W. Hol, “Matrix sum-of-squares relaxations for robust semi-definite programs,” *Mathematical programming*, vol. 107, no. 1-2, pp. 189–211, 2006.

[83] H. Horisberger and P. Belanger, “Regulators for linear, time invariant plants with uncertain parameters,” *IEEE Transactions on Automatic Control*, vol. 21, pp. 705–708, October 1976.

[84] B. Polyak and R. Tempo, “Probabilistic robust design with linear quadratic regulators,” *Systems & Control Letters*, vol. 43, pp. 343–353, aug 2001.

[85] S. Lucia, J. A. Paulson, R. Findeisen, and R. D. Braatz, “On stability of stochastic linear systems via polynomial chaos expansions,” in *2017 American Control Conference (ACC)*, pp. 5089–5094, IEEE, may 2017.

[86] T. J. Rivlin, *An introduction to the approximation of functions*. Courier Corporation, 2003.

[87] G. M. Phillips, *Interpolation and approximation by polynomials*, vol. 14. Springer Science & Business Media, 2003.

[88] L. N. Trefethen, *Approximation theory and approximation practice*, vol. 128. Siam, 2013.

[89] S. A. Smolyak, “Quadrature and interpolation formulas for tensor products of certain classes of functions,” in *Doklady Akademii Nauk*, vol. 148, pp. 1042–1045, Russian Academy of Sciences, 1963.

[90] V. Barthelmann, E. Novak, and K. Ritter, “High dimensional polynomial interpolation on sparse grids,” *Advances in Computational Mathematics*, vol. 12, no. 4, pp. 273–288, 2000.

[91] M. Eldred, “Recent Advances in Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Analysis and Design,” in *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, no. May, (Reston, Virigina), pp. AIAA 2009—-2274, American Institute of Aeronautics and Astronautics, may 2009.

[92] B. Ganapathysubramanian and N. Zabaras, "Sparse grid collocation schemes for stochastic natural convection problems," *Journal of Computational Physics*, vol. 225, pp. 652–685, jul 2007.

[93] E. Novak and K. Ritter, "Simple cubature formulas with high polynomial exactness," *Constructive Approximation*, vol. 15, no. 4, pp. 499–522, 1999.

[94] F. Nobile, R. Tempone, and C. G. Webster, "A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data," *SIAM Journal on Numerical Analysis*, vol. 46, pp. 2309–2345, jan 2008.

[95] K. L. Judd, L. Maliar, S. Maliar, and R. Valero, "Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain," *Journal of Economic Dynamics and Control*, vol. 44, pp. 92–123, jul 2014.

[96] D. E. Ingber, "The architecture of life," *Scientific American*, vol. 278, no. 1, pp. 48–57, 1998.

[97] S. M. Levin, "The tensegrity-truss as a model for spine mechanics: biotensegrity," *Journal of mechanics in medicine and biology*, vol. 2, no. 03n04, pp. 375–388, 2002.

[98] G. Scarr, *Biotensegrity*. Handspring Publishing, United Kingdom, 2014.

[99] S. Dischiavi, A. Wright, E. Hegedus, and C. Bleakley, "Biotensegrity and myofascial chains: A global approach to an integrated kinetic chain," *Medical hypotheses*, vol. 110, pp. 90–96, 2018.

[100] S. M. Levin and D.-C. Martin, "Biotensegrity: the mechanics of fascia," *Fascia e the Tensional Network of the Human Body. The Science and Clinical Applications in Manual and Movement Therapy. Elsevier, Edinburgh*, pp. 137–142, 2012.

[101] A. Agogino, V. SunSpiral, and D. Atkinson, "Super ball bot-structures for planetary landing and exploration," *NASA Innovative Advanced Concepts (NIAC) Program, Final Report*, pp. 5–13, 2013.

[102] J. Rieffel and J.-B. Mouret, “Adaptive and resilient soft tensegrity robots,” *Soft robotics*, vol. 5, no. 3, pp. 318–329, 2018.

[103] C. Paul, J. W. Roberts, H. Lipson, and F. V. Cuevas, “Gait production in a tensegrity based robot,” in *Advanced Robotics, 2005. ICAR’05. Proceedings., 12th International Conference on*, pp. 216–222, IEEE, 2005.

[104] M. Shibata, F. Saijyo, and S. Hirai, “Crawling by body deformation of tensegrity structure robots,” in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*, pp. 4375–4380, IEEE, 2009.

[105] J. Rifkin, “Tensegrity joints for prosthetic, orthotic, and robotic devices,” Sept. 29 2005. US Patent App. 11/080,972.

[106] R. Goyal and R. E. Skelton, “Tensegrity system dynamics with rigid bars and massive strings,” *Multibody System Dynamics*, feb 2019.

[107] E. Coumans, “Bullet physics simulation,” in *ACM SIGGRAPH 2015 Courses*, p. 7, ACM, 2015.

[108] R. Smith *et al.*, “Open dynamics engine,” 2005.

[109] B. T. Mirletz, I.-W. Park, R. D. Quinn, and V. SunSpiral, “Towards bridging the reality gap between tensegrity simulation and robotic hardware,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 5357–5363, IEEE, 2015.

[110] MATLAB, *version 9.0 (R2016a).* Natick, Massachusetts: The MathWorks Inc., 2016.

[111] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 5026–5033, IEEE, 2012.

[112] J. Lee, M. X. Grey, S. Ha, T. Kunz, S. Jain, Y. Ye, S. S. Srinivasa, M. Stilman, and C. K. Liu, “Dart: Dynamic animation and robotics toolkit,” *The Journal of Open Source Software*, vol. 3, no. 22, p. 500, 2018.

[113] M. A. Sherman, A. Seth, and S. L. Delp, "Simbody: multibody dynamics for biomedical research," *Procedia Iutam*, vol. 2, pp. 241–261, 2011.

[114] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4397–4404, May 2015.

[115] R. E. Skelton and M. C. de Oliveira, *Tensegrity systems*, vol. 1. Springer, 2009.

[116] E. J. Haug and J. Yen, "Generalized coordinate partitioning methods for numerical integration of differential-algebraic equations of dynamics," in *Real-time integration methods for mechanical system simulation*, pp. 97–114, Springer, 1990.

[117] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer methods in applied mechanics and engineering*, vol. 1, no. 1, pp. 1–16, 1972.

[118] C. Chang and P. Nikravesh, "An adaptive constraint violation stabilization method for dynamic analysis of mechanical systems," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 107, no. 4, pp. 488–492, 1985.

[119] D.-S. Bae and S.-M. Yang, "A stabilization method for kinematic and kinetic constraint equations," in *Real-Time Integration Methods for Mechanical System Simulation*, pp. 209–232, Springer, 1990.

[120] S.-T. Lin and J.-N. Huang, "Stabilization of baumgarte's method using the runge-kutta approach," *Journal of Mechanical Design*, vol. 124, no. 4, pp. 633–641, 2002.

[121] K. Park and J. Chiou, "Stabilization of computational procedures for constrained dynamical systems," *Journal of Guidance, Control, and Dynamics*, vol. 11, no. 4, pp. 365–370, 1988.

[122] T. Park and E. Haug, "A hybrid numerical integration method for machine dynamic simulation," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 108, no. 2, pp. 211–216, 1986.

[123] P. Flores, M. Machado, E. Seabra, and M. T. da Silva, “A parametric study on the baumgarte stabilization method for forward dynamics of constrained multibody systems,” *Journal of computational and nonlinear dynamics*, vol. 6, no. 1, p. 011019, 2011.

[124] U. M. Ascher, H. Chin, L. R. Petzold, and S. Reich, “Stabilization of constrained mechanical systems with daes and invariant manifolds,” *Journal of Structural Mechanics*, vol. 23, no. 2, pp. 135–157, 1995.

[125] S. Yoon, R. Howe, and D. Greenwood, “Geometric elimination of constraint violations in numerical simulation of lagrangian equations,” *Journal of Mechanical Design*, vol. 116, no. 4, pp. 1058–1064, 1994.

[126] Q. Yu and I.-M. Chen, “A direct violation correction method in numerical simulation of constrained multibody systems,” *Computational Mechanics*, vol. 26, no. 1, pp. 52–57, 2000.

[127] W. Blajer, “Elimination of constraint violation and accuracy aspects in numerical simulation of multibody systems,” *Multibody System Dynamics*, vol. 7, no. 3, pp. 265–284, 2002.

[128] J. Zhang, D. Liu, and Y. Liu, “A constraint violation suppressing formulation for spatial multibody dynamics with singular mass matrix,” *Multibody System Dynamics*, vol. 36, no. 1, pp. 87–110, 2016.

[129] F. Marques, A. P. Souto, and P. Flores, “On the constraints violation in forward dynamics of multibody systems,” *Multibody System Dynamics*, vol. 39, no. 4, pp. 385–419, 2017.

[130] P. Flores and P. E. Nikravesh, “Comparison of different methods to control constraints violation in forward multibody dynamics,” in *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V07AT10A028–V07AT10A028, American Society of Mechanical Engineers, 2013.

[131] W. Blajer, “Methods for constraint violation suppression in the numerical simulation of constrained multibody systems–a comparative study,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 13-16, pp. 1568–1576, 2011.

[132] J. L. Lagrange, *Mécanique analytique*, vol. 1. Mallet-Bachelier, 1853.

[133] J. G. Papastavridis, “Analytical mechanics: A comprehensive treatise on the dynamics of constrained systems,” 2002.

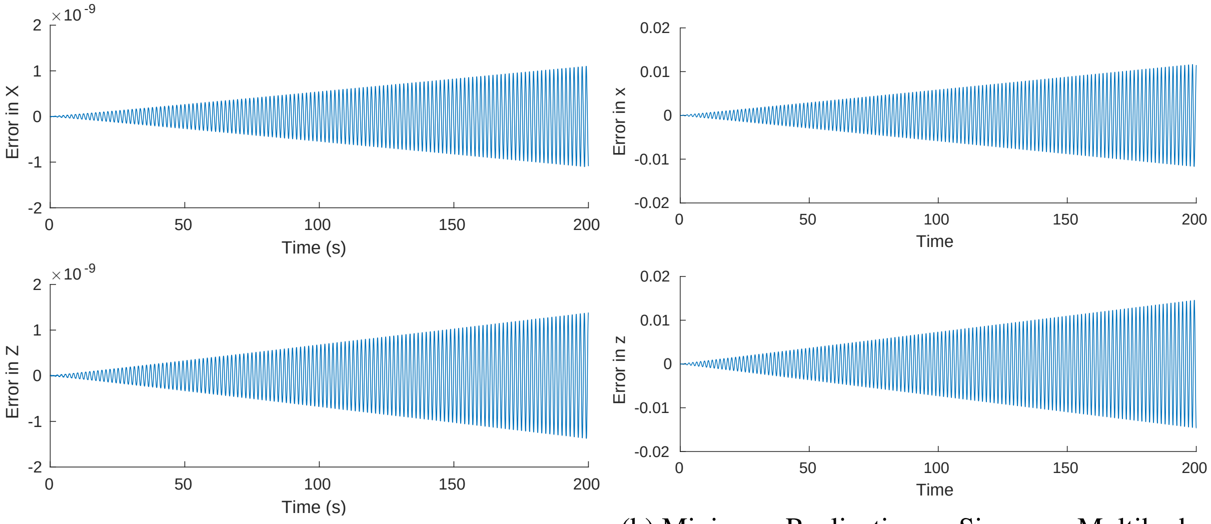
[134] J. R. Dormand and P. J. Prince, “A family of embedded runge-kutta formulae,” *Journal of computational and applied mathematics*, vol. 6, no. 1, pp. 19–26, 1980.

[135] L. F. Shampine and M. W. Reichelt, “The matlab ode suite,” *SIAM journal on scientific computing*, vol. 18, no. 1, pp. 1–22, 1997.

[136] S. Yoon, *Real-time simulation of constrained dynamic systems*. PhD thesis, University of Michigan, 1990.

# APPENDIX A

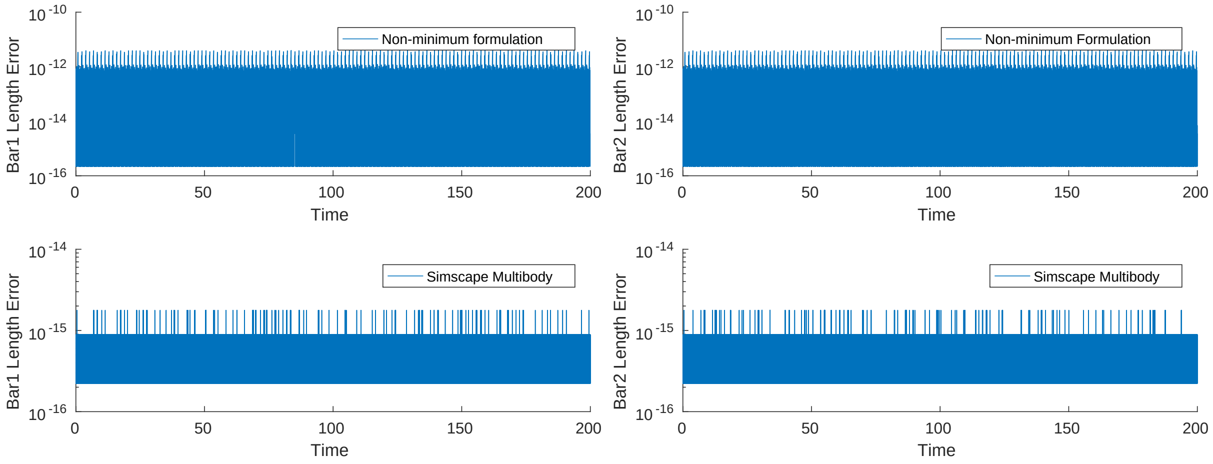
## PLOTS IN SECTION 5.4 OF Chapter 5



(a) Minimum Realization vs Proposed Method

(b) Minimum Realization vs Simscape Multibody

Figure A.1: Difference in motion of node 3 between minimum realization, the proposed method and Simscape Multibody in the example of the 2-bar structure shown in Figure 5.1a.



(a) Bar 1 length error

(b) Bar 2 length error

Figure A.2: Bar Length constraint violations observed in the example of the 2-bar structure described in Figure 5.1a.

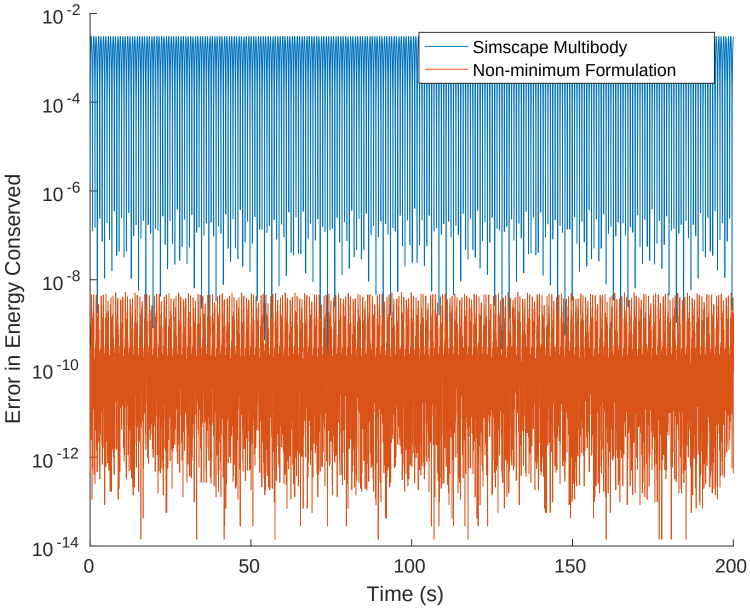
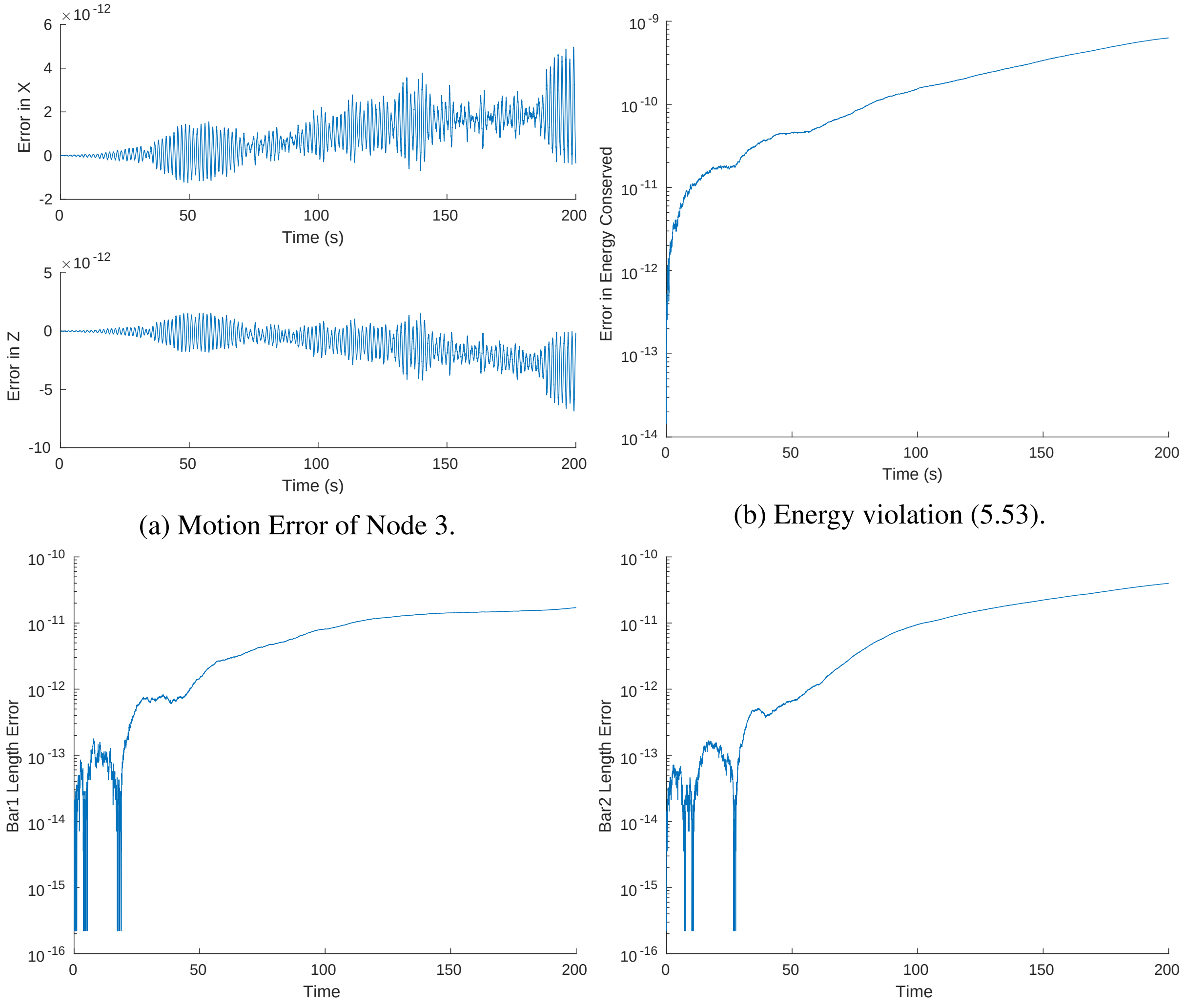


Figure A.3: Energy violation (5.53) observed in the example of the 2-bar structure shown in Figure 5.1a.



(a) Motion Error of Node 3.

(b) Energy violation (5.53).

(c) Bar 1 length error

(d) Bar 2 length error

Figure A.4: Motion error, constraint and energy violations if simulated at $10^{-14}$ tolerance without any correction in the example of the 2-bar structure shown in Figure 5.1a.

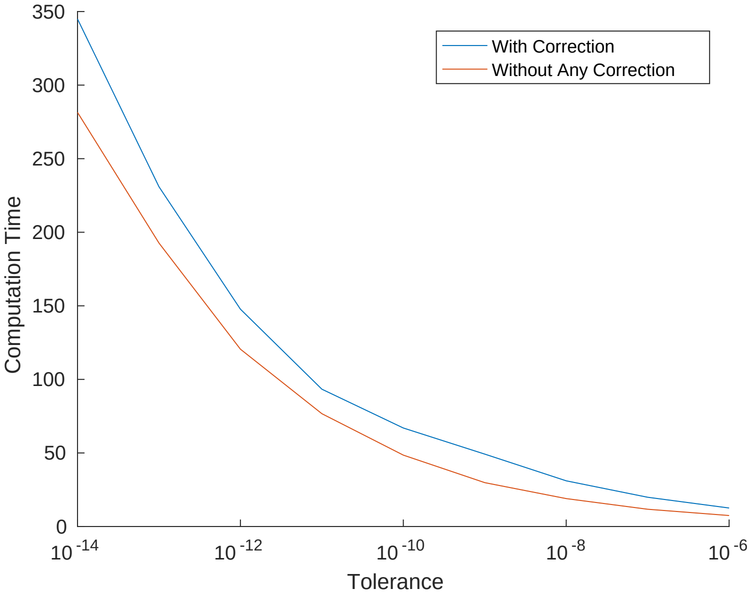
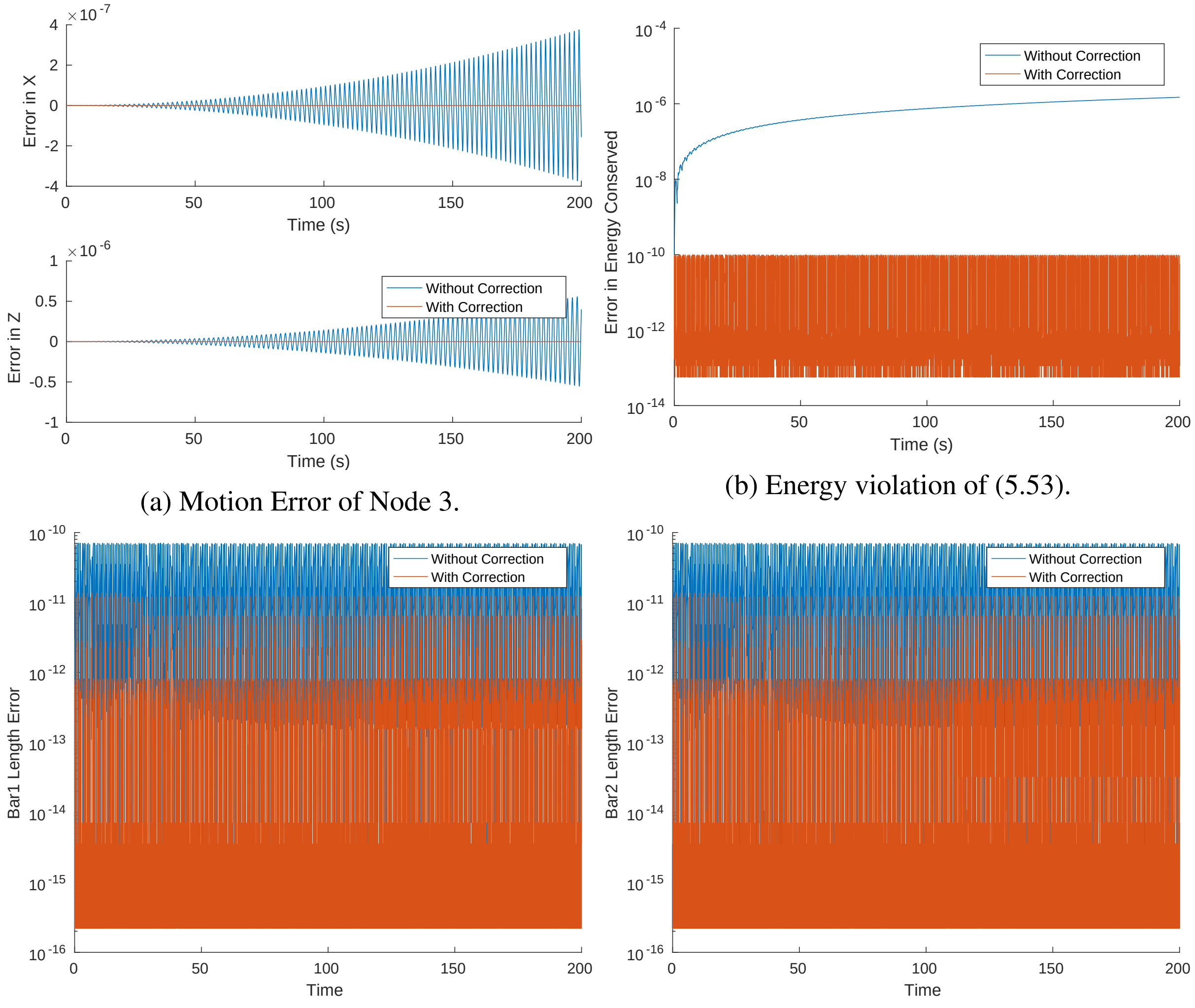


Figure A.5: Computation Times for the 2-bar structure example, simulated at different tolerances.



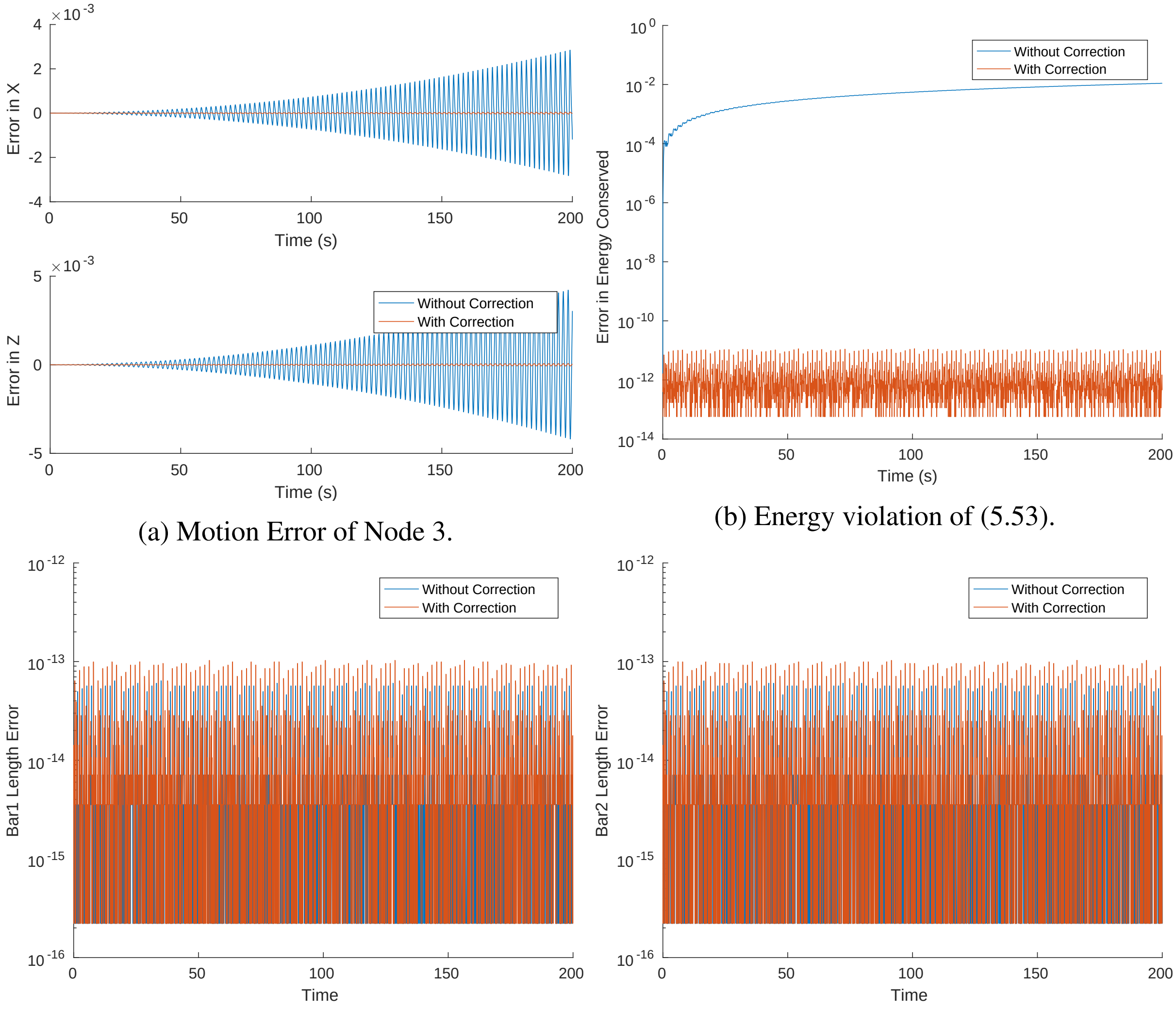
(a) Motion Error of Node 3.

(b) Energy violation of (5.53).

(c) Bar 1 Length Error.

(d) Bar 2 Length Error.

Figure A.6: Motion error, constraint and energy violations if simulated at $10^{-10}$ tolerance with and without energy correction in the example of the 2-bar structure shown in Figure 5.1a.



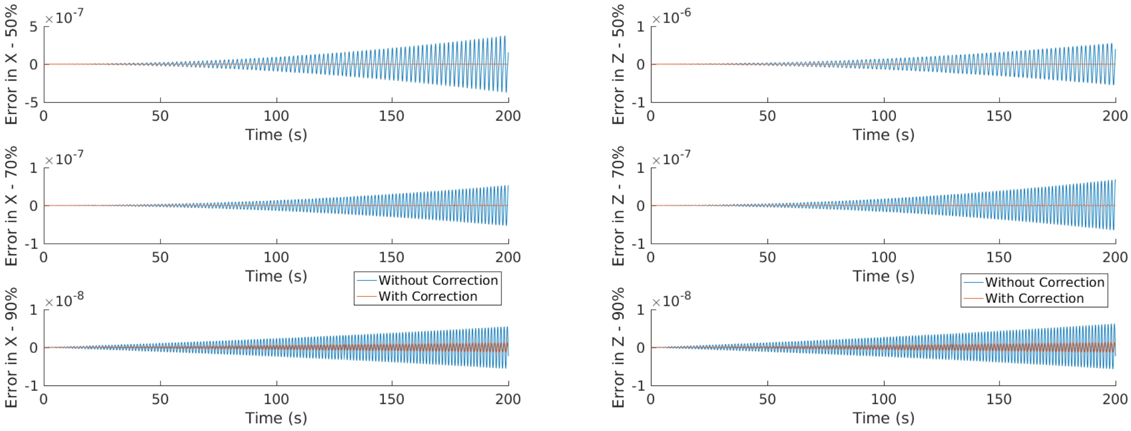
(a) Motion Error of Node 3.

(b) Energy violation of (5.53).

(c) Bar 1 Length Error.

(d) Bar 2 Length Error.

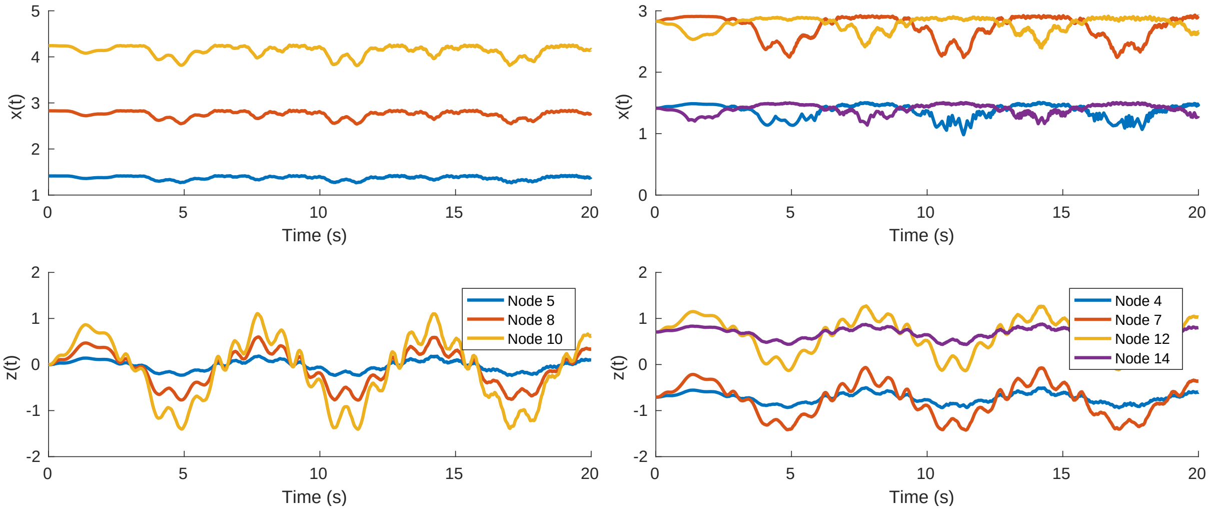
Figure A.7: Motion error, constraint and energy violations if simulated at $10^{-6}$ tolerance with and without energy correction in the example of the 2-bar structure shown in Figure 5.1a.



(a) Motion error in X of Node 3.

(b) Motion error in Z of Node 3.

Figure A.8: Motion error of Node 3 if simulated at $10^{-10}$ tolerance with and without energy correction at different rest lengths in the example of the 2-bar structure shown in Figure 5.1a.



(a) Motion of Tip and Connection Points.

(b) Motion of Nodes 4,7,12,14.

Figure A.9: Motion trajectories of the particular nodes in the example of the arm shown in Figure 5.1b.

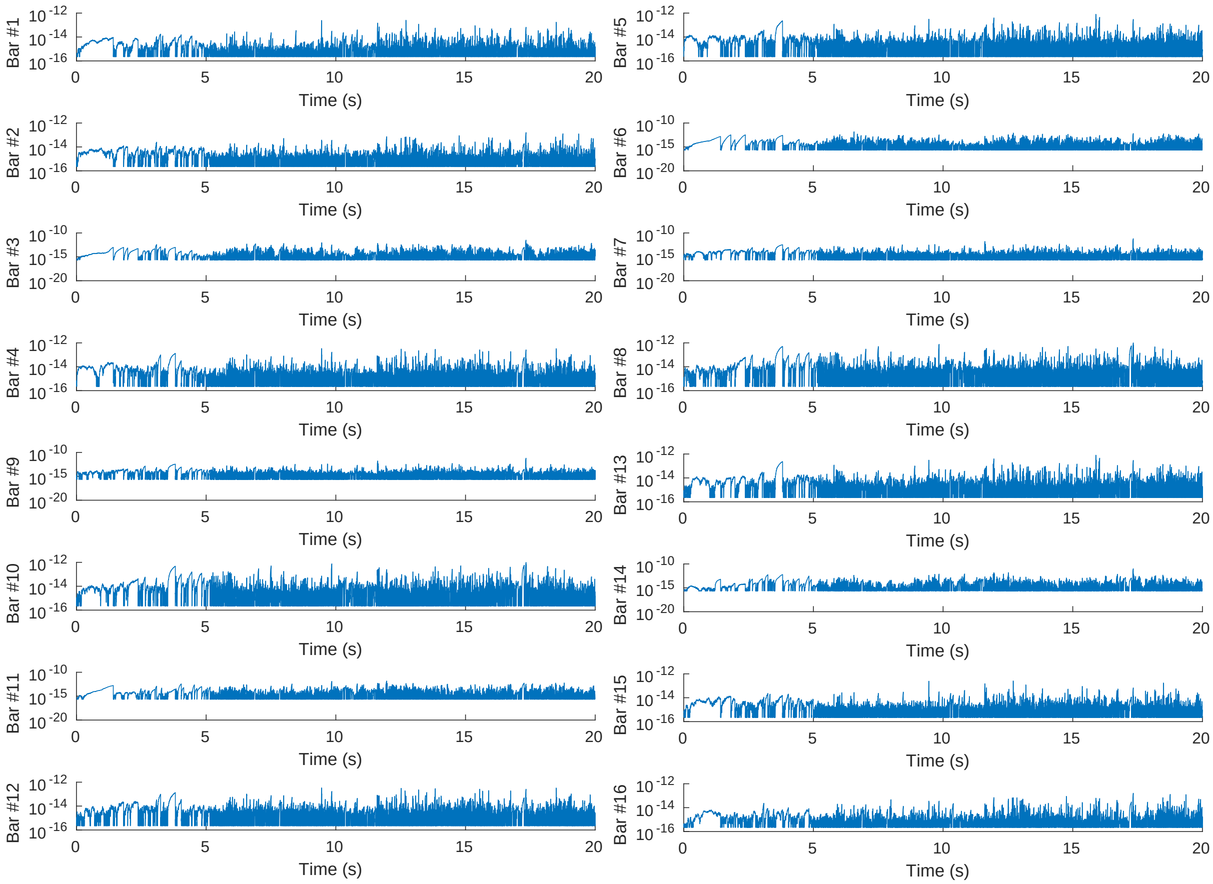


Figure A.10: Bar length errors in the example of the arm shown in Figure 5.1b, where the value of zero is set to the minimum positive double precision number.

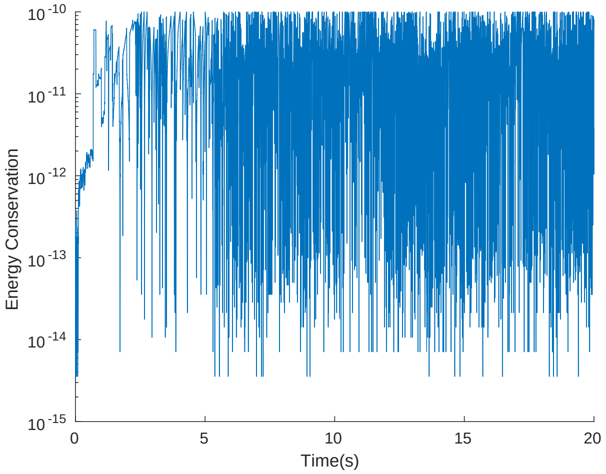
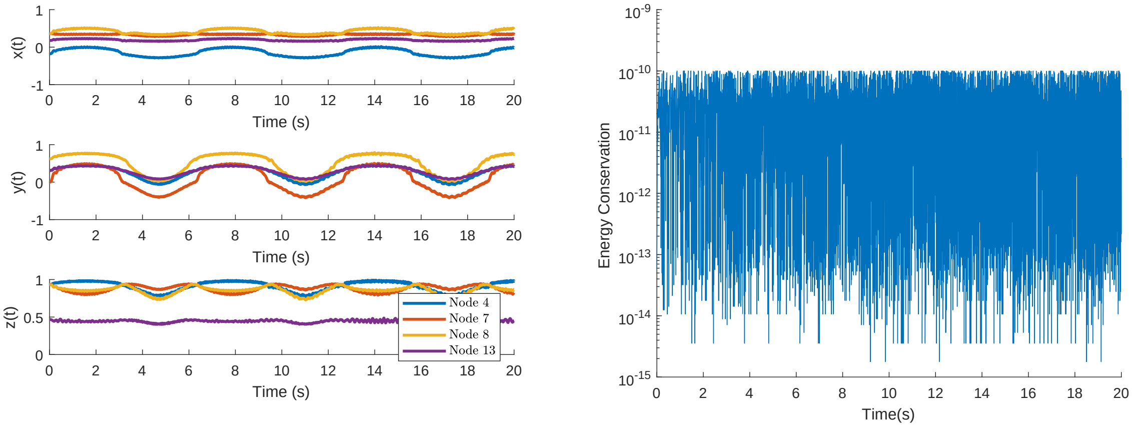


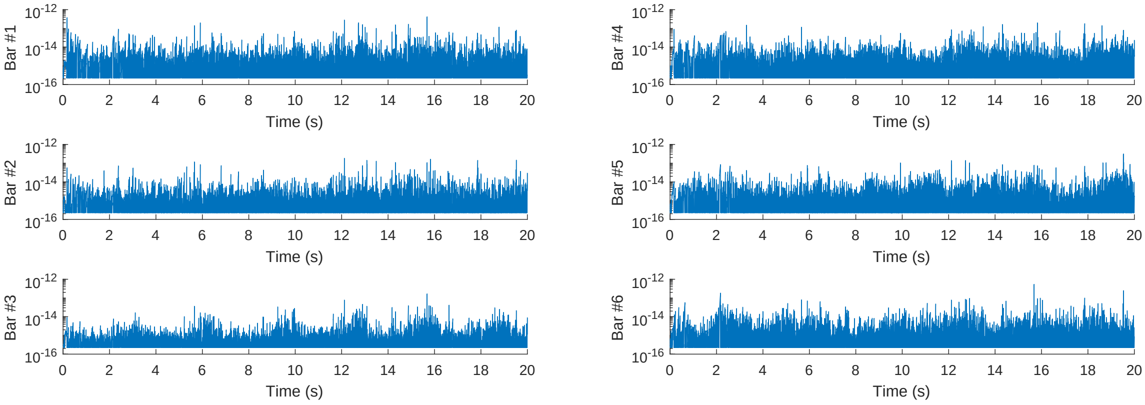
Figure A.11: Energy violation (5.53) of the arm shown in Figure 5.1b.



(a) Motion of nodes 4,7,8,13.

(b) Energy violation of (5.53).

Figure A.12: Motion and energy violations in the example of the ball shown in Figure 5.1c.



(a) Bar length error.

(b) Bar length error.

Figure A.13: Bar length errors in the example of the ball shown in Figure 5.1c, where the value of zero is set to the minimum positive double precision number.

# APPENDIX B

## PLOTS IN CHAPTER 6

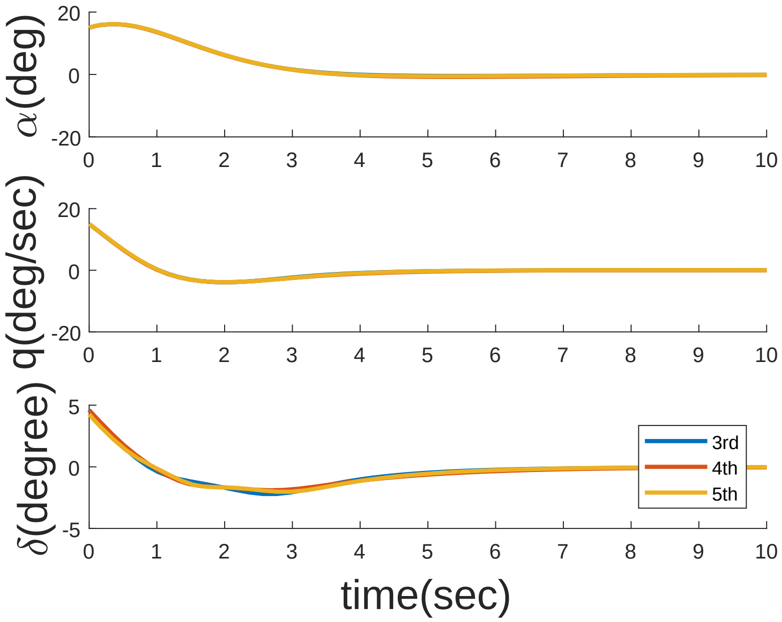


Figure B.1: State and control trajectories for the missile autopilot by applying $\boldsymbol{K}_{\text{pcLPV}}$ control with different order polynomial chaos expansion.

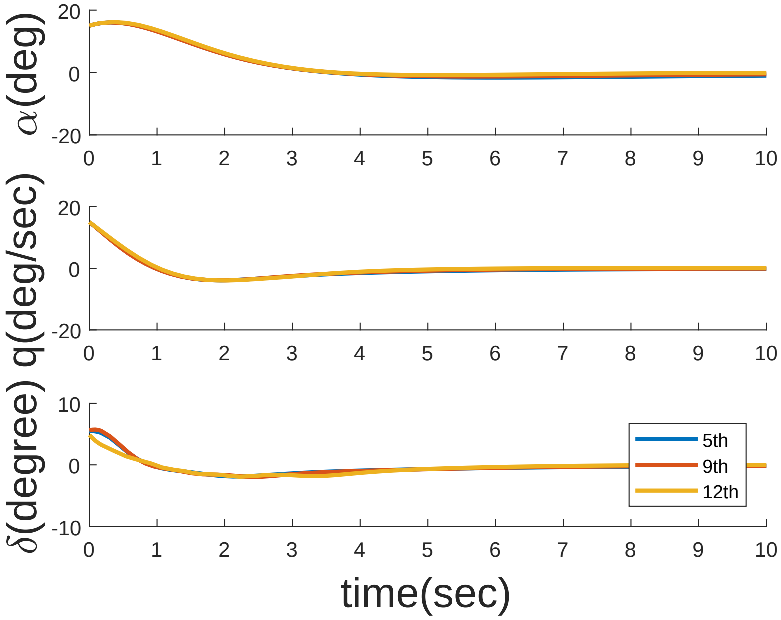


Figure B.2: State and control trajectories for the missile autopilot by applying $\boldsymbol{K}_{\text{scLPV}}$ with different order stochastic collocation expansion.

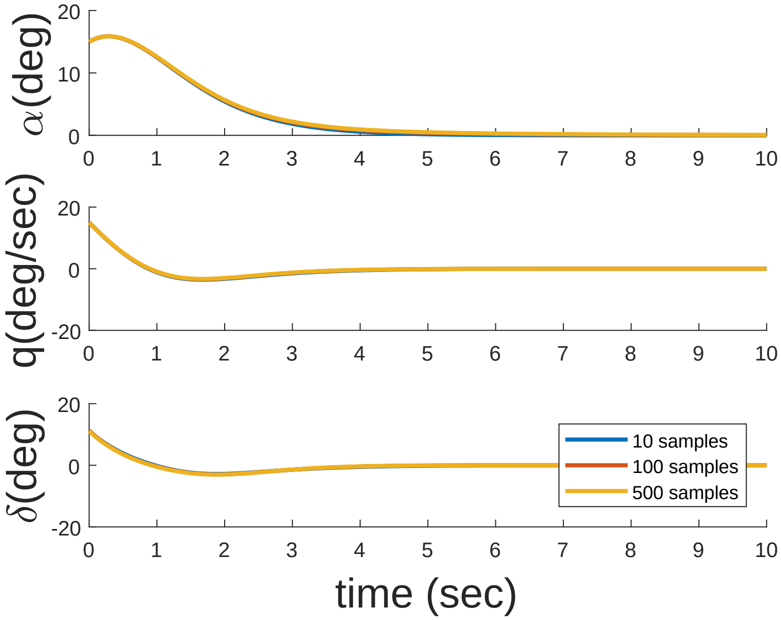


Figure B.3: State and control trajectories for the missile autopilot by applying $\boldsymbol{K}_{\text{LPV}}$ with different amount of sample points.

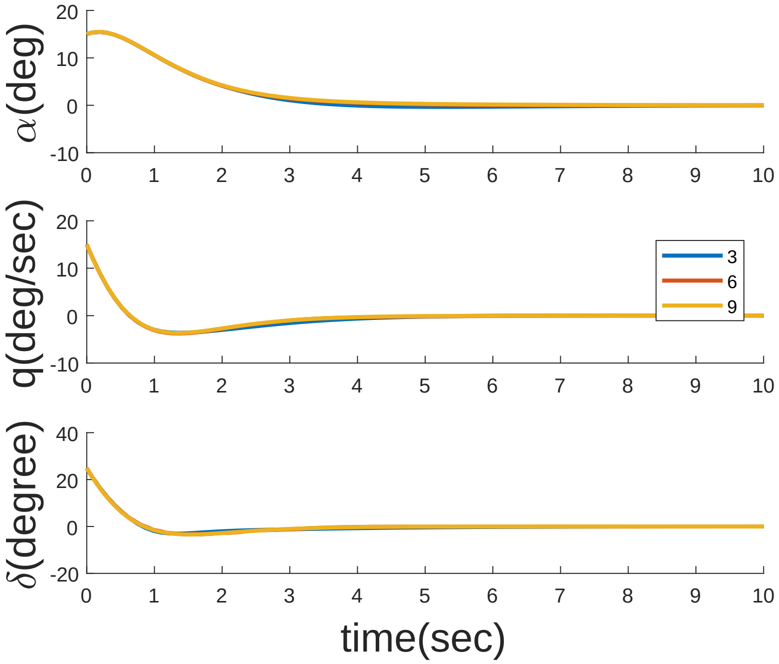


Figure B.4: State and control trajectories for the missile autopilot by applying $\boldsymbol{K}_{\mathrm{SG}}$ with different levels.

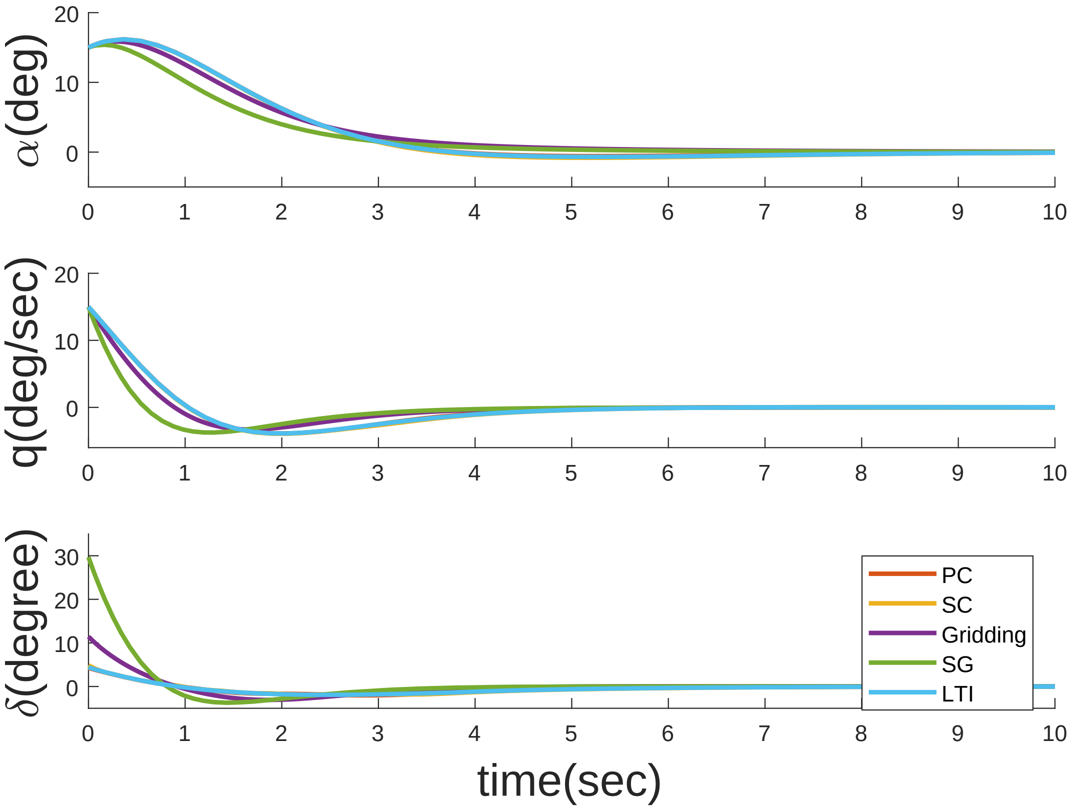


Figure B.5: State and control trajectories for the missile autopilot by applying types of controllers.

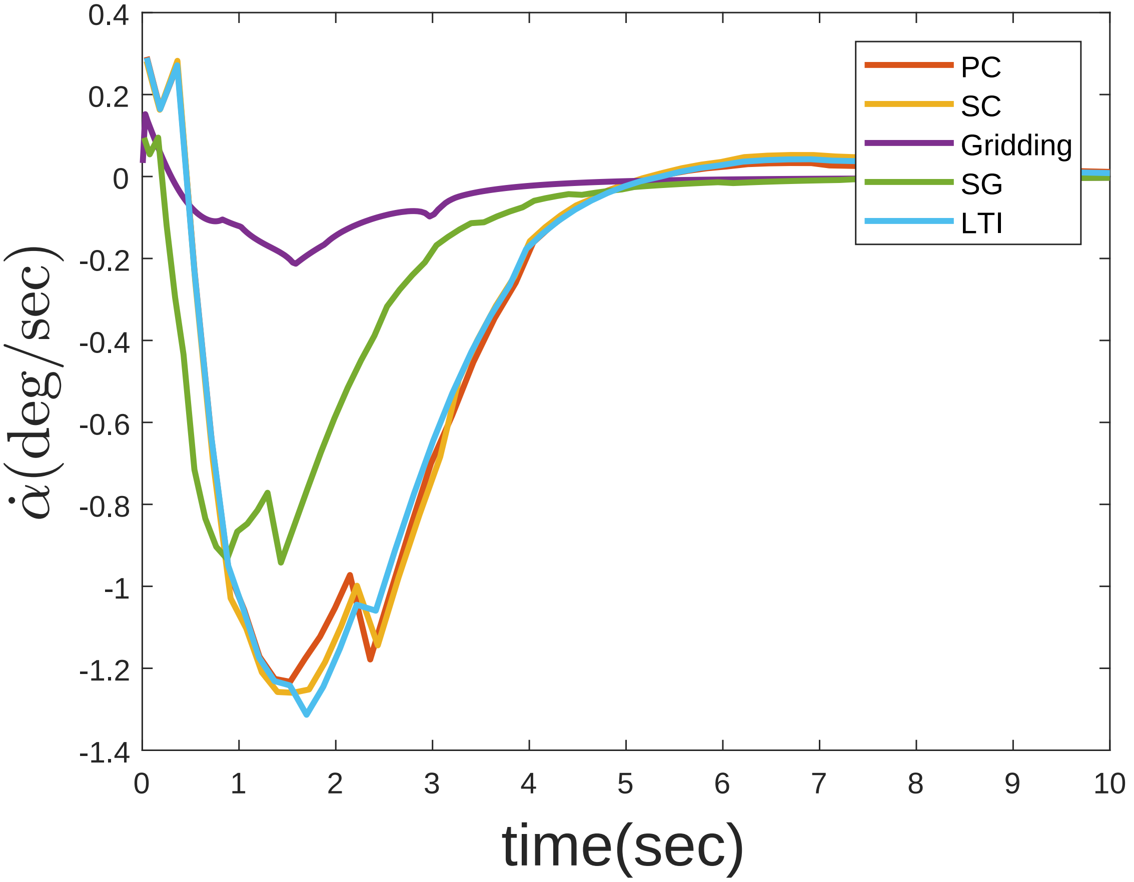


Figure B.6: Trajectories of the rate of angle of attack for the missile autopilot by applying types of controllers.

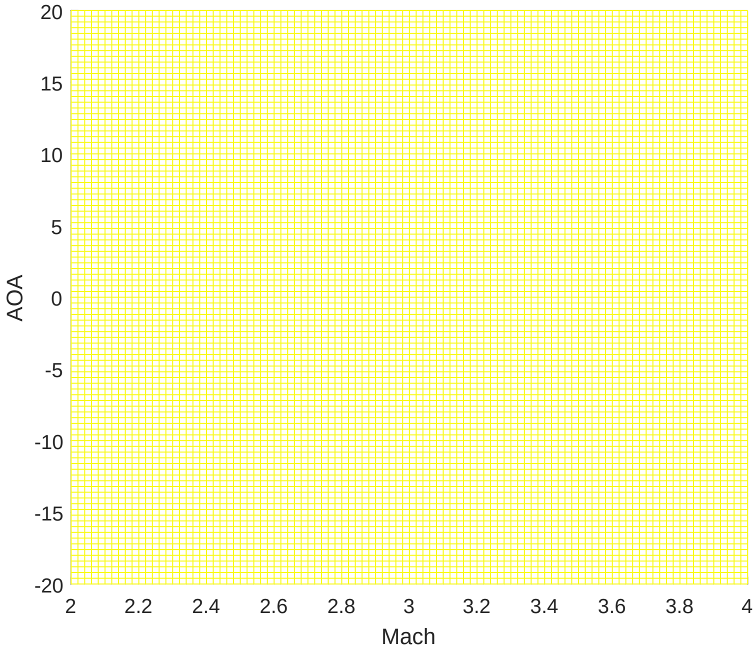


Figure B.7: Mesh plot of a finer grid to check the feasibility of conditions for $\boldsymbol{K}_{\text{sc}}$ with 7-th order expansion, where yellow indicates infeasible region and blue means feasible region.

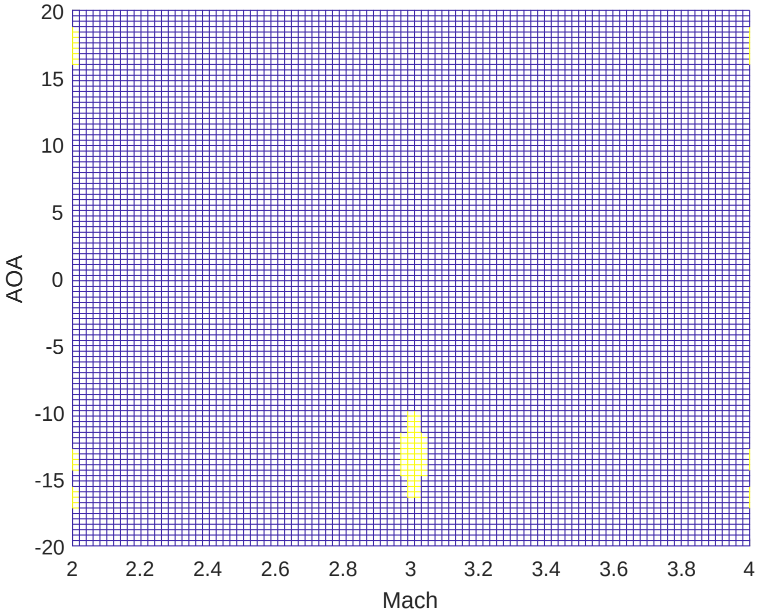


Figure B.8: Mesh plot of a finer grid to check the feasibility of conditions for $\boldsymbol{K}_{\text{LPV}}$ with 900 sample points, where yellow indicates infeasible region and blue means feasible region.

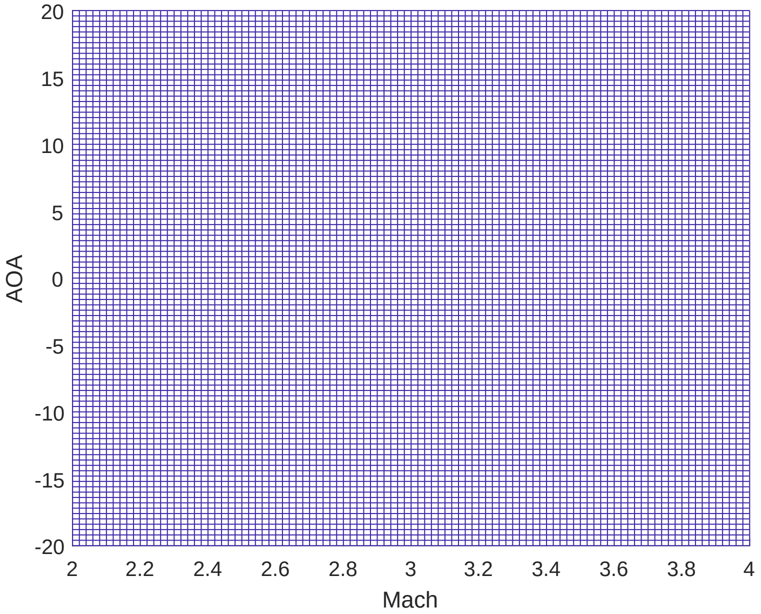


Figure B.9: Mesh plot of a finer grid to check the feasibility of conditions for $\boldsymbol{K}_{\text{LPV}}$ with level $k = 6$, where yellow indicates infeasible region and blue means feasible region.

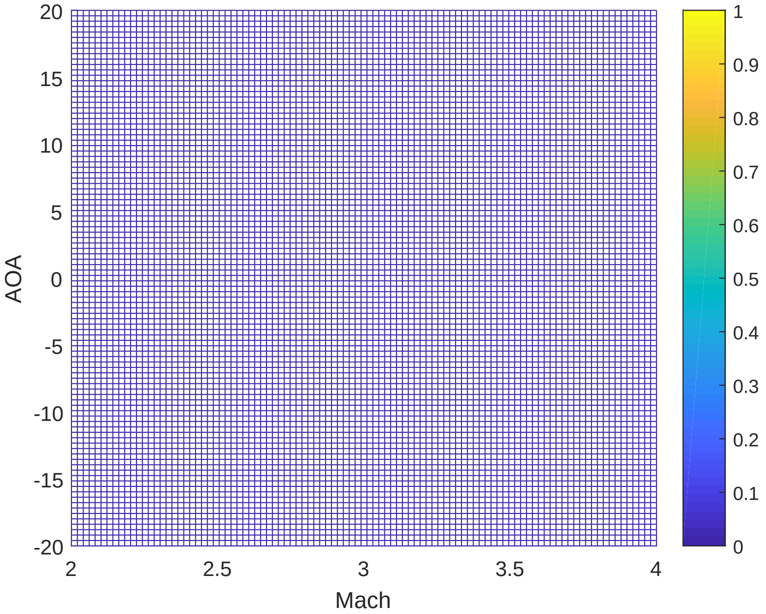


Figure B.10: Mesh plot of a finer grid to check the feasibility of conditions for $\boldsymbol{K}_{\text{LPV}}$ with 900 sample points and relaxation $\alpha = 0.01$, where yellow indicates infeasible region and blue means feasible region.

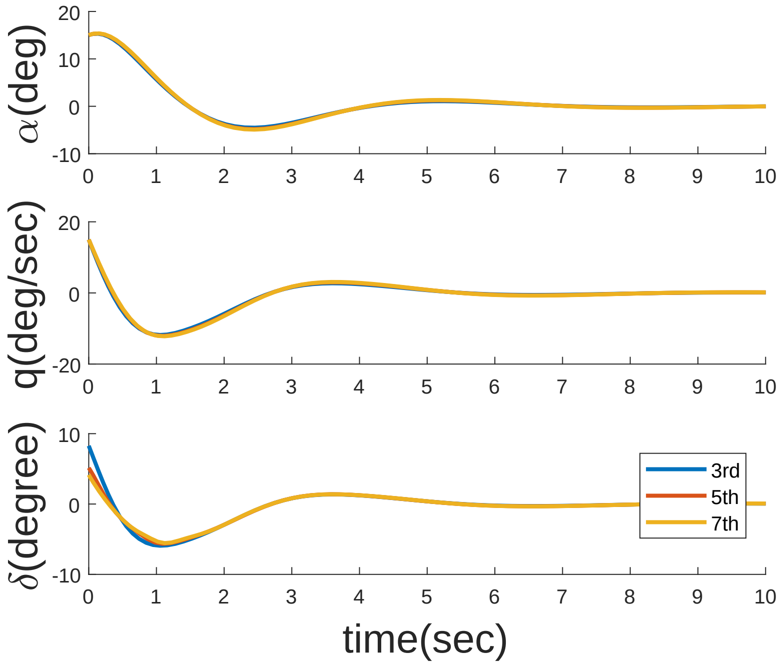


Figure B.11: State and control trajectories for the missile autopilot by applying $\boldsymbol{K}_{\text{scLPV}}$ with different order stochastic collocation expansion for multivariate case.

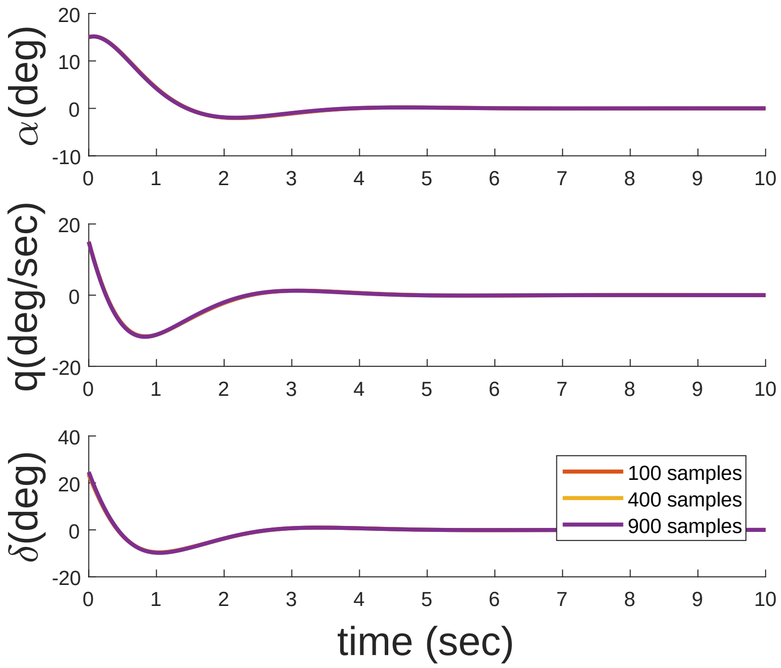


Figure B.12: State and control trajectories for the missile autopilot by applying $\boldsymbol{K}_{\mathrm{LPV}}$ with different amount of sample points for multivariate case.

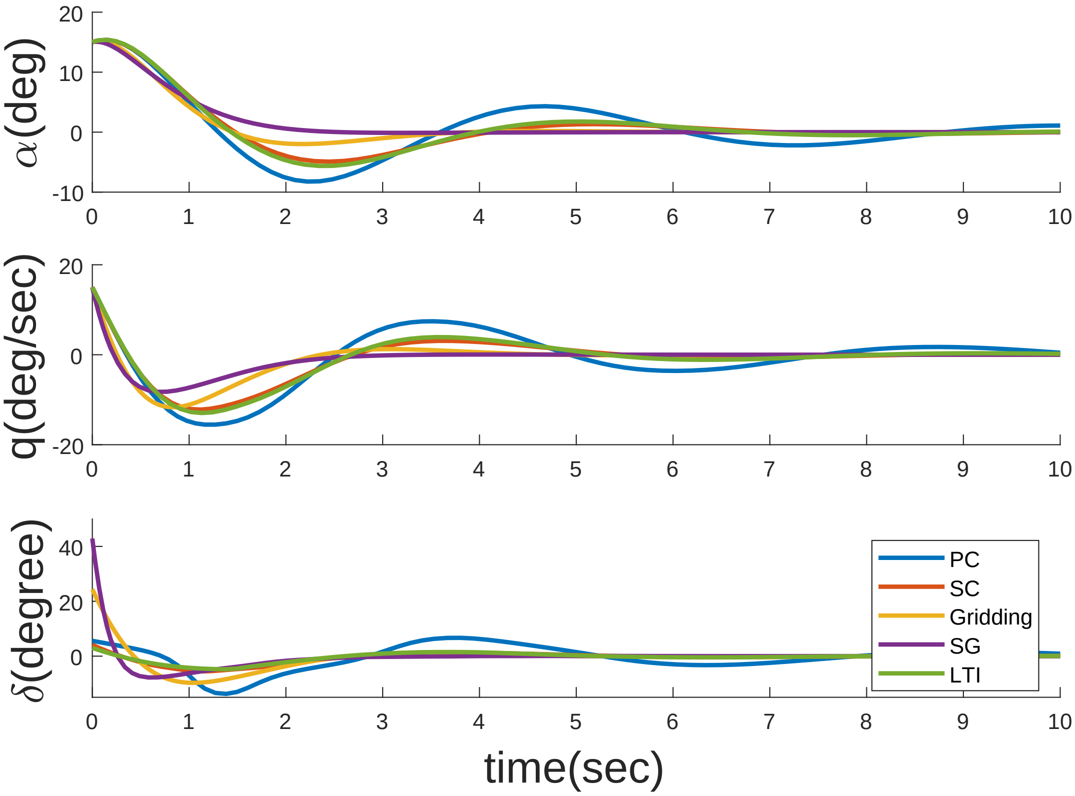


Figure B.13: State and control trajectories for the multivariate missile autopilot by applying types of controllers.

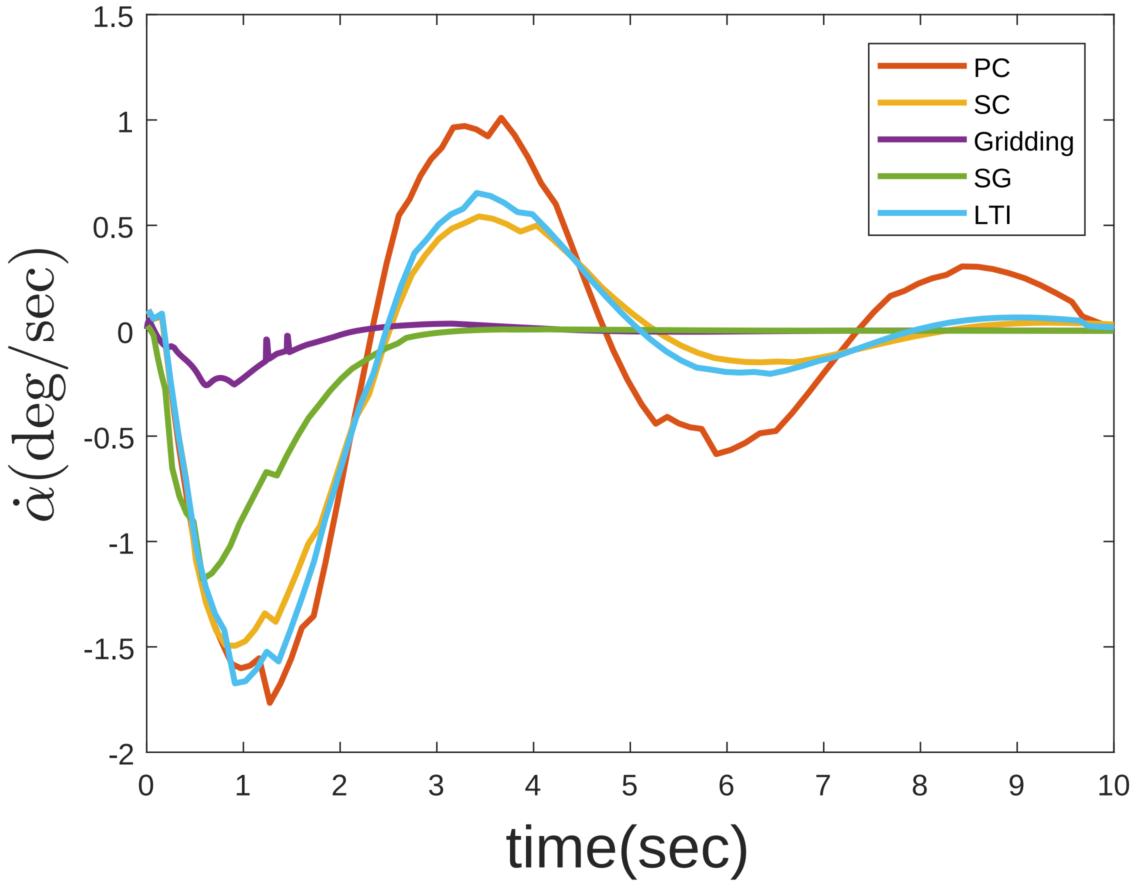


Figure B.14: Trajectories of the rate of angle of attack for the multivariate missile autopilot by applying types of controllers.

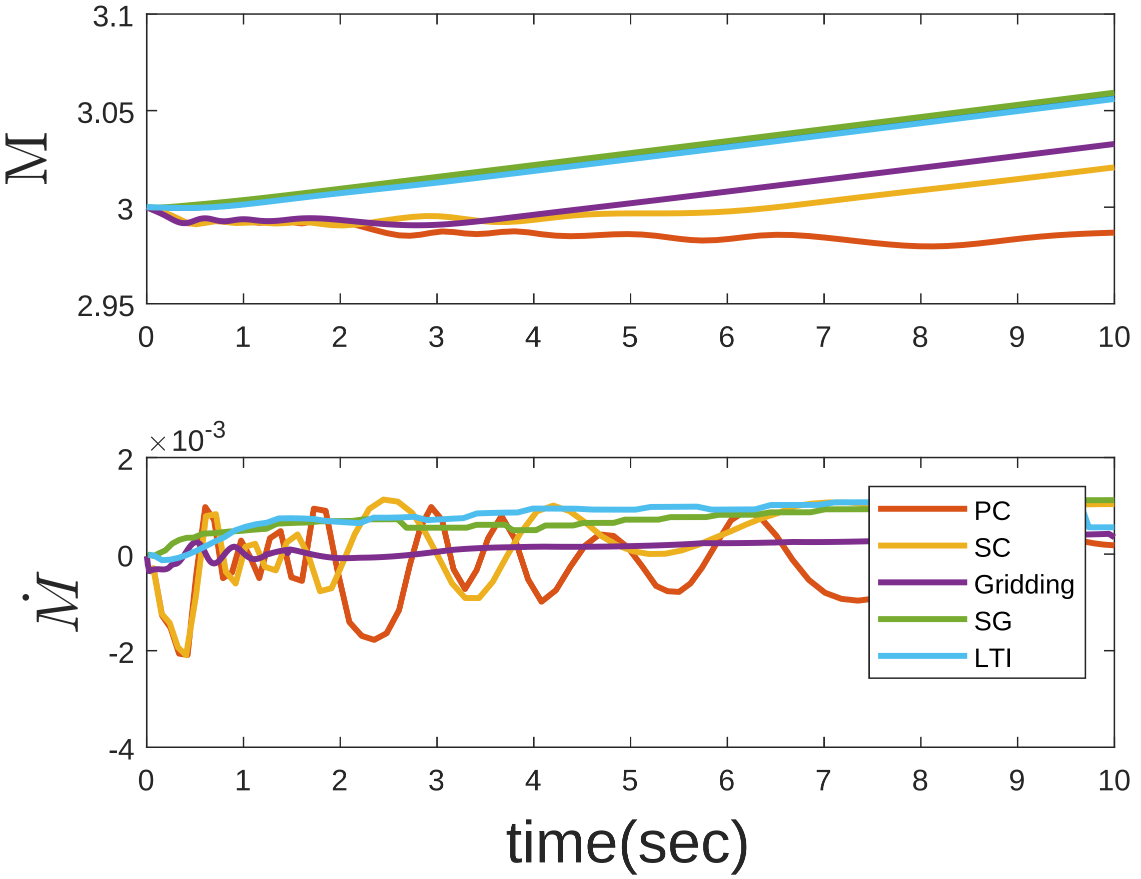


Figure B.15: Trajectories of the rate of Mach number for the multivariate missile autopilot by applying types of controllers.

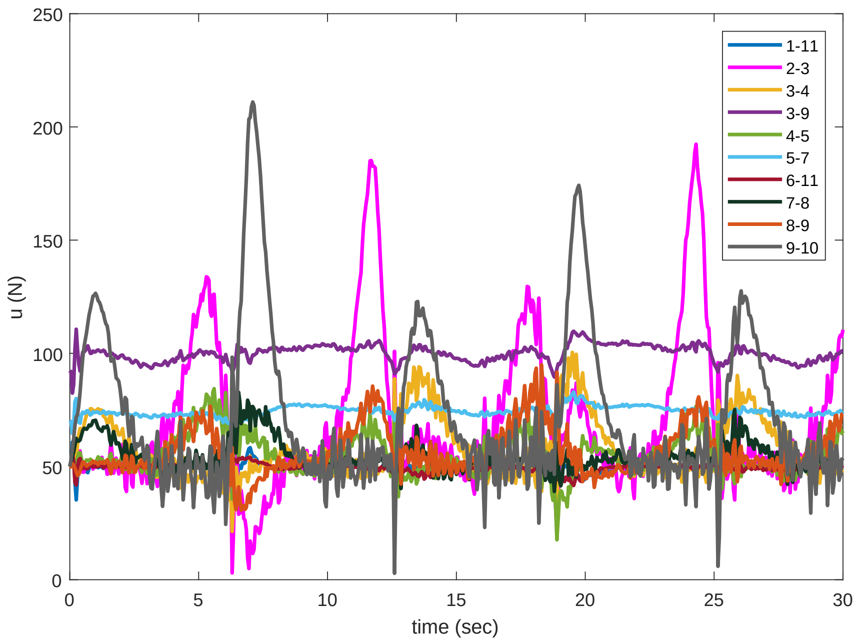


Figure B.17: The trajectory of control for the system, where each two numbers indicate a cable between two nodes.

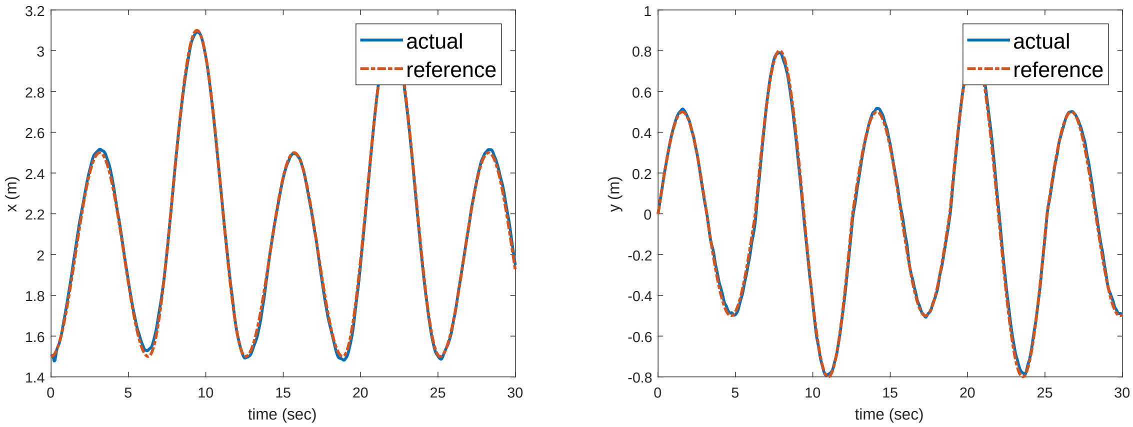


Figure B.16: The motion trajectories of the tip, i.e. node 6.