VIRTUAL CONTROL WITH APPLICATION TO HARD CONSTRAINED CONTROL PROBLEMS

By

Ali Mohammed Hussein Al-Hajjar

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mechanical Engineering – Doctor of Philosophy

2019

ABSTRACT

VIRTUAL CONTROL WITH APPLICATION TO HARD CONSTRAINED CONTROL PROBLEMS

By

Ali Mohammed Hussein Al-Hajjar

The saturation problem was and still a challenging research topic in the control community. For many decades it has attracted and still attracts many researchers and scientists to investigate and trials to give a resolution to the problem. There are many recent methods and techniques introduced in the literature as a solutions for the problem, but most of them are introducing an ad hoc methods, nonlinear or anti-windup schemes were in most, these methods are either incompetent in the sense of applicability to the broad control problems and/or they are computationally expensive. This study presents novel solutions modeling and control as a remedy for the saturation problem with applications. These novel proposed methods are systematic, numerically sounds and are applicable for a wide range of Linear Time Invariant (LTI), Linear Parameter Varying (LPV), discrete and continuous systems.

The first novel idea introduced in this dissertation is for LPV systems. The idea of scaling scheduling parameters is introduced and synthesized. Two novel Linear Parameter Varying (LPV) modeling and control techniques are proposed for active flutter suppression of a smart airfoil model. The smart airfoil model is instrumented with a moving mass that can be used to actively control the airfoil pitching and plunging motions. The first LPV modeling approach makes use of the moving mass position as a scheduling parameter, and the hard constraint at the boundaries is imposed by proper selection of the parameter varying function. The second modeling technique utilizes nonlinear springs and dampers, which are added to both ends of the airfoil groove to confine the motion of the moving mass. A state feedback based LPV gain-scheduling controller with the guaranteed H_∞ performance is proposed by utilizing the dynamics of the moving mass. In this novel idea, both the position of the moving mass and the free stream airspeed are considered as the scheduling parameters.

The second novel idea is the virtual hard constraints control approach. The proposed idea is introduced to deal with saturation control problems for linear time invariant (LTI) systems using virtual nonlinear springs and dampers for the optimal control design and then incorporates the virtual dynamics into the control input for real-time control. The basic idea is to increase the control cost when these state(s) move close to the hard constraint(s) and saturation limit(s), and the designed optimal control strategy will avoid moving these state(s) close to the hard constraint area. The proposed method is applied to a ball and beam system to demonstrate its feasibility and effectiveness, where virtual spring and damper are introduced during the control design. The performance of conventional LQR state feedback control and the associated virtual LQR-based MPC control are compared. The concept is especially relevant when implementing a physical hardware system, where one of the primary control design objectives is to safe-guard the hardware constraints during its operations.

The third novel idea is the LPV virtual control (LPVVC) scheme with hard constraints for LPV system is proposed. The main idea of virtual control and virtual mechanism is introduced here to the LPV systems to form LPVVC. The proposed control scheme is applied to the flutter suppression of a smart airfoil to demonstrate its effectiveness and ability of performance enhancement. The state-feedback LPV (gain-scheduling) control with guaranteed H_∞ performance is used for designing controller based on the model with virtual mechanisms. The virtual mechanisms used in this work are in terms of springs and dampers located at both ends of the airfoil groove to prevent the control mass from moving outside of the groove. Noticing that the virtual mechanisms are not limited to springs and dampers and can be choosing to be in any other form.

ACKNOWLEDGMENTS

This work is a culmination and dedication of many people's efforts in pursuing my PhD road and the words are unable to represents my feelings towards them. In addition, this section will fail to mention them all.

Human can meet and know many people in his life, but the chance of meeting good, humble, dedicated, understanding and lovely persons are very little. I would like to present my sincere thanks and gratitude to my advisor and mentor Professor Guoming Zhu who presents his efforts and experience and providing supports, excellence guidance, understanding and patience during these years of my work. His deep experience and perspective of the field has been great source of knowledge to me. Also, I would like to present my sincere thanks and appreciation to my second advisor Professor Sean Shan-Min Swei who offered his time and efforts generously and provided understanding and valuable feedback and patience during these years. His notes, advices and deep experience was great valuable source to me.

I would like to thank my committee members Professor Hassan Khalil, Professor Ranjan Mukherjee and Professor Zhaojian Li for they always offer their knowledge and time when I come to them without any appointment. I have learned a lot from the courses they taught. Further, a special thank for Professor Weihua Su for his valuable notes and support.

In addition, special thank to my friends Dr. Assaad Alsahlani , Dr. Ali Aljiboory, Tiany He, Dr. Yifan Men and Yingxu Wang for their help and support. Moreover, I would like to thank my friends in the ERC Lab in Michigan State University and in the Mechanical engineering department of Kufa university specially Professor Luay Alansari, Professor Ali S. Alathari, Professor Muhaned Alwaeli and Dr. Hayder Zainy, Professor Ahmed Alrajihy, Professor Alaa Aljasani, Dr. Moayed S. Radi (RH), Dr. Noor Muhsen, Dr. Maher Albaghdady, Dr. Yassar Sh., Hayder Hazim, , Sayed Dawood, Hayder Alsaraf, Karar Khafaji, Dr. Moayed Alharis, Ali Mamory, Ali M. Asadi, Dr. Moayed Zahid, Dr. Hyder Abdulhussein for their help and support.

Finally, I would like to acknowledge the financial support of the HCED (The Higher Committee For Education Development in Iraq) and University of Kufa for providing the financial support during my study at Michigan State University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

KEY TO SYMBOLS AND ABBREVIATIONS

Part (A): Symbols and abbreviations used for smart airfoil example.

b	Typical section semi-chord (the length of the groove).
c	Typical section chord.
C_L	Lift coefficient.
C_{L_α}	$\frac{dC_L}{d\alpha}$, ($C_{L_\alpha} = 2\pi$).
e	Elastic axis (e.a.) from elastic aerodynamic center, aft positive.
$\bar{e}$	Nondimensional e, e/b.
$F_a(t)$	Aerodynamic load on the airfoil.
g	Gravity constant.
$\bar{g}$	Nondimensional g, $\frac{g}{\omega_\alpha^2 b}$.
h	Plunging displacement.
$\bar{h}$	Nondimensional plunging displacement, $\frac{h}{b}$.
I_α	Mass moment of inertia about e.a. per unit span.
K_h	Spring constant for plunging mode.
K_α	Spring constant for pitching mode.
M	Mass per unit span of the typical section.
m	Mass of the control device.
q_p	Dynamic pressure, $\frac{\rho_a V^2}{2}$.
r_α	Radius of gyration about e.a., $\sqrt{\frac{I_\alpha}{Mb^2}}$.
t	Time.
u	Control input.
$\bar{u}$	Nondimensional control input, $\frac{u}{M\omega_\alpha^2 b}$.
V	Free stream airspeed.
$\bar{V}$	Nondimensional free stream airspeed, $\frac{V}{\omega_\alpha b}$.
x_α	Static unbalance, distance from e.a. to inertia axis, aft positive.
$\bar{x}_\alpha$	Nondimensional static unbalance, $\frac{x_\alpha}{b}$.
y	Displacement, traveling distance of control mass m.
$\bar{y}$	Nondimensional displacement, $\frac{y}{b}$.
α	Angel of incidence, positive nose up.
α_S	Small initial α.
α_L	Large initial α.
β	Mass ratio, $\frac{m}{M}$.
θ_1	First scheduling parameter, a function of $\bar{y}$.
θ_2	Second scheduling parameter, $\bar{V}$.
θ_3	Third scheduling parameter, $\bar{V}^2$.
μ	Mass ratio of the typical section to the apparent mass, $\frac{M}{\pi\rho_a b^2}$.

ω_h Natural frequency of uncoupled plunging.
ω_α Natural frequency of uncoupled pitching.
ρ_a Air density.
τ Nondimensional time, $\omega_\alpha t$.

Part (B): Symbols and abbreviations used for beam and ball example.
d Ball position w.r.t. beam center.
g Earth gravity.
I Moment of inertia of the beam.
L Length of the beam.
m Ball mass.
$k(\alpha)$ Virtual spring constant.
$c(\alpha)$ Virtual damper constant.
α Beam rotation angle.
τ Control torque.

CHAPTER 1

INTRODUCTION

1.1 Motivation and Vision

Saturation problem is inherent problem of the physical systems since actuator motion is always limited due to physical constraints. Consequently, handling this problem in the linear control theory and designing a linear controller is still a challenge. The problem in the literature was handled by using the antiwindup schemes or controller switching techniques [4, 5, 6, 7], to mention some but not to limit, which could cause substantial performance degradation. Nonlinear Model Predictive Controllers (MPC) [8, 9, 10] were also used to resolve the hard constrained control problems with significantly increased control complexity mainly due to the high computational power required by nonlinear MPC control. Many other methods and techniques were proposed and used in literature that either propose ad hoc techniques or posses the problems mentioned above. Therefore, for practical control applications, there is a need to develop practical control strategies, that can be implemented for real-time control, for systems with hard constraints to keep the system states staying within hard constraints. To the best of the author knowledge, there is no linear control design methods can handle the saturation problems efficiently, numerically sound, simple implementation and in the same time can be applied to various problems and systems which are LTI, LPV, continuous and discrete systems. This study presents such control design methods.

1.2 Contribution

The following are brief contribution of the dissertation:

1- A novel Linear Parameter Varying LPV modeling and control (LPV-1) is proposed, where the idea of scaling scheduling parameter introduced and synthesized with the application to smart airfoil for active flutter suppression control. The smart airfoil model is instrumented with a moving

mass that can be used to actively control the airfoil pitching and plunging motions. The LPV modeling approach makes use of the moving mass position as a scheduling parameter, and the hard constraint at the boundaries is imposed by proper selection of the parameter varying function. A state-feedback based LPV gain-scheduling controller with the guaranteed H_∞ performance is proposed by utilizing the dynamics of the moving mass. In this study, both the position of the moving mass and the free stream airspeed are considered as the scheduling parameters. The numerical simulations demonstrate the effectiveness of the proposed LPV control architectures by significantly improving the performance, while increasing the flutter speed and reducing the control effort.

2- A modeling technique utilizing nonlinear springs and dampers (LPV-2) is proposed and synthesized with the application to smart airfoil for active flutter suppression control, where the nonlinear springs and dampers are added to both ends of the airfoil groove to confine the motion of the moving mass. Also a state-feedback based LPV gain-scheduling controller with the guaranteed H_∞ performance is proposed by utilizing the dynamics of the moving mass. A simulation study is presented shows the efficiency and effectiveness of the proposed method in reducing the control effort while suppressing the airfoil flutter.

3- A modeling technique (LPV-2A) to reduce the number of scheduling parameters and hence reduce the conservativeness of the controller is proposed and applied to the smart airfoil for flutter suppression example.

4- A novel virtual hard constraints control is introduced to deal with saturation control problems for linear time invariant (LTI) systems using virtual springs and dampers for the optimal control design (such as linear quadratic regulation (LQR) based model-predictive-control (MPC)) and then incorporating the virtual dynamics into the control input for real-time control. The idea is to increase the control cost when these state(s) move close to the hard constraint(s) and saturation

limit(s), and the designed optimal control strategy will avoid moving these state(s) close to the hard constraint area. The proposed method is applied to a ball and beam system to demonstrate its feasibility and effectiveness, where virtual spring and damper are introduced during the control design. The performance of conventional LQR state feedback control and the associated virtual LQR-based MPC control are compared. Simulation results show effectiveness of the proposed method with additional design parameters (virtual spring and damper) introduced for control design.

5- A shape functions represents the profile of the virtual spring and damper is proposed and used in virtual LQR-based MPC control. The shape functions are used with the beam ball example to give extra tuning parameters to tune the virtual spring and damper and hence as a tuning parameters of the virtual controller. Introducing these shape functions opens the door for many other shape functions to be proposed and used along with the virtual controller as an extra tuning parameters and to show the flexibility and powerfulness of the proposed virtual control method.

6- A novel Linear Parameter Varying Virtual control (LPVVC) with the application to active flutter suppression control of smart airfoil. The novel proposed virtual hard constraints control is proposed and synthesized for the LPV systems. A state-feedback based LPV gain-scheduling controller with the guaranteed H_∞ performance is used. The virtual spring and damper is introduced during the modeling and controller design. Simulation results and numerical investigation is introduced to show the effectiveness and efficiency of the method.

1.3 Dissertation Outline

The dissertation is organized as following: Chapter two introduces some LPV preliminaries that will be used for the subsequent chapters. Chapter three presents the novel scaling scheduling parameter Linear Parameter Varying (LPV) control of active flutter suppression of smart airfoil along with the scaling scheduling parameter with weighing function (LPV-1), scaling scheduling

parameter with nonlinear spring and damper (LPV-2) and reduced number scheduling parameter modeling (LPV-2A). Chapter four introduces the novel proposed virtual control with application to the beam and ball system. In addition, the proposed shape functions for the virtual spring and damper are presented. Then in chapter five the novel proposed Linear Parameter Varying Virtual control (LPVVC) with application to flutter suppression control of smart airfoil is investigated. Chapter six summarizes the conclusions and recommendations for future works.

CHAPTER 2

PRELIMINARIES

This chapter will briefly presents some LPV modeling preliminaries which will be used for the chapters three and five. Generally the polytopic representation of linear parameter varying system is solved by convex optimization, where the system is parametrized and transfered from affine representation to the multi-simplex representation to be solved using optimization packages [11] and then utilizing some general equations [3, 12] the system matrices to be reconstructed in the affine representations for the controller implementation. In doing so, some definitions and lemmas are needed for this transformation and will be presented in this chapter.

2.1 Linear Parameter Varying (LPV) preliminaries

Generally the set of the natural numbers is referred by N and the set of the real numbers is referred by R.

Definition 1: [13, 14] A unit simplex Θ_r is a polytope of r vertices defined as

$$\Theta_r = \left\{ \rho = [\rho_1, \cdots, \rho_r] : \sum_{i=1}^{r} \rho_i = 1, \rho_i \geq 0, i = 1, 2, .., r \right\}$$

since variable ρ_i changes inside a unit simplex Θ_r.

Definition 2: [15] The Cartesian product of a limited number of simplexes is a multi-simplex Θ. For instance, if there are q simplexes, then

$$\Theta = \Theta_{N_1} \times \Theta_{N_2} \times \Theta_{N_3} \times \cdots \Theta_{N_q} = \prod_{i=1}^{q} \Theta_{N_i},$$

where the dimension of multi-simplex Θ is the index $N = (N_1, \ldots, N_q)$, such that ρ in Θ represents $(\rho_1, \rho_2, ..., \rho_q)$ and each ρ_i in Θ_{Ni} represents $(\rho_i(1), \rho_i(2), ..., \rho_i(r))$.

Definition 3: [13] (Homogeneous Polynomial): Given a unit-simplex Γ_N of dimension $N \in \mathcal{N}$, a polynomial $P(\rho)$ defined on $\mathcal{R}^N$ of degrees $g \in \mathcal{N}$ is called homogeneous if all of its monomials have the same total degree g.

Definition 4: [13, 14] (Γ-Homogeneous Polynomial): Given a Multi-simplex Γ of dimension $N \in \mathcal{N}^L$, a polynomial $P(\rho)$ defined on $\mathcal{R}^N$ of degrees $g \in \mathcal{N}^L$ is called Γ-homogeneous if for any given integer i_0, with $1 \leq i_0 \leq L$, and for any given $\rho_i \in \mathcal{R}^{N_i}$, for $1 \leq i \neq i_0 \leq L$, the partial application $\rho_{i_0} \in \mathcal{R}^{N_{i_0}} \mapsto P(\rho)$ is a homogeneous polynomial in ρ_{i_0}.

For $N \in \mathcal{N}$ and $g \in \mathcal{N}$, let $\mathcal{K}_N(g)$ be the set of N-tuples obtained from all possible combinations of N nonnegative integers $k_i, i = 1, ..., N$ with sum $k_1 + k_2 + ... + k_N = g$. The number of elements in $\mathcal{K}_N(g)$ is given by

$$J_N(g) = \frac{(N+g-1)!}{g!(N-1)!}$$

Now, let $N \in \mathcal{N}^L$ and $g \in \mathcal{N}^L$. In this case, the set $\mathcal{K}_N(g)$ is defined as the Cartesian product $\mathcal{K}_N(g) = \mathcal{K}_{N_1}(g_1) \times ... \times \mathcal{K}_{N_L}(g_L)$. A Γ-homogeneous polynomial matrix $A(\rho)$ of degree $g \in \mathcal{N}^L$ is defined as follows

$$A(\rho) = \sum_{k \in \mathcal{K}_N(g)} \rho^k A_k$$

where $A_k \in \mathcal{R}^{n \times r}$ is a matrix- valued coefficient and ρ^k the corresponding monomial which is homogeneous of degree g_i in each variable ρ_i, that is, ρ^k is given by

$\rho^k = \rho_1^{k_1} \rho_2^{k_2} ... \rho_L^{k_L}$, where $\rho_i^{k_i} = \rho_{i,1}^{k_{i,1}} \rho_{i,2}^{k_{i,2}} ... \rho_{i,N_i}^{k_{i,N_i}}$, $k_i = (k_{i,1}, k_{i,2}, ..., k_{i,N_i})$ such that $k_{i,1} + k_{i,2} + ... + k_{i,N_i} = g_i$. Note that the indices $k = (k_1, k_2, ..., k_L)$ are obtained by combining all of the N-tuples of the sets $\mathcal{K}_{N_i}(g_i)$, for $i = 1, ..., L$, yielding a total of $J_N(g)$ monomials equal to

$$J_N(g) = \prod_{i=1}^{L} J_{N_i(g_i)}$$

Example 2.1.1 *If $\rho \in \Gamma_4$, then the following polynomial $P(\rho) = 5\rho_1^5 + 2\rho_2^3\rho_3^2 - \rho_1^2\rho_2^1\rho_4^2 + 3\rho_1^3\rho_2^2 + \rho_1^2\rho_2\rho_3\rho_4$ is a polynomial if degree $g = 5$*

Example 2.1.2 *Let ρ take values in the multi-simplex Γ of dimension N=(2,3), so the following polynomial $P(\rho) = \rho_{1,1}^3\rho_{2,2} - 3\rho_{1,1}\rho_{1,2}^2\rho_{2,3} - \rho_{1,2}^3\rho_{2,1} + 6\rho_{1,1}^2\rho_{1,2}\rho_{2,2}$ is a Γ-homogeneous polynomial of degree g=(3,1).*

Example 2.1.3 *For example, a Γ-homogeneous polynomial of degree g=(2,3) defined over the multi-simplex Γ of dimension N=(2,2) yields $\mathcal{K}_N(g) = \mathcal{K}_2(2) \times \mathcal{K}_2(3) = (2,0),(1,1),(0,2) \times (3,0),(2,1),(1,2),(0,3)$, with $J_N(g) = J_2(2).J_2(3) = 3.4 = 12$. Over this multi-simplex, a Γ-homogeneous matrix valued polynomial $A(\rho)$ takes the form*

$$
\begin{aligned}
A(\rho) = \rho_{1,1}^2(\rho_{2,1}^3 A_{((2,0),(3,0)} + \rho_{2,1}^2\rho_{2,2}A_{((2,0),(2,1)} + \rho_{2,1}\rho_{2,2}^2 A_{((2,0),(1,2)} + \rho_{2,2}^3 A_{((2,0),(0,3)}) \\
+\rho_{1,1}\rho_{1,2}(\rho_{2,1}^3 A_{((1,1),(3,0)} + \rho_{2,1}^2\rho_{2,2}A_{((1,1),(2,1)} + \rho_{2,1}\rho_{2,2}^2 A_{((1,1),(1,2)} + \rho_{2,2}^3 A_{((1,1),(0,3)}) \\
+\rho_{1,2}^2(\rho_{2,1}^3 A_{((0,2),(3,0)} + \rho_{2,1}^2\rho_{2,2}A_{((0,2),(2,1)} + \rho_{2,1}\rho_{2,2}^2 A_{((0,2),(1,2)} + \rho_{2,2}^3 A_{((0,2),(0,3)})
\end{aligned}
$$

More details will be presented in chapter three and chapter five. Also, for more LPV preliminaries one can consult [11, 12, 13, 14, 15] and references therein.

2.2 Linear Quadratic Regulator (LQR) preliminaries

In this work, a LQR-based MPC control is utilized in chapter two for the virtual control with beam and ball system example. In this section, a brief preliminaries of Linear Quadratic Regulation will be introduced. Most of the formulation are from [16].

The linear quadratic optimal control problem approached using Lagrangian and Hamiltonian functions. Considering a discrete system described as following:

$$x(k+1) = A(k)x(k) + B(k)u(k) \tag{2.1}$$

where, $k = k_0, ..., k_f - 1$, $x(k)$ is an (nth) order state vector, $u(k)$ is an (rth) order control vector, and $A(k)$ and $B(k)$ are matrices of appropriate dimensions. With initial condition $x(k = k_0) = x(k_0)$ and a general performance index (PI) with terminal cost as follows:

$$J = J(x(k_0), u(k_0), k_0 = \frac{1}{2}x'(k_f)F(k_f)x(k_f) + \frac{1}{2}\sum_{k=k_0}^{k_f-1}[x'(k)Q(k)x(k) + u'(k)R(k)u(k)] \tag{2.2}$$

where, $F(k)$ and $Q(k)$ are symmetric positive semi-definite matrices with appropriate dimensions, and $R(k)$ is symmetric positive definite matrix. Generally the linear quadratic optimal control problem is performed with six steps as following: starting with augmented performance index. Then performing the Lagrangian function. after that, constructing the Euler-Lagrange Equation. In the fourth step, introducing the Hamiltonian function. Then, finding the open loop optimal control. Then finally, constructing the state and costate system. These steps will be explained more in details next.

First step: Augmented Performance Index

Formulating the augmented cost function will be done by combining the cost function 2.2 with the system 2.1 utilizing Lagrange multiplier function (costate function) $\lambda(k+1)$ as following:

$$\begin{aligned} J_a = &\frac{1}{2}x'(k_f)F(k_f)x(k_f) \\ &+\frac{1}{2}\sum_{k=k_0}^{k_f-1}[x'(k)Q(k)x(k) + u'(k)R(k)u(k)] + \lambda(k+1)[A(k)x(k) + B(k)u(k) - x(k+1)] \end{aligned} \tag{2.3}$$

Second step: Lagrangian Function

Define the Lagrangian function as

$$\begin{aligned}\mathcal{L}(x(k), u(k), x(k+1), \lambda(k+1)) = \frac{1}{2}x'(k)Q(k)x(k) \\ +\frac{1}{2}u'(k)R(k)u(k) + \lambda'(k+1)[A(k)x(k) + B(k)u(k) - x(k+1)]\end{aligned} \tag{2.4}$$

Third step: Euler-Lagrange Equations

Then, applying Euler-Lagrange equation to the Lagrangian function ($\mathcal{L}$) with respect to the variables $x(k), u(k)$ and $\lambda(k+1)$. Then, one can get:

$$\frac{\partial \mathcal{L}(x^*(k), x^*(k+1), u^*(k), \lambda^*(k+1)}{\partial x^*(k)} + \frac{\partial \mathcal{L}(x^*(k-1), x^*(k), u^*(k-1), \lambda^*(k)}{\partial x^*(k)} = 0 \tag{2.5}$$

$$\frac{\partial \mathcal{L}(x^*(k), x^*(k+1), u^*(k), \lambda^*(k+1)}{\partial u^*(k)} + \frac{\partial \mathcal{L}(x^*(k-1), x^*(k), u^*(k-1), \lambda^*(k)}{\partial u^*(k)} = 0 \tag{2.6}$$

$$\frac{\partial \mathcal{L}(x^*(k), x^*(k+1), u^*(k), \lambda^*(k+1)}{\partial \lambda^*(k)} + \frac{\partial \mathcal{L}(x^*(k-1), x^*(k), u^*(k-1), \lambda^*(k)}{\partial \lambda^*(k)} = 0 \tag{2.7}$$

and the final (boundary) condition will be

$$\frac{\partial \mathcal{L}(x^*(k-1), x^*(k), u^*(k-1), \lambda^*(k)}{\partial x(k)} + \frac{\partial \mathcal{S}(x(k), (k)}{\partial x(k)}\delta x(k)\Big|_{k=k_0}^{k=k_f} = 0 \tag{2.8}$$

where from 2.2

$$S(x(k_f), k_f) = \frac{1}{2}x'(k_f)F(k_f)x(k_f) \tag{2.9}$$

Fourth step: Hamiltonian

Proceeding to get the solution in terms of Hamiltonian which is defined as following

$$\begin{aligned}\mathcal{H}(x^*(k), u^*(k), \lambda^*(k+1)) = \frac{1}{2}X^{*'}(k)Q(k)x^*(k) \\ +\frac{1}{2}u^{*'}(k)R(k)u^*(k) + \lambda^{*'}(k+1)[A(k)x^*(k) + B(k)u^*(k)]\end{aligned} \tag{2.10}$$

Thus the 2.4 and 2.10 are related as

$$\mathcal{L}(x^*(k), x^*(k+1), u^*(k), \lambda^*(k+1) = \mathcal{H}(x^*(k), u^*(k), \lambda^*(k+1)) - \lambda^*(k+1)x^*(k+1) \tag{2.11}$$

then using 2.5 to 2.7 with 2.11 to get the required conditions for the extremum in terms of Hamiltonian as follows:

$$\lambda^*(k) = \frac{\partial \mathcal{H}(x^*(k), u^*(k), \lambda^*(k+1)}{\partial x^*(k)} \tag{2.12}$$

$$0 = \frac{\partial \mathcal{H}(x^*(k), u^*(k), \lambda^*(k+1)}{\partial u^*(k)} \tag{2.13}$$

$$x^*(k) = \frac{\partial \mathcal{H}(x^*(k-1), u^*(k-1), \lambda^*(k)}{\partial \lambda^*(k)} \tag{2.14}$$

note 2.14 also can be written as

$$x^*(k+1) = \frac{\partial \mathcal{H}(x^*(k), u^*(k), \lambda^*(k+1)}{\partial \lambda^*(k+1)} \tag{2.15}$$

For the system 2.1 and performance index 2.2 the relations 2.12 2.13 and 2.15 are gotten for the state, costate and control, and can be written as

$$x^*(k+1) = A(k)x^*(k) + B(k)u^*(k) \tag{2.16}$$

$$\lambda^*(k) = Q(k)x^*(k) + A'(k)\lambda^*(k+1) \tag{2.17}$$

$$0 = R(k)u^*(k) + B'(k)\lambda^*(k+1) \tag{2.18}$$

Fifth step: Open-Loop Optimal Control

Then utilizing 2.18, the optimal control can be given as

$$u^*(k) = -R(k)^{-1}B'(k)\lambda^*(k+1) \tag{2.19}$$

Using the optimal control 2.19 and 2.16 one can get

$$x^*(k+1) = A(k)x^*(k) - B(k)R^{-1}(k)B'(k)\lambda^*(k+1) = A(k)x^*(k) - E(k)\lambda^*(k+1) \tag{2.20}$$

where $E(k) = B(k)R^{-1}(k)B'(k)$

Six step: State and Costate System

The state and costate of 2.17 and 2.20 will be as

$$\begin{bmatrix} x^*(k+1) \\ \lambda^*(k) \end{bmatrix} = \begin{bmatrix} A(k) & -E(k) \\ Q(k) & A'(k) \end{bmatrix} + \begin{bmatrix} x^*(k) \\ \lambda^*(k+1) \end{bmatrix} \tag{2.21}$$

2.2.1 Summary

This chapter introduced some preliminaries that will be useful for the subsequent chapters. Firstly, some preliminaries on LPV modeling are introduced. Then, preliminaries on optimal control problem formulation are introduced. For more general information on LPV one can consult [11, 12, 13, 14, 15] and more specific details will be presented in chapters three and five. More general details on LQR control problem can be found in [16] and more specific details on LQR-based MPC will be presented in chapter four.

CHAPTER 3

NOVEL LINEAR PARAMETER VARYING MODELING AND FLUTTER SUPPRESSION CONTROL OF A SMART AIRFOIL

3.1 Introduction

Active flutter suppression has been a critical research topic in aerospace applications for many decades. Reducing the aircraft weight, improving the aerodynamic efficiency, and increasing the critical flight speed continue to be the main thrusts for future aeronautical research, especially, as the emerging air vehicle structures become highly flexible, active flutter suppression becomes a key technical design requirement. There is a good body of design methods that are available in literature concerning suppression of flutter phenomena. Passive methods have been used to solve this problem for many years, however, these methods lead to increased aircraft mass, which is undesirable; see references [17, 18, 19, 20]. On the other hand, active control techniques can provide crucial and liable solutions that would increase the aircraft critical speed and suppress the oscillations, while enhancing flight efficiency and performance.

There are many active control techniques in literature for flutter suppression. Using piezoelectric actuation to control flutter was given by Han et al. [21], where experimental and numerical study were conducted for active flutter suppression of a sweptback cantilevered lifting surface. The finite element analysis, panel aerodynamics, and the minimum state space realization were used to develop the equations of motion, which were then used for system analysis and control design by utilizing the H_2 and μ-synthesis control techniques, and subsequently the flutter suppression performance was evaluated through wind tunnel tests. De Marqui et al. [22] introduced a flexible hanged arrangement for flutter investigation with hard wings in a wind tunnel. The wing model was a rectangular shape with NACA 0012 model and a rear edge surface mechanism control. They introduced aeroelastic model to emulate the aeroelastic behavior of the corresponding system. A full state feedback control was designed for this model to cancel flutter and retain the stability of the closed-loop system. Zhang

and Behal [23] introduced a continuous-time controller to suppress the aeroelastic oscillations of the wing shape in an unsteady aerodynamic incompressible flow environment. The flap connection force of the wing rear edge was used to represent the input and the pitching angle to represent the output. The numerical simulation results demonstrated the effectiveness in canceling aeroelastic vibrations in before and after flutter travel speed regions subjected to multiple external disturbances. Additional active flutter suppression techniques included electro-hydraulic mechanical actuation of control surface [24], reaction jets [25], and micro-flaps [26], to just name a few.

A special attention was given to Linear Parameter-Varying (LPV) modeling and control of an airfoil. For example, Wingerden et al. [27] developed a system identification algorithm for an LPV aeroelastic system outfitted with rear edge control surfaces, where a factorization was used to form a sensor-like depend on on the old inputs, outputs, and given aeroelastic data, and these sensor-like were then used to calculate the state progression to form LPV aeroelastic matrices of the system. As the algorithm can be utilized in a closed-loop arrangement, it can be used to flutter suppression problems. Barker and Balas [28] designed gain-scheduled depends on two-parameter LPV controllers for active flutter suppression of the Benchmark Active Control Technology (BACT) wing section at NASA Langley Research Center. The BACT wing dynamics shifts notably relative to the Mach number and dynamic pressure. The LPV gain-scheduled controllers depending on two parameters incorporated these alternations as well as limits on the rate of change of Mach number and dynamic pressure. The incorporation of the rate limits in the design procedure resulted in better performance over a vast range of operational conditions when compared to a previously designed gain-scheduled controller based on linear fractional transformation. Lau and Kerner [29] utilized a regular linear system for the control of a slim airfoil under subsonic flow. The two-dimensional airfoil was designed with three degrees of freedoms; flap, plunge, and pitch angle, resulting a linear system of six dimension with a control input at the flap hinge. The objective was to utilize feedback for stabilization the airfoil at its flutter speed or above it with many control procedures. Chen et al. [30] developed LPV aeroservoelastic system utilizing adaptive method with nonlinear aerodynamics. The LPV controller was able to suppress flutter with perfect

precision and validity, moreover, it provided a feasible tool for practical flutter flight tests. Balas et al. [31] utilizing LPV and H_∞ control design procedures to a body-freedom-flutter (BFF) airplane designed aeroservoelastic controllers and compared the performances of the controllers in both time domain and frequency simulations. Though the performance from the designed LPV controller was satisfactory, it did not attain the same degree of performance as from the H_∞ controllers. It should be noted that, from the literature review, one can see that the conventional LPV modeling considers the scheduling parameter as a physical parameter of the system matrices that changes with time [12, 32, 33, 34], whereas the quasi-LPV model [35, 36, 37] considers systems with one or more states in the system matrices.

In this chapter, we consider the smart airfoil model proposed by Swei and Jiang [2] (see Figure 3.1), to investigate the problem of active flutter suppression by utilizing the novel LPV modeling and control techniques of scaling the scheduling parameter. The smart airfoil is a two-dimensional airfoil with a groove along its chord that contains a moving light mass. The mass is allowed to move along the groove, and through its coupling with airfoil aerodynamics, it can control and suppress the pitching and plunging motion of the airfoil. The airspeed and position of the moving mass are considered as scheduling parameters in the LPV model and the mass position is utilized as a scheduling parameter in the control design. In this study, we propose to utilize the scheduling parameter as part of a scaling factor for the smart airfoil model. In particular, position of the moving mass is scaled and parametrized such that it is confined within the length of groove. Furthermore, we propose another novel LPV modeling technique by implementing a pair of nonlinear springs and dampers at both ends of the groove to gracefully prevent the moving mass from reaching the hard boundaries. To the best of authors' knowledge, the integration of scheduling parameter with the scaled control effector and the parametrization of boundary conditions are novel approach that have never been reported in the LPV control literature in the past. In this chapter, a number of LPV control models is developed to best describe the effect of boundary constraints and also to reduce the level of conservativeness. A full state feedback LPV gain-scheduling controller with guaranteed H_∞ output performance is proposed, in which the controller gains are obtained by

solving the numerically tractable Parameterized Linear Matrix Inequality (PLMI).

This chapter is organized as follows. Section 3.2 "LPV Modeling of a Smart Airfoil" presents the nonlinear model of the smart airfoil, the baseline LPV model (LPV-0), the LPV model with parameter scaling (LPV-1), the LPV model with nonlinear springs and dampers (LPV-2) at the boundaries, and the LPV model that is derived from LPV-2 model with less number of scheduling parameters (LPV-2A). Section 3.3 "LPV Controller Design" contains LPV problem formulation and controller design. Comparisons and simulation results are presented in Section 3.4 "Numerical Studies". Conclusions and future work are given in Section 3.5."Conclusion"

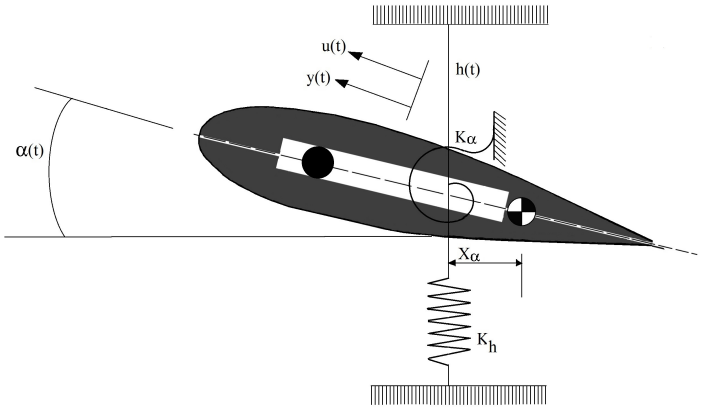


Figure 3.1: The smart airfoil model.

3.2 LPV Modeling of a Smart Airfoil

In this section, the mathematical model of the smart airfoil is presented. The linearized equations of motion of the airfoil aeroservoelastic model can be written as [2]

$$\begin{bmatrix} m+M & Mx_\alpha \\ Mx_\alpha & I_\alpha \end{bmatrix} \begin{bmatrix} \ddot{h}(t) \\ \ddot{\alpha}(t) \end{bmatrix} + \begin{bmatrix} K_h & 0 \\ 0 & K_\alpha \end{bmatrix} \begin{bmatrix} h(t) \\ \alpha(t) \end{bmatrix} = \begin{bmatrix} 0 \\ mg \end{bmatrix} y(t) + F_a(t)$$

$$m\ddot{y}(t) = mg\alpha(t) + u(t) \tag{3.1}$$

where $F_a(t)$ denotes the aerodynamic loading, m and M the moving mass and airfoil mass, respectively, additional variables and parameters used in equations (3.1) and (3.1) can be found in Figure 3.1 [2]. It is important to note that the position of moving mass $y(t)$ in equation (3.1) can be considered as the control input to the airfoil, whereas $u(t)$ in equation (3.1) can be considered as the control input to the moving mass m. The airfoil with such a control device is called "Smart Airfoil" [2]. The following quasi-steady aerodynamic load model [38] for $F_a(t)$ is adapted in this study,

$$F_a(t) = P\left(\begin{bmatrix} \frac{-1}{V} & 0 \\ \frac{e}{V} & 0 \end{bmatrix}\begin{bmatrix} \dot{h}(t) \\ \dot{\alpha}(t) \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 0 & e \end{bmatrix}\begin{bmatrix} h(t) \\ \alpha(t) \end{bmatrix}\right) \tag{3.2}$$

$$P = q_p c C_{L_\alpha} .$$

Now, substituting equation (3.2) into equation (3.1) and performing nondimensionalization for all the physical parameters, we obtain the nondimensionalized equations of motion for the smart airfoil model as follows,

$$\begin{bmatrix} 1+\beta & \bar{x}_\alpha \\ \bar{x}_\alpha & \bar{r}_\alpha^2 \end{bmatrix}\begin{bmatrix} \ddot{\bar{h}}(\tau) \\ \ddot{\alpha}(\tau) \end{bmatrix} + \begin{bmatrix} \frac{2\bar{V}}{\mu} & 0 \\ \frac{-2\bar{V}\bar{e}}{\mu} & 0 \end{bmatrix}\begin{bmatrix} \dot{\bar{h}}(\tau) \\ \dot{\alpha}(\tau) \end{bmatrix} + \begin{bmatrix} \frac{\omega_h^2}{\omega_\alpha^2} & \frac{2\bar{V}^2}{\mu} \\ 0 & \frac{-2\bar{V}^2\bar{e}}{\mu} + \bar{r}_\alpha^2 \end{bmatrix}\begin{bmatrix} \bar{h}(\tau) \\ \alpha(\tau) \end{bmatrix} = \begin{bmatrix} 0 \\ \beta\bar{g} \end{bmatrix}\bar{y}(\tau)$$

$$\ddot{\bar{y}}(\tau) = \bar{g}\alpha(\tau) + \bar{u}(\tau) \tag{3.3}$$

where $\tau = \omega_\alpha t$ is the nondimensional time. Note that to simplify the notation, the overhead "dot" in equations (3.3) and (3.3) represents the time derivative with respect to τ. When the flutter occurs, the plunging displacement h and pitching angle α are fed back in order to properly position the moving mass m, which generates a damping effect to the airfoil, hence reducing the flutter vibration and increasing the critical flutter speed.

3.2.1 LPV plant model: LPV-0

Rearranging equations (3.3) and (3.3) yields the following

$$\begin{bmatrix} \ddot{\bar{h}}(\tau) \\ \ddot{\alpha}(\tau) \end{bmatrix} + \begin{bmatrix} \frac{-2\bar{r}^2\bar{V}}{q\mu} - \frac{2\bar{V}\bar{e}\bar{x}_\alpha}{q\mu} & 0 \\ \frac{2\bar{x}_\alpha\bar{V}}{q\mu} + \frac{2\bar{V}\bar{e}(1+\beta)}{q\mu} & 0 \end{bmatrix} \begin{bmatrix} \dot{\bar{h}}(\tau) \\ \dot{\alpha}(\tau) \end{bmatrix} + \begin{bmatrix} \frac{-\bar{r}_\alpha^2\omega_h^2}{q\omega_\alpha^2} & \frac{-2\bar{r}_\alpha\bar{V}^2}{q\mu} - \frac{2\bar{V}^2\bar{e}\bar{x}_\alpha}{q\mu} + \frac{\bar{r}^2\bar{x}_\alpha}{q} \\ \frac{\bar{x}_\alpha\omega_h^2}{q\omega_\alpha^2} & \frac{2\bar{x}_\alpha\bar{V}^2}{q\mu} + \frac{2\bar{V}^2\bar{e}(1+\beta)}{q\mu} + \frac{\bar{r}^2(1+\beta)}{q} \end{bmatrix} \begin{bmatrix} \bar{h}(\tau) \\ \alpha(\tau) \end{bmatrix} = \begin{bmatrix} \frac{-\bar{x}_\alpha\beta g}{q} \\ \frac{(1+\beta)\beta\bar{g}}{q} \end{bmatrix} \bar{y}(\tau) \tag{3.4}$$

$$\ddot{\bar{y}}(\tau) = \bar{g}\alpha(\tau) + \bar{u}(\tau) \tag{3.5}$$

where $q = -[\bar{r}_\alpha^2(1+\beta) - \bar{x}_\alpha^2]$. Now, we define the augmented state x as

$$x = \begin{bmatrix} \bar{x}, & x_u \end{bmatrix}^T, \tag{3.6}$$

where

$$\bar{x} = \begin{bmatrix} \bar{h}, \alpha, \dot{\bar{h}}, \dot{\alpha} \end{bmatrix}^T \text{ and } x_u = \begin{bmatrix} \bar{y}, \dot{\bar{y}} \end{bmatrix}^T. \tag{3.7}$$

Then, equations (3.4) and (3.5) can be described in the state space representation as follows,

$$\begin{cases} \dot{x}(t) = A(\theta(t))x(t) + B_u(\theta(t))\bar{u}(t) \\ y(t) = C(\theta(t))x(t) + D_u(\theta(t))\bar{u}(t) \end{cases} \tag{3.8}$$

where $y(t)$ is the controlled output, and the system matrices $(A(\theta), B_u(\theta), C(\theta), D_u(\theta))$ are affine in θ and given by

$$A(\theta) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{\bar{r}_\alpha^2 \omega_h^2}{q\omega_\alpha^2} & \frac{2\bar{r}_\alpha^2\theta_3}{q\mu} + \frac{2\theta_3\bar{e}\bar{x}_\alpha}{q\mu} - \frac{\bar{r}_\alpha^2\bar{x}_\alpha}{q} & \frac{2\bar{r}_\alpha^2\theta_2}{q\mu} + \frac{2\theta_2\bar{e}\bar{x}_\alpha}{q\mu} & 0 & \frac{-\bar{x}_\alpha\beta}{q} & 0 \\ \frac{-\bar{x}_\alpha\omega_h^2}{q\omega_\alpha^2} & \frac{-2\bar{x}_\alpha\theta_3}{q\mu} - \frac{2\theta_3\bar{e}(1+\beta)}{q\mu} - \frac{\bar{r}_\alpha^2(1+\beta)}{q} & \frac{-2\bar{x}\theta_2}{q\mu} - \frac{2\theta_2\bar{e}(1+\beta)}{q\mu} & 0 & \frac{\beta(1+\beta)}{q} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \bar{g} & 0 & 0 & a_{65} & a_{66} \end{bmatrix},$$

$$B_u(\theta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & b_6 \end{bmatrix}', \qquad C(\theta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \qquad D_u(\theta) = 0, \tag{3.9}$$

where $\theta := [\theta_1, \theta_2, \theta_3]^T$, and θ_1 is the scheduling parameter utilized to properly scale and constrain the position of the moving mass at the boundaries, θ_2 and θ_3 are the scheduling parameters representing, respectively, the normalized airspeed $\bar{V}$ and its square $\bar{V}^2$. In this study, we consider $\theta_2 \in [0.5, 2.92]$, hence, $\theta_3 \in [0.25, 8.52]$. Furthermore, the parameters a_{65} and a_{66} are used to represent nonlinear spring and damping effects at the boundaries. It should be emphasized that θ_2 and θ_3 are not independent parameters, therefore inevitably it would introduce conservativeness in the modeling process. However, in Section 3.2.4 an approximate approach is proposed.

To formulate the baseline LPV-0 plant model, we set the parameters $b_6 = 1$ and $a_{65} = 0, a_{66} = 0$. This LPV-0 model will be used to assess the baseline closed-loop system performance.

3.2.2 LPV control design model: LPV-1

The proposed LPV-1 model is based on the LPV-0 model described in equations (3.8) and (3.9) with $a_{65} = 0$, $a_{66} = 0$, but $b_6 = \theta_1$, where θ_1 is a function of $\bar{y}$, i.e. $\theta_1 = f(\bar{y})$, which is used to constrain the moving mass. In particular, θ_1 is devised such that

$$\begin{cases} \theta_1 = 1, & \text{if } \bar{y} \in [-0.35,\ 0.35] \\ \theta_1 \bar{y} = 0.35, & \text{if } \bar{y} > 0.35 \\ \theta_1 \bar{y} = -0.35, & \text{if } \bar{y} < -0.35 \end{cases}$$

It is clear from the first condition above that when $\bar{y} \in [-0.35,\ 0.35]$ and $\theta_1 = 1$, the LPV-1 model will be equivalent to the LPV-0 model. The purpose of the second and third conditions is to impose a constraint on the moving mass when it travels beyond ±0.35, by modulating the control gains through θ_1. The variation of θ_1 as function of the position of the moving mass is illustrated in Figure 3.2. To constrain the mass at ±0.5 the damping function must be active starting at ±0.35. This approach may introduce a slight conservativeness to the control design, it is intuitively appealing and proved to be effective.

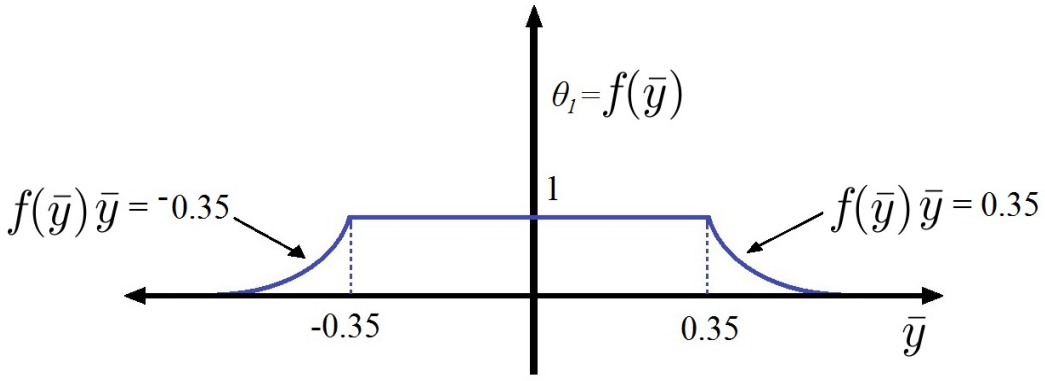


Figure 3.2: A saturation function.

3.2.3 LPV control design model with nonlinear springs and dampers: LPV-2

The second proposed LPV control design model contains nonlinear springs and dampers at both ends of the groove. They are to gradually stop and reverse the motion of the control mass. Figure 3.3 shows an illustration of the proposed boundary constraints with nonlinear springs and dampers, and as indicated earlier, the scheduling parameter θ_1, which is a function of $\bar{y}$, is implemented to scale the control effector.

$$\begin{cases} \theta_1 = 0, & \text{if } \bar{y} \in [-0.35,\ 0.35] \\ \theta_1 = \bar{y} - 0.35, & \text{if } \bar{y} > 0.35 \\ \theta_1 = -0.35 - \bar{y}, & \text{if } \bar{y} < -0.35 \end{cases}$$

It should be emphasized that the choice of K_{damp} and C_{damp} depends on the location from which the spring and damping effect are expected to engage, i.e. the closer to ±0.5, the higher the K_{damp} and C_{damp} value.

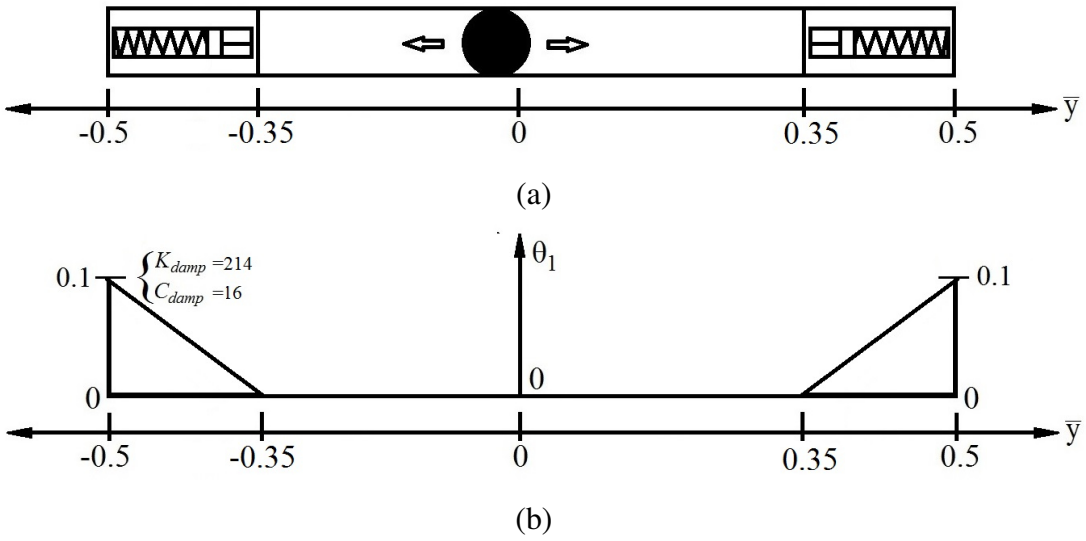


Figure 3.3: (a) Airfoil groove with nonlinear springs and dampers at the boundaries; (b) θ_1 as function of moving mass position.

3.2.4 LPV control design model with two scheduling parameters: LPV-2A

A third LPV control design model is modified from LPV-2, in which the parameter θ_3 is approximated as a function of θ_2, hence the number of scheduling parameters is reduced to two. Since θ_2 and θ_3 represent the velocity and its squared value, they are not independent and will introduce conservativeness due to over bounding the parameters. Kwiatkowski et al. [39] suggested a method to reduce conservativeness by taking the intersection between the two polytopes. In this section, we propose to approximate θ_3 as a linear function of θ_2 with slope β_1, as shown in Figure 3.4.

Therefore, the resulting LPV model, denoted as LPV-2A, contains two scheduling parameters: θ_1 and θ_2.

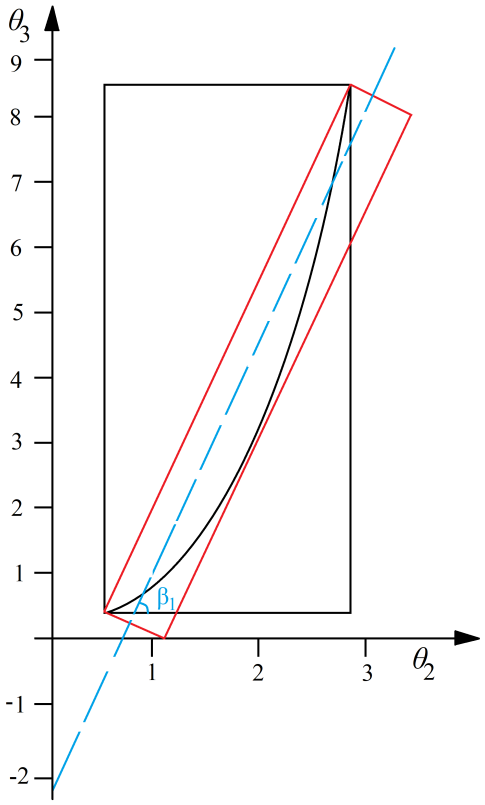


Figure 3.4: Proposed method to reduce conservativeness.

Recall the affine matrix $A(\theta)$ is of the form

$$A(\theta) = A_0 + \theta_1 A_1 + \theta_2 A_2 + \theta_3 A_3 \tag{3.10}$$

From Figure 3.4, θ_3 can be approximated as

$$\theta_3 = \theta_{30} + \theta_2 \tan(\beta_1) \tag{3.11}$$

Substituting equation (3.11) into equation (3.10) yields

$$A(\theta) = \underbrace{A_0 + \theta_{30} A_3}_{A_{0new}} + \theta_1 A_1 + \theta_2 \underbrace{(A_2 + \tan(\beta_1) A_3)}_{A_{2new}} \tag{3.12}$$

Therefore, the affine matrix $A(\theta)$ in the LPV-2A model can be described by

$$A(\theta) = A_{0new} + \theta_1 A_1 + \theta_2 A_{2new} \tag{3.13}$$

3.3 LPV Controller Design

In this section, we develop state feedback based gain-scheduling LPV controllers for LPV-0, LPV-1, LPV-2, and LPV-2A models. The goal of LPV control design is to achieve active flutter suppression with guaranteed H_∞ output performance at varying airspeed, through modulating the dynamics of moving mass in the smart airfoil model, subject to actuation constraints and external disturbances. In particular, we consider the following LPV systems,

$$\begin{cases} \dot{x}(t) = A(\theta(t))x(t) + B_u(\theta(t))\bar{u}(t) + B_w(\theta(t))w(t) \\ y(t) = C(\theta(t))x(t) + D_u(\theta(t))\bar{u}(t) + D_w(\theta(t))w(t) \end{cases} \tag{3.14}$$

where $w(t)$ denotes the disturbance input.

3.3.1 Problem formulation

The LPV model described in equation (3.14) is assumed to have affine parameters. For instance, matrix $A(\theta)$ can be represented by

$$A(\theta(t)) = A_0 + \sum_{i=1}^{q} \theta_i(t) A_i \tag{3.15}$$

where A_0 and A_i are constant matrices, and q denotes the number of scheduling parameters. Scheduling parameter vector $\theta(t)$ is defined as

$$\theta(t) = [\theta_1(t), \theta_2(t), \theta_3(t), ..., \theta_q(t)]^T \tag{3.16}$$

and each θ_i is bounded by

$$\eta_{1,i} \leq \theta_i \leq \eta_{2,i} \tag{3.17}$$

where $\eta_{1,i}$ and $\eta_{2,i}$ denote the upper and lower bound. In addition, these scheduling parameters also have the rate bound given by

$$\mu_{1,i} \leq \dot{\theta}_i \leq \mu_{2,i} \tag{3.18}$$

The proposed LPV control law is a state feedback controller as

$$\bar{u}(t) = K(\theta)x(t) \tag{3.19}$$

where $K(\theta)$ asymptotically stabilizes the closed-loop system subject to the $\mathcal{H}_\infty$-norm constraint from the exogenous input w to the measured output y over the entire parameter variation range. Substituting controller equation (3.19) into equation (3.14) yields the closed-loop system representation given by

$$\begin{cases} \dot{x}(t) = A_c(\theta)x(t) + B_w(\theta)w(t) \\ y(t) = C_c(\theta)x(t) + D_w(\theta)w(t) \end{cases} \tag{3.20}$$

where

$$\begin{aligned} A_c(\theta) &= A(\theta) + B_u(\theta)K(\theta) \\ C_c(\theta) &= C(\theta) + D_u(\theta)K(\theta) \end{aligned} \tag{3.21}$$

It should be noted that though $(A(\theta), B_u(\theta), C(\theta), D_u(\theta))$ matrices are all affine in θ, the gain-scheduling feedback $K(\theta)$ is not. Before proceeding further, we introduce the following definitions.

Definition 1: [13, 14] A unit simplex Θ_r is a polytope of r vertices defined as

$$\Theta_r = \left\{ \rho = [\rho_1, \cdots, \rho_r] : \sum_{i=1}^{r} \rho_i = 1, \rho_i \geq 0, i = 1, 2, .., r \right\}$$

since variable ρ_i changes inside a unit simplex Θ_r.

Definition 2: [15] The Cartesian product of a limited number of simplexes is a multi-simplex Θ. For instance, if there are q simplexes, then

$$\Theta = \Theta_{N_1} \times \Theta_{N_2} \times \Theta_{N_3} \times \cdots \Theta_{N_q} = \prod_{i=1}^{q} \Theta_{N_i},$$

where the dimension of multi-simplex Θ is the index $N = (N_1, \ldots, N_q)$, such that ρ in Θ represents $(\rho_1, \rho_2, ..., \rho_q)$ and each ρ_i in Θ_{Ni} represents $(\rho_i(1), \rho_i(2), ..., \rho_i(r))$.

3.3.1.1 Transferring from affine to multi-simplex

To formulate a convex control design problem, we first need to perform a transformation on the system matrices, from the affine parameter space θ to the multi-simplex convex space Θ. Each affine scheduling parameter θ_i is transferred over a unit simplex ρ_i of two vertices as follows [40],

$$\begin{aligned} \rho_i(1) &= \frac{\theta_i + \eta_{2,i}}{2\eta_{2,i}} \longrightarrow \theta_i = 2\eta_{2,i}\rho_i(1) - \eta_{2,i} \\ \rho_i(2) &= 1 - \rho_i(1) = 1 - \frac{\theta_i + \eta_{2,i}}{2\eta_{2,i}} = \frac{\eta_{2,i} - \theta_i}{2\eta_{2,i}} \\ \rho_i &= (a_i(1), \rho_i(2)) \in \Theta_2, \ \forall\, i = 1, 2, \cdots, q\,. \end{aligned} \tag{3.22}$$

With this transformation, the affine scheduling parameters in equation (3.14) can be converted into a system with multi-simplex parameters, where the multi-simplex variables are defined as

$$\rho \in \Theta = \underbrace{\Theta_2 \times \Theta_2 \times \cdots \Theta_2}_{q}\,.$$

Moreover, for three scheduling parameters, i.e. $q = 3$ as the case of our current consideration, the homogeneous term in the multi-simplex variables can be written as

$$\rho_1 = (\rho_1(1), \rho_1(2)),\ \rho_2 = (\rho_2(1), \rho_2(2)),\ \rho_3 = (\rho_3(1), \rho_3(2)).$$

Hence, by utilizing the above transformation, the LPV system equation (3.14), which was described in affine parameter space θ, can be equivalently described in multi-simplex convex space Θ as follows,

$$\begin{cases} \dot{x}(t) = A(\rho)x(t) + B_u(\rho)\bar{u}(t) + B_w(\rho)w(t) \\ y(t) = C(\rho)x(t) + D_u(\rho)\bar{u}(t) + D_w(\rho)w(t) \end{cases} \tag{3.23}$$

Furthermore, the proposed state feedback gain scheduling controller has the form

$$\bar{u}(t) = K(\rho)x(t) \tag{3.24}$$

where $K(\rho)$ is to be determined. Therefore, the closed-loop system representation in multi-simplex space is given by

$$\begin{cases} \dot{x}(t) = A_c(\rho)x(t) + B_w(\rho)w(t) \\ y(t) = C_c(\rho)x(t) + D_w(\rho)w(t) \end{cases} \tag{3.25}$$

where

$$\begin{aligned} A_c(\rho) &= A(\rho) + B_u(\rho)K(\rho) \\ C_c(\rho) &= C(\rho) + D_u(\rho)K(\rho) \end{aligned} \tag{3.26}$$

In addition, the rate changes of the scheduling parameters in the unit simplex can also be described by

$$\dot{\rho}_i(1) + \dot{\rho}_i(2) = 0, \qquad i = 1, 2, 3\,. \tag{3.27}$$

From equations (3.18) and (3.22) we can derive the rate bounds between affine scheduling parameters and multi-simplex variables, and they are given by

$$\frac{\mu_{1,i}}{2\eta_{2,i}} \leq \dot{\rho}_i(1) \leq \frac{\mu_{2,i}}{2\eta_{2,i}} \tag{3.28}$$

where $\mu_{2,i}$ and $\mu_{1,i}$ are rate bounds given in equation (3.18), and $\eta_{2,i}$ is the upper bound of θ_i given in equation (3.17). Furthermore, from equation (3.27), we obtain that $\dot{\rho}_i(2) = -\dot{\rho}_i(1)$.

3.3.1.2 $H\infty$ control problem

The proposed LPV control problem can be stated as follows: Design a gain-scheduling state feedback controller of the form equation (3.24), such that for given $\gamma_\infty > 0$ and any given pair $(\rho_i, \dot{\rho}_i)$; $i = 1, 2, 3$, the closed-loop system equation (3.25) is stabilized and satisfies the following

H_∞ performance constraint,

$$\sup_{(\rho_i,\dot{\rho}_i)} \sup_{\substack{w\in l_2 \\ w\neq 0}} \frac{||z||_2}{||w||_2} < \gamma_\infty \tag{3.29}$$

To synthesize the $\mathcal{H}_\infty$ control problem, we utilize the following theorem.

Theorem 1 *[12]: For any given pair $(\rho_i, \dot{\rho}_i)$, $i = 1, 2, 3$, the closed-loop system equation (3.25) will be stabilized by the gain scheduling controller $K(\rho) = Z(\rho)G(\rho)^{-1}$ with guaranteed $\mathcal{H}_\infty$ performance level γ_∞, if there exist a scalar $\epsilon > 0$, a continuously differentiable symmetric and positive-definite matrix $P(\rho)$, and matrices $G(\rho)$ and $Z(\rho)$ satisfying the following parameterized linear matrix inequality (PLMI),*

$$\begin{bmatrix} Q(\rho) & * & * & * \\ P(\rho) - G(\rho) + \epsilon(A(\rho)G(\rho) + B_u(\rho)Z(\rho))' & -\epsilon(G(\rho)G(\rho)') & * & * \\ C(\rho)G(\rho) + D_u(\rho)Z(\rho) & \epsilon C(\rho)G(\rho) + \epsilon D_u(\rho)Z(\rho) & -I & * \\ B_w(\rho)' & 0_{r\times n} & D_w(\rho)' & -\gamma_\infty^2 I \end{bmatrix} < 0 \tag{3.30}$$

where $*$ denotes symmetric entry and $Q(\rho) = A(\rho)G(\rho) + B_u(\rho)Z(\rho) + G(\rho)'A(\rho)' + Z(\rho)'B_u(\rho)' + \frac{\partial P(\rho)}{\partial \rho}\dot{\rho}$.

It should be noted that though the $\mathcal{H}_\infty$ control problem is presented in multi-simplex convex space Θ, the actual gain-scheduling controllers are implemented in affine θ domain. Roughly speaking, in order to compute the affine feedback gain matrix described in equation (3.19), we first need to utilize Theorem 1 to synthesize the solutions $Z(\rho)$ and $G(\rho)$ at the vertices of multi-simplex space Θ, followed by the inverse transformation process [12] to convert the multi-simplex solutions into affine representations $Z(\theta)$ and $G(\theta)$. The detail procedure is given below.

For three scheduling parameters, the parameters $Z(\theta)$ and $G(\theta)$ in affine θ domain can be

described by

$$\begin{aligned} Z(\theta) &= Z_0 + \theta_1 Z_1 + \theta_2 Z_2 + \theta_3 Z_3, \\ G(\theta) &= G_0 + \theta_1 G_1 + \theta_2 G_2 + \theta_3 G_3 \end{aligned} \tag{3.31}$$

where $(Z_0,\ G_0)$ and $(Z_i,\ G_i)$, $i = 1, 2, 3$, are constant matrices to be determined in the sequel. Moreover, given $Z(\theta)$ and $G(\theta)$, the affine gain matrix $K(\theta)$ in equation (3.19) will then be given by

$$K(\theta) = Z(\theta)G^{-1}(\theta)\,. \tag{3.32}$$

Therefore, it is clear that the control gain $K(\theta)$ is not an affine function of θ. For three scheduling parameters, there are 8 polytopic solutions for $Z(\rho)$ and $G(\rho)$ in equation (3.30). Let cZ_{ijk} and cG_{ijk}, $i, j, k = 1, 2$, to denote these solutions, then $Z(\rho)$ and $G(\rho)$ can be represented as

$$\begin{aligned} X(\rho) = {} & \rho_1(1)\rho_2(1)\rho_3(1)cX_{111} + \rho_1(1)\rho_2(1)\rho_3(2)cX_{112} \\ & +\rho_1(1)\rho_2(2)\rho_3(1)cX_{121} + \rho_1(1)\rho_2(2)\rho_3(2)cX_{122} \\ & +\rho_1(2)\rho_2(1)\rho_3(1)cX_{211} + \rho_1(2)\rho_2(1)\rho_3(2)cX_{212} \\ & +\rho_1(2)\rho_2(2)\rho_3(1)cX_{221} + \rho_1(2)\rho_2(2)\rho_3(2)cX_{222} \\ = {} & X(\theta) \end{aligned} \tag{3.33}$$

where $(X,\ cX)$ denote $(Z,\ cZ)$ or $(G,\ cG)$. Hence, it follows from the inverse transformation process given in [12] that

$$\begin{aligned} X_0 &= \frac{1}{64} \sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \sum_{j_3=1}^{2} cX_{j_1 j_2 j_3}, \\ X_i &= \frac{1}{64\eta_{2,i}} \sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \sum_{j_3=1}^{2} (-1)^{j_i+i}\, cX_{j_1 j_2 j_3} \end{aligned} \tag{3.34}$$

where the constant matrices $(X_0,\ X_i)$ denote $(Z_0,\ Z_i)$ or $(G_0,\ G_i)$, and $cX_{j_1 j_2 j_3}$ denotes the polytopic solutions $cZ_{j_1 j_2 j_3}$ or $cG_{j_1 j_2 j_3}$ from Theorem 1. Now, substituting $(Z_0,\ Z_i)$ and $(G_0,\ G_i)$ into equation (3.31) yields $Z(\theta)$ and $G(\theta)$, hence the feedback gain matrix $K(\theta)$ can be obtained from equation (3.32). The inverse transformation formulas for any number of scheduling parameters

$q > 3$ can be found in [12], [13], and [40]. For the purpose of demonstration, a list of (Z_0, Z_i) and (G_0, G_i) matrices for various LPV models are presented in the Appendix.

It should be noted that PLMI is an infinite dimensional linear matrix inequality (LMI), which is in general difficult to solve. Many efficient solvers are available in dealing with such problem; for instance, see references [41, 42, 43, 44]. This work adapts the relaxation method for PLMIs relaxation [13], where it treats PLMIs with multi-simplex parameters. Numerically, the ROLMIP package [45] along with YALMIP [11] using SeDuMi [46] solver are used to solve the convex optimization problem.

Generally, the controller solution process for the LPV control synthesis in this chapter considers seven steps. Firstly, scaling scheduling parameter introduced. Secondly, change of variables from affine representation domain to the multi-simplex (convex) domain is performed. Thirdly, modeling the rates of variation of scheduling parameters in convex domain. In the fourth stage, performing the PLMI synthesis conditions. The fifth stage is to perform PLMI relaxation to transfer the infinite dimension PLMI into finite dimension LMI to be solved using optimization package. Then the six stage involve to utilize inverse transformation to transfer from the multi-simplex representation into the affine representation. Then finally the seven stage is the controller implementation. The steps are shown in the following diagram Figure 3.5.

3.3.2 PLMI relaxation

As mentioned above, that the synthesis condition 3.30 is a parameter dependent LMI (PLMI) which is an infinite dimension LMI and needs to relaxed to finite dimension to be solved in convex optimization. There are many relaxation methods in the literature to relax a PLMI such as multi-convexity technique [47], SOS [48, 49], semidefinite programming relaxations for the generalized problem of moments [50], griding method [51] vertex method [52, 53] dilated LMI [54] multi-convexity [55] slack variable [44] and many other methods. Most of them either provided conservative solution, large number of divisions to reach a good results and/or depend on constant

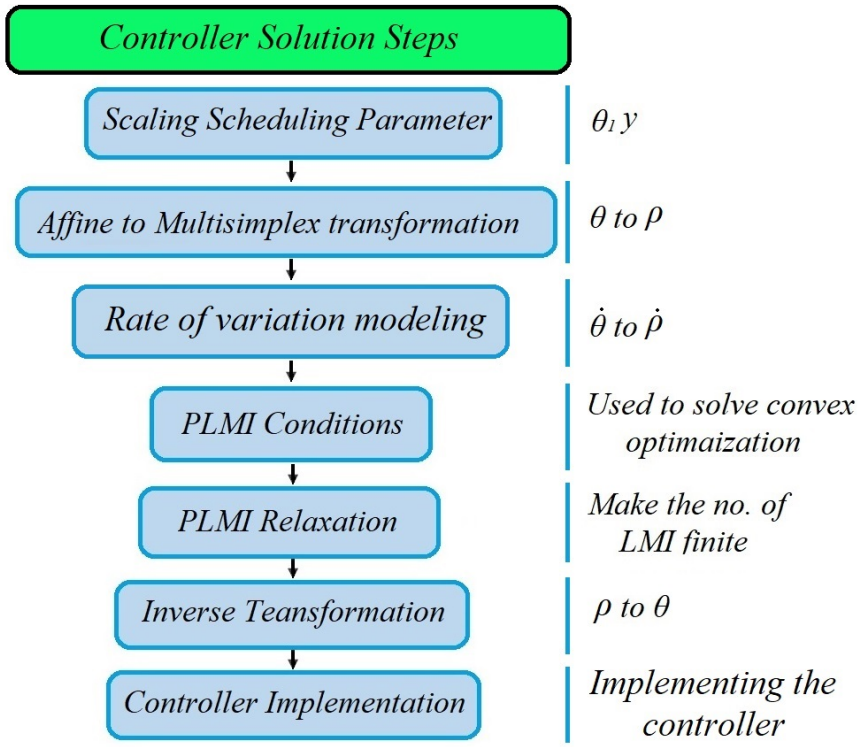


Figure 3.5: Controller steps.

Lyapunov functions that prevents using parameter dependent Lyapunov function which will lead for more conservativeness solution. For that, the method of [13] is used in this work which overcomes the shortcomings of the mentioned methods. This section will present the solution paradigm of the method of handling the PLMI and gives general examples for more explanation and most of the formulation are from [13]. Consider the PLMI can be written as

$$A(\rho(t))^T P(\rho(t)) + P(\rho(t))A(\rho(t)) + \sum_{i=1}^{m}\sum_{j=1}^{N_i} \frac{\partial P(\rho(t))}{\partial \rho_{ij}(t)}\dot{\rho}_{ij}(t) < 0 \tag{3.35}$$

for all $\rho(t) \in \Lambda$ and $\dot{\rho} \in \Omega = \Omega_1 \times ...m$. where $\rho_i(t), i = 1, ..., m$ are independent from each other this applied also to their derivatives.

$$\dot{\rho}_{i1}(t) + \dot{\rho}_{i2}(t) + ... + \dot{\rho}_{iN_i}(t) = 0 \tag{3.36}$$

Let $G^{(i)}$ denotes the i-th column of matrix G. The sets $\Omega_i, i = 1, ..., m$ are defined as

$$\Omega_i = \delta \in \mathcal{R}^{N_i} : \delta = \sum_{l=1}^{M_i} \eta_{i,l} H_i^{(l)}, \sum_{l=1}^{N_i} H_i(l,j) = 0, j = 1, ..., N_i, \quad \eta_i \in \Lambda_{M_i} \tag{3.37}$$

For example [13], if

$$-1 \leq \dot{\rho}_{11(t)} \leq 1, -1 \leq \dot{\rho}_{12(t)} \leq 1, -2 \leq \dot{\rho}_{13(t)} \leq 2 \tag{3.38}$$

then the solution of 3.36 by 3.38 will give results as

$$(1, 1, -2), (-1, -1, 2), (1, -1, 0), (-1, 1, 0)$$

the convex combination of these results will give

$$\begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix} \eta_{11} + \begin{bmatrix} -1 \\ -1 \\ 2 \end{bmatrix} \eta_{12} + \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \eta_{13} + \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \eta_{14}, \quad \eta_i \in \Lambda_4 \tag{3.39}$$

in other form

$$\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ -2 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} \eta_{11} \\ \eta_{12} \\ \eta_{13} \\ \eta_{14} \end{bmatrix}, \qquad H_1 \in \mathcal{R}^{N_1 \times M_1} \tag{3.40}$$

where N_1=3 is the number of parameters in Λ_1 and M_1=4 is the number of solutions of 3.36 by 3.38. Notice that the number M_1 is not known a priori, where it is determined by the number of extremal solutions. But it could be different from M_1=4, if distinct bounds were considered in 3.38. for instance, if in 3.38 one considers $-1 \leq \dot{\rho}_{13(t)} \leq 1$, the number of extremal solutions will be M_1=6.

Since $\rho(t) \in \Lambda$ implies $\dot{\rho}(t) \in \Omega$for all $t0$ and using the definition of a generic $\rho_{ij}(t)$ belonging to Ω_i, inequality 3.35 can be rewritten as

$$A(\rho(t))^T P(\rho(t)) + P(\rho(t))A(\rho(t)) + \sum_{i=1}^{m}\sum_{j=1}^{N_i} \frac{\partial P(\rho(t))}{\partial \rho_{ij}(t)} \sum_{l=1}^{M_i} \eta_{il} H_i(j,l) < 0 \tag{3.41}$$

For Λ-homogeneous functions $P(\rho)$ and $A(\rho)$ of partial degrees $g = (g1, ..., gm)$ and $r = (r1, ..., rm)$ respectively, the total degree of the first two terms $A(\rho)^T P(\rho) + P(\rho)A(\rho)$ will be $\bar{g} = (g1 + r1, g2 + r2, ..., gm + rm)$. Then, the next step is to homogenize the third term in ρ. The general expression for the derivative of the Lyapunov matrix $P(\rho)$ with respect to the $i-th$ component of the multi-simplex, $i = 1, ..., m$ and then with respect to its $j - th$ component, $j = 1, ..., N_i$ is given by

$$\begin{aligned}\frac{\partial P(\rho)}{\partial \rho_{ij}} &= \sum_{k \in K_N(g)} k_{ij} \rho_1^{k_1} ... \rho_{i1}^{k_{i1}} .... \rho_{ij}^{k_{ij}-1} ..... \rho_{iN_i}^{k_{iN_i}} .... \rho_m^{k_m} P_k \\ &= \sum_{k \in K_N(g - e_{i|m})} \rho^k ((k + e_{i|m} \oplus e_{j|N_i})_{ij} P_{k + e_{i|m} \oplus e_{j|N_i}})\end{aligned} \tag{3.42}$$

where $e_{i|m}$ is the vector of dimension m with zero components, except 1 in the $i - th$ position. To fit (on ρ) with the partial degrees $\bar{g}$, the following homogenization is necessary:

$$\begin{aligned}\sum_{i=1}^{m} (\rho_{i1} + ... + \rho_{iN_i})^{r_i+1} \sum_{j=1}^{N_i} \frac{\partial P(\rho)}{\partial \rho_{ij}} = \sum_{i=1}^{m}\sum_{j=1}^{N_i} \sum_{k \in K_{N(g+r)}} \rho^k \Bigg(\sum_{\substack{\hat{k} \in K_{N(r+e_{i|m})} \\ \hat{k} \le k}} \\ \times \frac{(r_i + 1)!}{\pi(\hat{k}_i)} \Big((k - \hat{k} + e_{i|m} \oplus e_{j|N_i})_{ij} P_{k - \hat{k} + e_{i|m} \oplus e_{j|N_i}} \Big) \Bigg)\end{aligned} \tag{3.43}$$

where $\pi(k_i) = (k_{i1}!)(k_{i2}!)(k_{iNi}!)$. Now, the third term of 3.41 must be homogenized to become multi-affine on η. This is done as follows

$$\prod_{\substack{p=1\\p\neq i}}^{m}(\eta_{p1}+...+\eta_{pM_p})\sum_{l=1}^{M_i}\eta_{il}H_i(i,l)=\sum_{p_1=1}^{M_1}...\sum_{p_i=1}^{M_i}....\sum_{p_m=1}^{M_m}\eta_{1p_1}...\eta_{ip_i}....\eta_{mp_m}H_i(j,p_i) \quad (3.44)$$

Regarding 3.43 and 3.44, the third term in the left-hand side of 3.41 will be as following

$$\sum_{i=1}^{m}\sum_{j=1}^{N_i}\frac{\partial P(\rho)}{\partial\rho_{ij}}\sum_{l=1}^{M_i}\eta_{il}H_i(j,l)=\sum_{P_1=1}^{M_1}...\sum_{P_m}^{M_m}\eta_{1p_1}...\eta_{mp_m}\Bigg(\sum_{k\in K_{N(g+r)}}\rho^k\sum_{i=1}^{m}\sum_{j=1}^{N_i}\sum_{\substack{\hat{k}\in K_{N(r+e_{i|m})}\\ \hat{k}\leq k}}\frac{(r_i+1)!}{\pi(\hat{k}_i)}\times\Big((k-\hat{k}+e_{i|m}\oplus e_{j|N_i})_{ij}P_{k-\hat{k}+e_{i|m}\oplus e_{j|N_i}}\Big)H_i(j,p_i)\Bigg) \quad (3.45)$$

Notice that

$$\begin{aligned}&\prod_{p=1}^{m}(\eta_{p1}+...+\eta_{pM_p})A(\rho(t))^TP(\rho(t))+P(\rho(t))A(\rho(t))\\&\quad=\sum_{p_1=1}^{M_1}...\sum_{p_i=1}^{M_i}....\sum_{p_m=1}^{M_m}\eta_{1p_1}...\eta_{ip_i}....\eta_{mp_m}\times A(\rho(t))^TP(\rho(t))+P(\rho(t))A(\rho(t))\end{aligned} \quad (3.46)$$

then finally, 3.41 can be tested since all terms have the same partial degrees on both ρ and η. The following theorem presents LMI relaxations of increasing precision for the problem of robust stability analysis of matrix $A(\rho)$ with parameters $\rho\in\Lambda,\dot{\rho}\in\Omega$.

Theorem 2 *[13] If Λ a multi-simplex of dimension $N=(N_1,...,N_m)$. The Λ-homogeneous polynomial matrix $A(\rho)$ of partial degrees $r=(r_1,...,r_m)$ is robustly stable $\forall\rho\in\Lambda,\dot{\rho}\in\Omega$ if there exist $g=(g_1,...,g_m),k\in K_N(g)$ and matrices $P_k\in\mathcal{S}^n$ such that $P_k>0_n,\forall k\in K_N(g)$ and for all $(i_1,...,i_m)\in 1,....,M_1\times...\times 1,...,M_m$ the following LMIs are verified*

$$T_k = \sum_{\substack{\tilde{k}\in K_N(r)\\ \tilde{k}\le k}} A_{\tilde{k}}^T P_{k-\tilde{k}} + P_{k-\tilde{k}} A_{\tilde{k}} + \Xi_k < 0_n, \forall k \in K_N(g+r) \tag{3.47}$$

where

$$\Xi_k = \sum_{i=1}^{m}\sum_{j=1}^{N_i} \sum_{\substack{\hat{k}\in K_{N(r+e_{i|m})}\\ \hat{k}\le k}} \frac{(r_i+1)!}{\pi(\hat{k}_i)} \times \left((k-\hat{k}+e_{i|m}\oplus e_{j|N_i})_{ij} P_{k-\hat{k}+e_{i|m}\oplus e_{j|N_i}}\right) H_i(j,p_i) \tag{3.48}$$

for the proof one can consult [13].

3.4 Numerical Studies

The smart airfoil parameters used in this study are given in Table 3.1.

Table 3.1: [1, 2] Parameters used for the smart airfoil example.

Parameter	Value	Units	Parameter	Value	Units
b (m)	0.127	(m)	initial $\bar{h}$	0	Nondimensionalized
$\bar{e}$	0.35	Nondimensionalized	initial $\dot{\bar{h}}$	0	Nondimensionalized
$\bar{r}_\alpha^2$	0.388	Nondimensionalized	α_S	0.01	(rad)
$\bar{x}_\alpha$	0.25	Nondimensionalized	α_L	0.6	(rad)
airspeed $\bar{V}$	2.92	Nondimensionalized	initial $\dot{\alpha}$	0	(rad/sec)
ω_α	64.1	(rad/sec)	β	0.01	Nondimensional
ω_h	55.9	(rad/sec)	μ	152	Nondimensional

In addition, in equation (3.14) we choose $B_w = [0\,0\,0\,0\,0\,0.1]^T$ and $D_w = 0$. The baseline LPV-0 controller and the proposed LPV-1 and LPV-2 controllers are designed based upon their corresponding models, with $\gamma_\infty = 0.0055$ in Theorem 1. For the purpose of comparison, the scheduling parameter θ_2 is varied from 4 m/s to 23.8 m/s, and θ_3 from 16 m^2/s^2 to 566.44 m^2/s^2. Note that the LPV-0 controller is designed using the (unsaturated) LPV-0 model. Hence, the resulting LPV-0 controller is in the form of equation (3.32) without the θ_1 terms in equation (3.31).

In developing two-scheduling parameters LPV-2A control model, we choose $\beta_1 = 73.3°$ in Figure 3.4 and $\theta_{30} = -1.2$. The scheduling gain matrix $K(\theta) = Z(\theta)G(\theta)^{-1}$ and the affine

parameters $Z(\theta)$ and $G(\theta)$ are given by

$$\begin{aligned} Z(\theta) &= Z_0 + \theta_1 Z_1 + \theta_2 Z_2, \\ G(\theta) &= G_0 + \theta_1 G_1 + \theta_2 G_2 \end{aligned} \tag{3.49}$$

where (Z_0, G_0) and (Z_i, G_i), $i = 1, 2$, are computed by following the similar procedure presented in Section 3.3.1.2. The design of LPV-2A control model is to reduce the number of scheduling parameters from three to two, by representing θ_3 in terms of θ_2, hence reducing the model conservativeness. Note that γ_∞ is a measure of robustness performance. The minimum achievable γ_∞ for the LPV-2 control is 0.00029, whereas for the LPV-2A control it is 0.0002, a 45% performance improvement.

Simulation results are compared with those obtained from the nonlinear controller presented in [2]. Figure 3.6 shows a comparison between the proposed LPV-1 controller and the nonlinear controller [2], with small initial pitch angle, i.e. $\alpha_S = 0.01$ rad. It is clearly can be seen that the proposed LPV-1 control can significantly improve the overall closed-loop performance. As mentioned, the control mass m is confined to move within the groove between -0.5 and 0.5. In the proposed LPV-1 control design, the scheduling parameter θ_1 is used to constrain the mass movement. Figure 3.7 shows the same comparisons but with larger initial pitch angle at $\alpha_L = 0.6$ rad. It is apparent that the nonlinear controller [2] cannot handle the large angle of attack, whereas the proposed LPV-1 control handles this condition effectively, with fast convergence and small control effort.

Figure 3.8 shows a comparison between the LPV-1 control design and the baseline LPV-0 design. Recall that in LPV-0 control design, no position limitation is imposed on the moving mass m. Therefore, as can be seen in Figure 3.8, for LPV-0 controller to be effective in suppressing airfoil vibration, the control mass needs to travel much larger displacement. Furthermore, Figure 3.9 shows a comparison between LPV-1 and LPV-2 controllers when the initial pitch angle is large. Both controllers, as shown, are able to effectively suppress the airfoil vibration, though the LPV-2 controller, with nonlinear springs and dampers imposed at the boundaries, produces less aggressive mass motion. Moreover, as shown in Figure 3.10, the mass motion of LPV-2A control is less

aggressive compared to that of LPV-2 control. Note that Figure 3.6 to Figure 3.10 are generated with a fixed airspeed at $\bar{V} = 2.92$ (or 23.8 m/s).

For flutter analysis, Figure 3.11 shows the closed-loop system responses with LPV-1 control at varying airspeed. As shown, the system is unstable when $\bar{V} = 3.4$. However, Figure 3.12 shows the responses of LPV-2A control with starting airspeed of $\bar{V} = 2.92$, then increasing to $\bar{V} = 3.08$ (or 25.1 m/s) at $\tau = 250$ and to $\bar{V} = 3.4$ (or 27.7 m/s) at $\tau = 450$. As shown, the LPV-2A control can effectively suppress the airfoil vibration beyond the previously identified flutter airspeed of $\bar{V} = 2.96$ (or 24.1 m/s) [2].

A quantitative study of the results, in terms of $\|.\|_\infty$ and $\|.\|_2$ norms of the signals, are presented in Tables 3.2, 3.3, and 3.4. Table 3.2 shows the comparison between the $\|.\|_\infty$ norms of the proposed design techniques and [2] with small and large initial conditions. It is clear that the proposed design techniques render more than 50% improvement for most of the system performance related norms, with nearly 50 time less control effort. A comparison of $\|.\|_\infty$ norms between LPV-1 control and baseline LPV-0 control is presented in Table 3.3, which reveals the benefit of modeling the boundary conditions. Table 3.4 shows a comparison of $\|.\|_2$ norms between the proposed design techniques and the one in [2] with small and large initial conditions. It is clear that the proposed LPV design techniques provide much improved closed-loop responses, with faster convergent rate as indicated in the percentage improvement in the last column.

Finally, to demonstrate the nonlinear nature of the gain-scheduling feedback matrix $K(\theta)$, we plot the first, second and fifth entries of the LPV-2A control gain, and they are shown in Figure 3.13, Figure 3.14 and Figure 3.15. It is clear that $K(\theta)$ is not an affine function of θ. Furthermore, these figures show that the control gain is clearly more affected by the boundary condition θ_1 than the airspeed θ_2.

3.5 Summary

This chapter presented two novel LPV modeling and control design techniques for a smart airfoil model that utilizes a moving mass for flutter suppression. The LPV gain-scheduling state

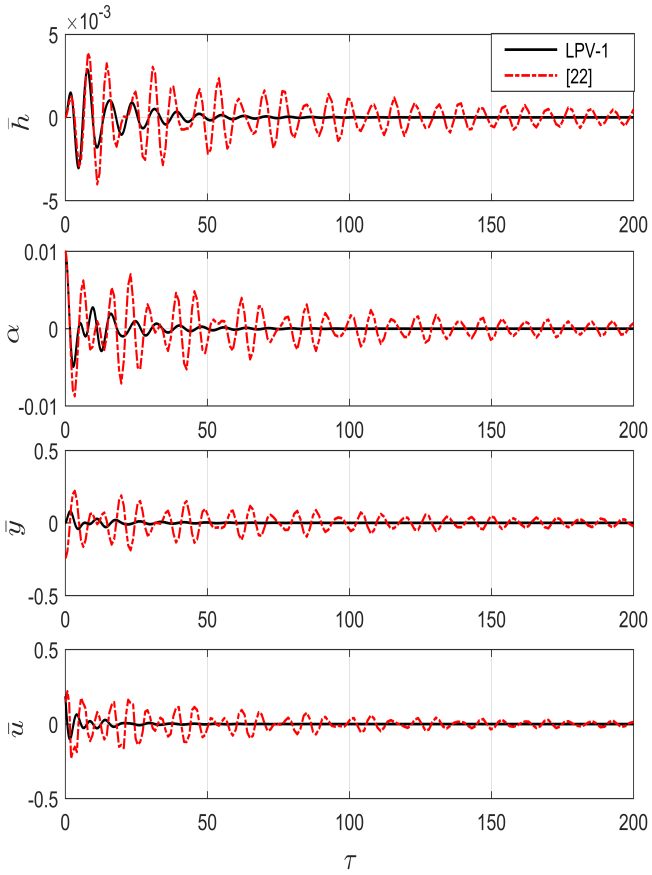


Figure 3.6: Comparison between proposed LPV-1 control and [2]: α_S.

feedback controllers based on the corresponding models were proposed. In the first proposed technique, the moving mass position was used as the scaled scheduling parameter, whereas in the second technique, nonlinear springs and dampers were added to the ends of the groove to constrain the moving mass. The performance of the proposed LPV controllers was compared with an earlier nonlinear controller from the literature, and the results clearly demonstrated the advantages and the effectiveness of the proposed LPV modeling and control techniques in active flutter suppression. In addition, the proposed LPV control design was also able to significantly increase the flutter airspeed.

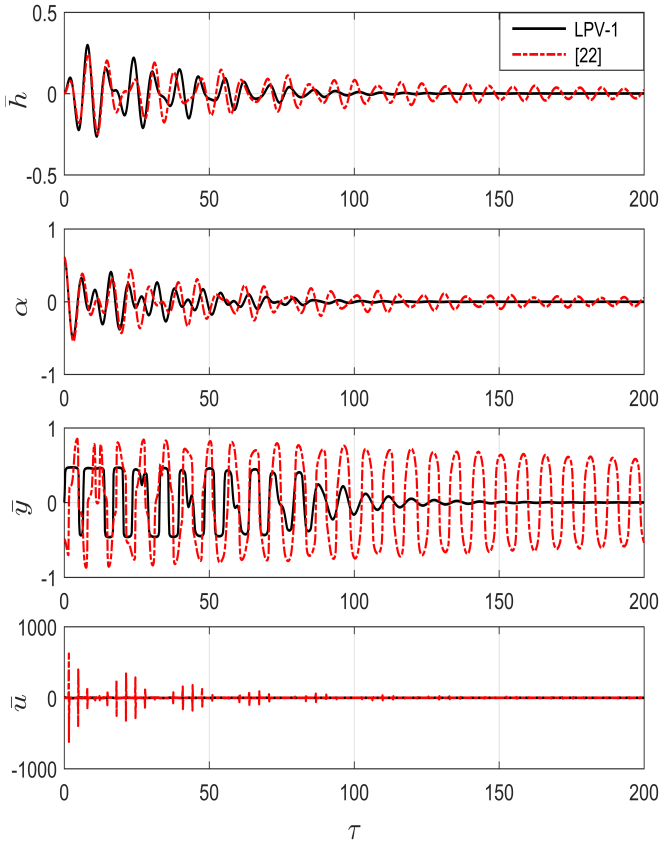


Figure 3.7: Comparison between proposed LPV-1 control and [2]: α_{L}.

Table 3.2: $\|.\| \infty$ comparison; LPV-1 and LPV-2 controllers vs. nonlinear controller [2].

Case	$\|.\|_\infty$	[2]	Proposed LPV
LPV-1 at α_S	$\bar{h}$	0.0040099	0.0029746
	α	0.01	0.01
	$\bar{y}$	0.24058	0.083381
	$\bar{u}$	0.23176	0.18602
LPV-1 at α_L	$\bar{h}$	0.24444	0.29736
	α	0.6	0.6
	$\bar{y}$	0.86839	0.4705
	$\bar{u}$	619.82	11.347
LPV-2 at α_L	$\bar{h}$	0.24444	0.23804
	α	0.6	0.6
	$\bar{y}$	0.86839	0.48297
	$\bar{u}$	619.82	5.5392

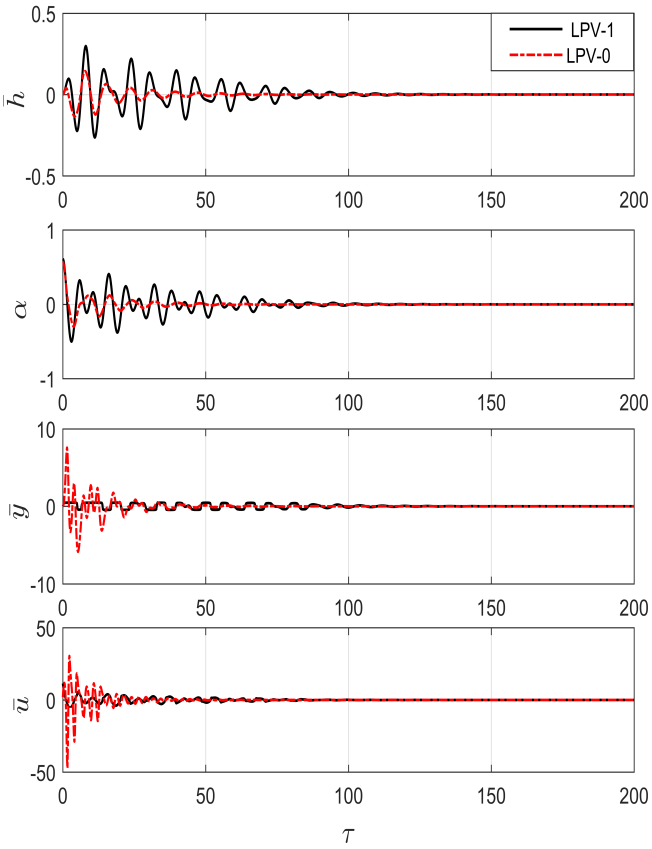


Figure 3.8: Comparison between LPV-1 control and LPV-0 control: α_{L}.

Table 3.3: LPV-1 controller vs. LPV-0 controller: α_{L}

$\|.\|_\infty$	LPV-0	LPV-1
$\bar{h}$	0.14294	0.29736
α	0.6	0.6
$\bar{y}$	7.8091	0.4705
$\bar{u}$	47.979	11.347

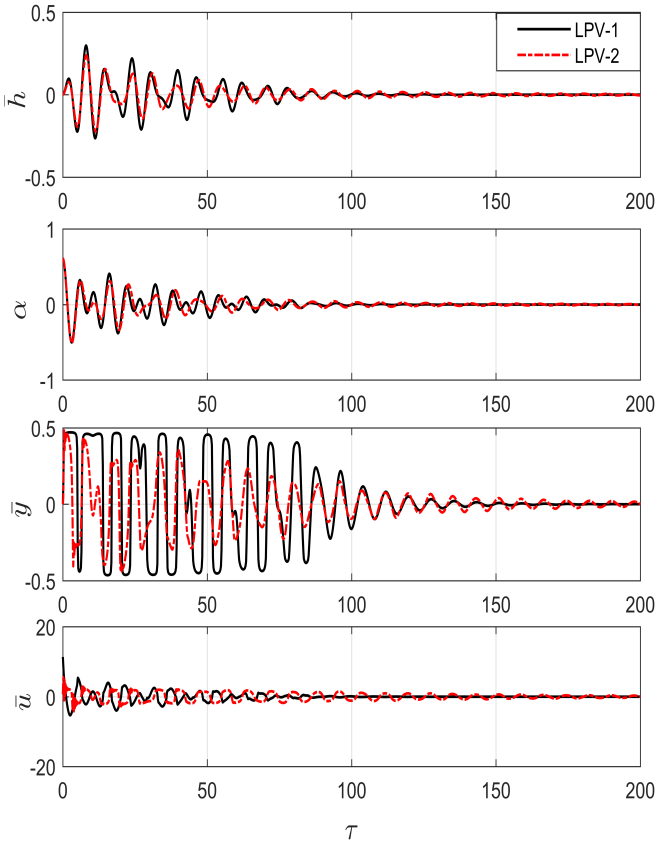


Figure 3.9: Comparison between LPV-1 control and LPV-2 control: α_{L}.

Table 3.4: $\|.\|_2$ comparison; LPV-1 and LPV-2 controllers vs. nonlinear controller [2]

Case	$\|.\|_2$	[2]	Proposed LPV	Convergence rate.%
LPV-1 at α_S	$\bar{h}$	0.4946	0.20804	57
	α	0.98788	0.38593	60
	$\bar{y}$	28.36	4.0848	85
	$\bar{u}$	26.744	11.359	57
LPV-1 at α_L	$\bar{h}$	30.216	27.771	8
	α	60.868	45.592	25
	$\bar{y}$	254.07	115.57	54
	$\bar{u}$	6802.3	519.59	92
LPV-2 at α_L	$\bar{h}$	30.216	22.143	27
	α	60.868	41.641	31
	$\bar{y}$	254.07	65.663	74
	$\bar{u}$	6802.3	502.97	92

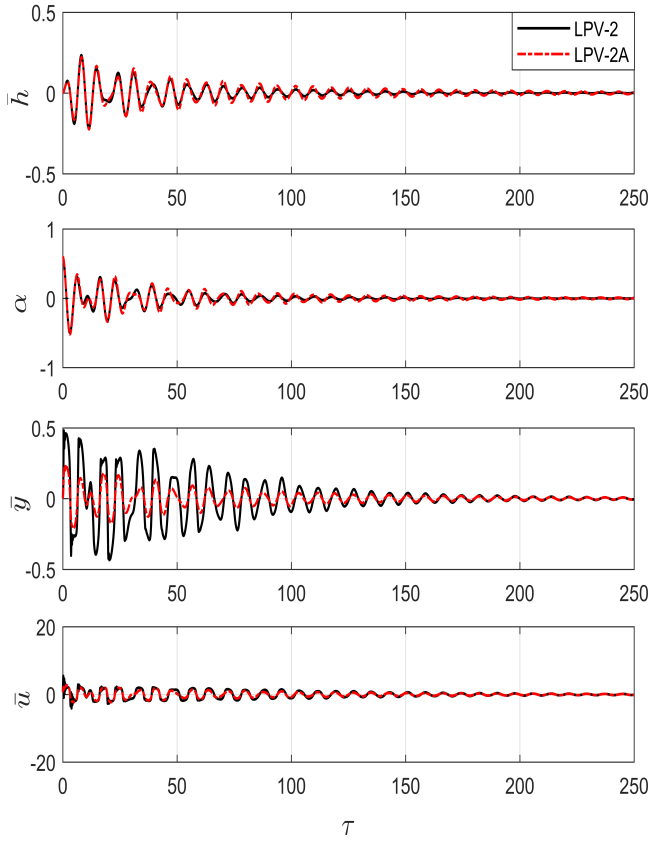


Figure 3.10: Comparison between proposed LPV-2 control and LPV-2A control: α_{L}.

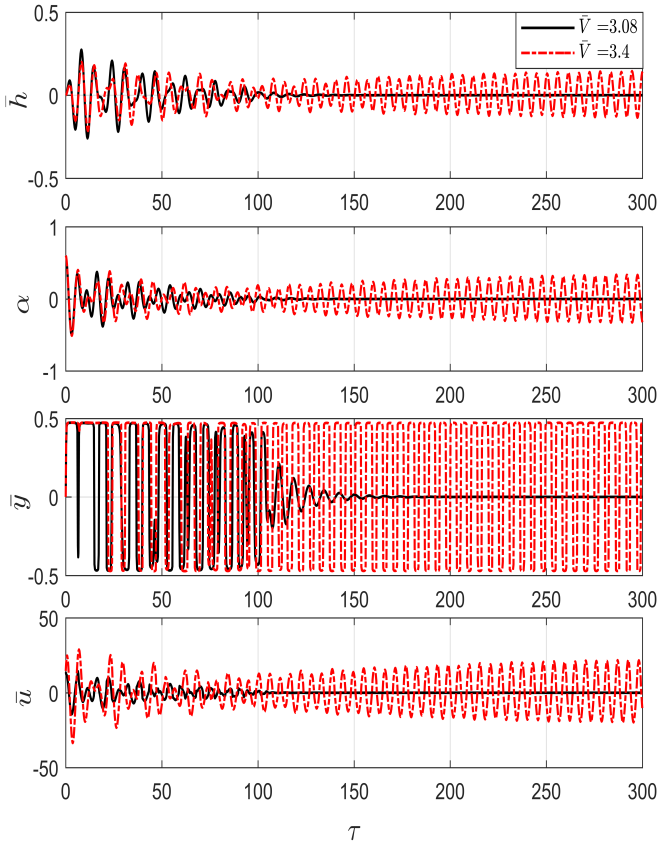


Figure 3.11: LPV-1 control at V=3.08 and V=3.4.

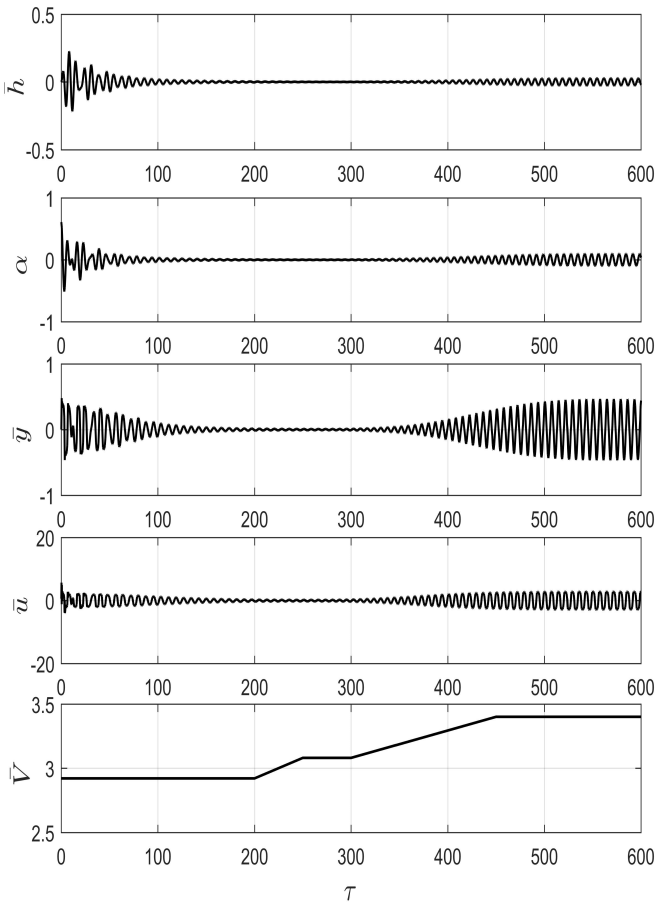


Figure 3.12: LPV-2A control performance at a defined velocity profile, $\bar{V} \in [2.92,\ 3.4]$.

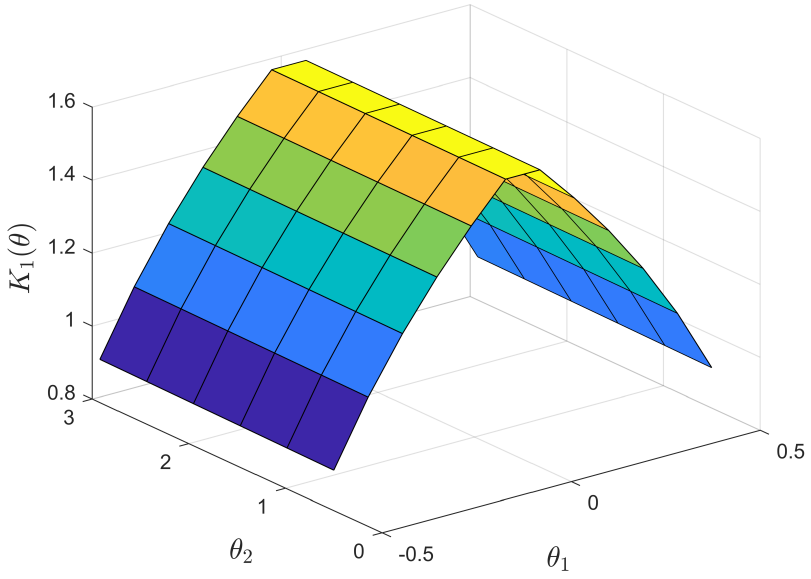


Figure 3.13: First entry of $K(\theta)$.

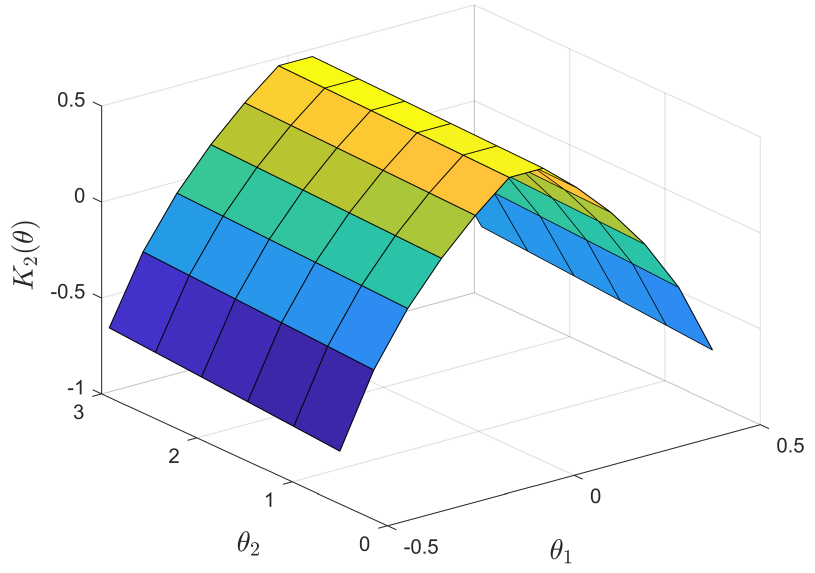


Figure 3.14: Second entry of $K(\theta)$.

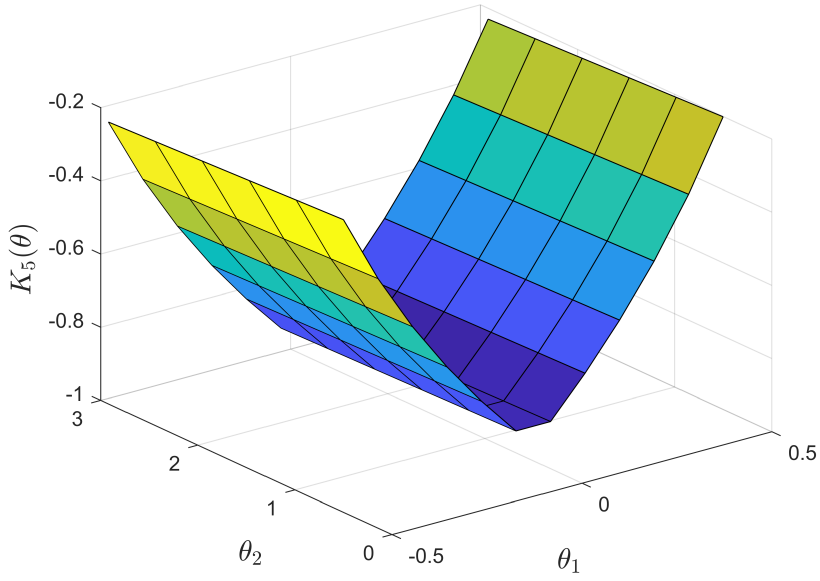


Figure 3.15: Fifth entry of $K(\theta)$.

CHAPTER 4

VIRTUAL CONTROL WITH HARD CONSTRAINTS

4.1 Introduction

Hard constraints in practical control systems are ubiquitous. These constraints constitute physical limitations on some of the state variables, actuation capacity, performance requirements, or any of these combinations. In [56] the Hubble telescope pointing error budget is considered as the constrained performance requirement, [57] and the references therein investigated the missile control problem subject to hard constraints on rudder/aileron deflections, and [58] considered the cutting head control for CNC machines, where the cutting head motion needs to be constrained within the error bound. The input/actuator saturation control problem has been extensively studied and the research results published in this area are abundant; especially, [59] and [60] offer a comprehensive coverage on the subject. The saturable servo systems were considered in [61], modification of the conventional anti-windup was presented in [62], event-triggered scheme was introduced in [63], [64] proposed the theory of maximal output admissible set of control and state saturation, and [65] proposed the concept of pseudo-control hedging (PCH) based on nonlinear dynamic inversion methodology, to just name a few examples. Each of these approaches provides a specific technique with varying complexity in handling the presence of hard constraints in control systems. In particular, application of antiwindup scheme or controller switching technique could potentially cause substantial performance degradation ([4, 5, 6, 7, 66, 67, 68]). In addition, [8, 9, 10] utilized nonlinear model predictive control (MPC) approach to resolve the hard constrained control problems, but with significantly increased control complexity due mainly to the high computational cost. Therefore, for practical control applications, there is still a need to develop an efficient control strategy that is best suited for real-time control implementation for systems with hard constraints.

This work proposes a novel virtual control concept by utilizing the linear MPC strategy in dealing with the control systems with hard constraints. The notion of virtual control known in the

literature is quite different from what is proposed in this work. For example, [69] introduced the virtual constraint that was used to limit the one degree-of-freedom joint motion in order to meet the required switching control rule. In this work, the main idea behind the proposed virtual control is to introduce the virtual mechanism and incorporate it as part of control model development process. This idea is motivated by the fact that one can always devise a physical stoppage mechanism to safe-guard specific states/inputs from ever reaching their saturation limits. In addition, implementation of this physical mechanism will certainly change how system behaves, especially when nearing the saturation. This system behavior is precisely what the virtual control is aimed to achieve, but without actually implementing the stoppage mechanism. Our ultimate control design goal is to better profiling the closed-loop system responses and keep them away from the imposed constraints, especially when they get closer to the constrained boundary. The virtual mechanisms considered in this study are in the forms of virtual nonlinear springs and dampers; of course they can be any other form of constrained mechanisms. The pairs of proposed virtual springs and dampers are devised to insert exponentially growing resistant forces in order to prevent the system from operating near the constrained boundary. The proposed stabilizing controllers are designed for systems with augmented virtual mechanisms, but in simulation analysis and actual implementation, these virtual effects will be taken away. In order to ensure that the virtual control is effective only in the region close to the constrained boundary, the Linear Quadratic Regulator (LQR) based MPC strategy is adopted for discrete real-time application. The virtual control will become inactive when the system is operated far away from the constrained boundary, and will exponentially increase its effect by stiffening both spring and damping effects when nearing the constrained boundary. Based on the MPC scheme, an optimal finite receding horizon LQR controller is designed at each control step and operational point. In addition to the LQR weighting matrices that are tunable to achieve desired performance, the virtual spring and damping parameters can also be used as additional control design variables, and therefore part of the proposed receding horizon LQR strategy can be calculated off-line, making the real-time control viable. Moreover, the proposed approach can easily be incorporated with other advanced control strategies.

The MPC approach is a matured methodology that has progressed rapidly in the past decades. For instance, MPC found its applications in numerous linear systems, [10, 70, 71, 72, 73, 74], and nonlinear and robust systems, [75, 76, 77, 78, 79], to just name a few. In particular, a nonlinear MPC requires very high computational load, especially for problems with hard constraints, which makes the real-time control almost impossible. It should be noted that after the nonlinear springs and dampers are incorporated in the control design model, the conventional linear control approach can no longer be used for control design, hence as a remedy the linear quadratic regulator based MPC method is proposed for solving the virtual control problem. The simplicity of the proposed approach enables the real-time implementation.

This chapter is organized as follows. Section 4.2 presents a ball balancing example with hard constraints. The general problem formulation is presented in Section 4.3 along with the control implementation with the proposed virtual spring and damper profile functions. Section 4.4 presents the simulation results and discussions. Summaries and conclusions are given in Section 4.5.

4.2 Virtual control: a motivational example

Most of the control systems considered in practical applications are usually subject to a given level of constraints on some of their states and/or inputs. In this work, these constraints are explicitly addressed by introducing spring and damper profile functions that are used to not only safe-guard the constraints from being violated, but also prescribe the system dynamics when nearing the constraints. The objective is to develop the capability to better tailor the overall system behaviors as desired, while meeting the practical limitations on the states and/or inputs. In addition, from the control design perspective, these added spring and damper profile functions can be considered as "virtual" control and incorporated as part of the state feedback design.

As a motivational example, we consider a balancer problem, [80, 81], as illustrated in Figure 4.1. We assume that the ball is rolling without slipping and its motion is controlled by an input torque subject to a constraint on rotational angle. The linearized equations of motion can be described by

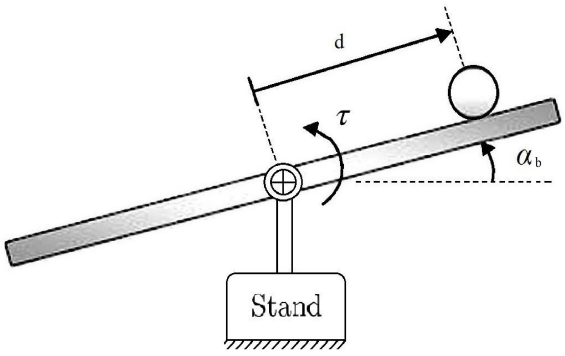


Figure 4.1: A ball on a balancing beam.

$$\begin{cases} m\ddot{d} = mg\alpha_b \\ I\ddot{\alpha_b} = -mgd + \tau \end{cases} \tag{4.1}$$

where m denotes the mass of moving ball, I the moment of inertia of the beam (assuming that the ball mass is relatively small), and τ the control torque. In addition, the rotational angle α_b is constrained by

$$|\alpha_b| \leq \alpha_h\,, \tag{4.2}$$

where α_h is a prescribed bound. The essence of this work is to highlight the fact that the constraint (4.2) can be effectively enforced by introducing a pair of "virtual" stoppage mechanisms of varying stiffness and damping properties, as shown in Figure 4.2.

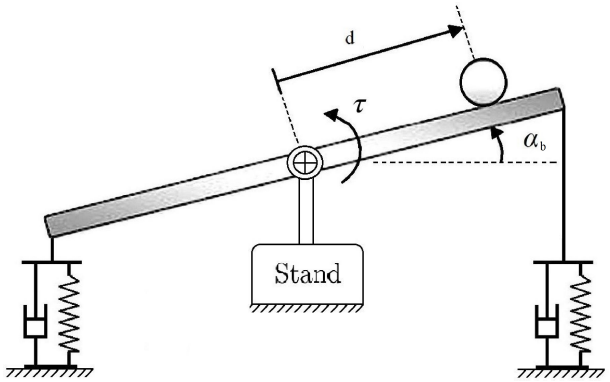


Figure 4.2: A ball on a balancing beam with virtual springs and dampers.

As mentioned earlier, these augmented virtual mechanisms are aimed to tailor the system

performance when operating near saturation limit, therefore safe-guarding it from reaching the saturation. And the resultant system behavior is precisely what virtual control system is aspired to achieve, but with absence of augmented mechanisms.

4.3 Virtual modeling and control framework

Consider a linear time invariant system subjected to an augmented function as follows,

$$\dot{x} = Ax + Bu + Df(x), \tag{4.3}$$

where (A, B) is controllable, B has full column rank, D is a known matrix, and $f(x)$ contains a vector of continuous profile functions that are to be augmented. It should be noted that the entries in D are identified apriorily to reflect the constraints on certain states.

Assumption 1: The image of matrix D belongs to the image of matrix B, that is, there exists a matrix M such that

$$D = BM\,.$$

The above matching condition reveals that the constraints are only imposed upon the system in the same way as its inputs. Therefore, without loss of generality, the system described in (4.3) can be rewritten as

$$\dot{x} = Ax + Bu + BMf(x) \tag{4.4}$$

In the case of balancing ball example, if we let $x = [d\ \dot{d}\ \alpha_b\ \dot{\alpha_b}]^T$ and $u = \tau$, then the state space representation of (4.1) can be written in the form of (4.4), where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ -mg/I & 0 & 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/I \end{bmatrix}. \tag{4.5}$$

In addition, $M = 1$ and the augmented profile function $f(x)$ can be chosen to be

$$f(x) = \begin{bmatrix} 0 & 0 & -k(\alpha_b) & -c(\alpha_b) \end{bmatrix} x = Gx, \tag{4.6}$$

where $k(\alpha_b)$ and $c(\alpha_b)$ are adjustable stiffness and damping parameters as function of α_b. In particular, these parameters are to be increased, hence spring/damper stiffened, as $|\alpha_b|$ gets closer to its hard limit α_h, so as to prevent it from reaching the bound. The mathematical description of various proposed profile functions for $k(\alpha_b)$ and $c(\alpha_b)$ will be presented later.

Since the profile function $f(x)$ is known a priori as in (4.6), let $A_v = BMG$ then the system (4.4) can be rewritten as follows,

$$\Sigma : \ \dot{x} = (A + A_v)x + Bu = Ax + B\hat{u}, \tag{4.7}$$

where $\hat{u} = u + Mf(x)$ is denoted as a "new" control input and $Mf(x)$ the virtual control. In what follows, the system Σ is used for designing the receding horizon LQR-based model predictive control law (u), where $Mf(x)$ is part of the LQR-MPC design model, and the presence of augmented profile functions are to ensure the constraints on states are met.

4.3.1 Receding horizon LQR control

Ideally, a continuous LQR-MPC controller should be designed, but due to high computational load of solving the differential Riccati equation, a finite receding horizon LQR-based discrete-time MPC approach is proposed for Σ, where the nonlinear model (due to inclusion of $Mf(x)$) is discretized with a given sample period T for a given $f(x(k_0T))$. The discretized state, control, and the system matrices are denoted by $x(k_0)$, $u(k_0)$, and $A_d(k_0)$ and $B_d(k_0)$, respectively, at sample time k_0T. The performance index at sample instant k_0 with k_f-step control policy is described by

$$J = \frac{1}{2}x(k_f + k_0)^T F x(k_f + k_0) + \frac{1}{2}\sum_{k=k_0}^{k_f+k_0-1} x(k)^T Q x(k) + u(k)^T R u(k), \tag{4.8}$$

where R is a positive definite symmetric matrix, and F and Q are positive semi-definite symmetric matrices. The goal is to design a receding horizon controller u of the form

$$u(k) = L(k)x(k), \;\; k = k_0, k_0 + 1, ..., k_0 + k_f - 1, \tag{4.9}$$

that minimizes J subject to discretized Σ. Following the standard MPC derivation; see for instance [16] and [82], the control gain matrix $L(k)$ is given by

$$L(k) = -R^{-1}B_d^T(k_0)A_d(k_0)^{-T}(P(k) - Q), \qquad k = k_0, k_0 + 1, ..., k_0 + k_f - 1,$$

where $P(k)$ solves the following discrete-time Riccati equation

$$\begin{cases} P(k) = A_d(k_0)^T P(k+1)[I + B_d(k_0)R^{-1}B_d^T(k_0)P(k+1)]^{-1}A_d(k_0) + Q, \\ P(k_0 + k_f) = F, \\ k = k_0, k_0 + 1, ..., k_0 + k_f - 1 \end{cases} \tag{4.10}$$

Recall the defined control $\hat{u}$ for Σ is given by

$$\hat{u}(t) = u(k_0) + Mf(x), \;\; k_0 T \leq t < (k_0 + 1)T. \tag{4.11}$$

where $u(k_0)$ is given in (4.9).

4.3.2 Virtual constraint profiles

As discussed earlier, the augmented virtual mechanisms are aimed to constrain the system responses within the imposed limitations. In the case of ball balancing problem, these virtual mechanisms can be devised in terms of parameter dependent springs and dampers. In this study, we propose three types of profile functions. The first type is a pair of constant spring and damper, i.e. $k(\alpha_b) = k$ and $c(\alpha_b) = c$. The second type is a piece-wise linear symmetric function for $k(\alpha_b)$ and $c(\alpha_b)$; a sample of such function is shown in blue solid-line in Figure 4.3, and mathematically it can be described by

$$k(\alpha_b) = \begin{cases} \frac{(|\alpha_b| - \alpha_m)}{\alpha_h - \alpha_m} k_2 + k_1, & \text{if} \quad \alpha_m \leq |\alpha_b| \leq \alpha_h \\ \frac{(|\alpha_b| - \alpha_s)}{\alpha_m - \alpha_s} k_1, & \text{if} \quad \alpha_s < |\alpha_b| < \alpha_m \\ 0, & \text{if} \quad |\alpha_b| \leq \alpha_s \end{cases} \tag{4.12}$$

$$c(\alpha_b) = \begin{cases} \frac{(|\alpha_b| - \alpha_m)}{\alpha_h - \alpha_m} c_2 + c_1, & \text{if} \quad \alpha_m \leq |\alpha_b| \leq \alpha_h \\ \frac{(|\alpha_b| - \alpha_s)}{\alpha_m - \alpha_s} c_1, & \text{if} \quad \alpha_s < |\alpha_b| < \alpha_m \\ 0, & \text{if} \quad |\alpha_b| \leq \alpha_s \end{cases} \tag{4.13}$$

where the coefficients (k_1, k_2, c_1, c_2) are tunable parameters according to the desired performance, α_s denotes where the spring/damper begins to affect, and α_m denotes the mid-point. It is important to note that both α_s and α_m can be chosen according to the desired performance. The third type of profile function is a scaled continuous tangent function; shown in black dashed-line in Figure 4.3, and a sample description for a pair of nonlinear spring and damper can be described by

$$k(\alpha_b) = \begin{cases} \gamma \tan\left(\frac{\pi(|\alpha_b| - \alpha_s)}{2(\alpha_h - \alpha_s)}\right), & \text{if} \quad \alpha_s < |\alpha_b| \leq \alpha_h \\ 0, & \text{if} \quad |\alpha_b| \leq \alpha_s \end{cases} \tag{4.14}$$

$$c(\alpha_b) = \begin{cases} \beta \tan\left(\frac{\pi(|\alpha_b| - \alpha_s)}{2(\alpha_h - \alpha_s)}\right), & \text{if} \quad \alpha_s < |\alpha_b| \leq \alpha_h \\ 0, & \text{if} \quad |\alpha_b| \leq \alpha_s \end{cases} \tag{4.15}$$

where γ and β are the scaling factors used to shape the spring and damping coefficients as desired. As can be seen, the effect of virtual mechanisms become active when α_b is outside of $[-\alpha_s \ +\alpha_s]$, and progressively become stiffened as α_b gets closer to the hard bound $\pm\alpha_h$. When $\alpha_b \in [-\alpha_s \ +\alpha_s]$ the virtual springs and dampers are inactive, and the system Σ behaves as a baseline linear time invariant system. The choice of profile function will depend on the system description and its constraints, as well as the available control authority. This will be further explored next.

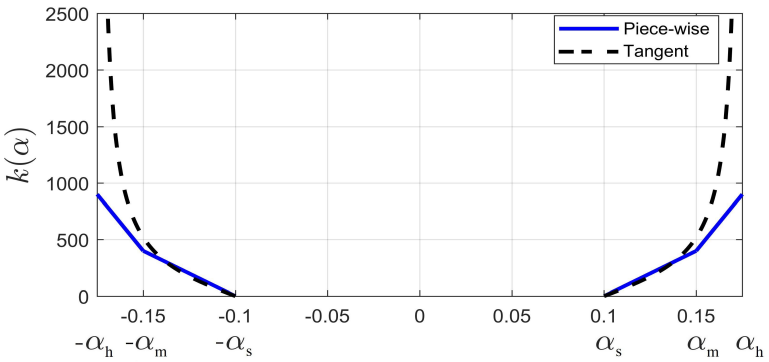


Figure 4.3: Virtual spring profile functions.

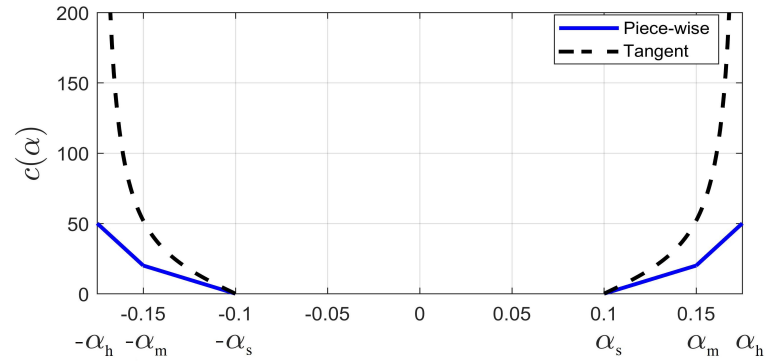


Figure 4.4: Virtual damper profile functions.

4.4 Concept validation through simulation study

In this section, we first demonstrate the effectiveness of the proposed virtual modeling and control approach by utilizing the ball balancing problem in Section 4.2. The receding horizon LQR based MPC controllers are designed with and without the augmented virtual springs and dampers. Table 4.1 shows the parameters used in the simulations.

Furthermore, the horizon step k_f and sample period T are set at 40 and $0.1s$, respectively, and the weighting matrices in the cost function are chosen to be

$$F = \text{diag}\,[512,\ 5,\ 4,\ 0.02],\ Q = \text{diag}\,[120,\ 1,\ 80,\ 1],\ R = 1\,.$$

Table 4.1: Ball balancing parameters.

Item	Parameter Name	Values	Units
I	Moment of inertia	0.0533	kg.m^2
L	Length of beam	0.8	m
m	Ball mass	0.05	kg
T	Sampling rate	0.1	s
α_s	$\|\alpha_b\|$ lower limit	0.1	rad
α_m	$\|\alpha_b\|$ mid-point for piece-wise function	0.15	rad
α_h	$\|\alpha_b\|$ upper limit	0.175	rad

4.4.1 Virtual control with constant profile

For preliminary analysis, a pair of simple constant spring k and damper c are considered as virtual constrained mechanism, and they are chosen to be 400 N/rad and 20 N.s/rad, respectively. A hard constraint on the beam rotational angle is set at $\alpha_h = 0.175$ rad ($\approx 10^\circ$). Four initial ball locations are considered, i.e. $x_0 = [0.2\ 0.25\ 0.345\ 0.4]$ m. Figure 4.5 shows the closed-loop responses when $x_0 = 0.2$ m for the two cases, where the dotted-lines are responses from the LQR controller designed without virtual spring and dampers (or A_v) and the solid-lines are responses from LQR-based MPC with implementation of virtual constant springs and dampers. As shown, even without the augmented springs and dampers, the rotational angle α_b is within the bound ± 0.175 rad, though with augmented virtual mechanisms the responses are more appreciative, especially the control effort is much less with the proposed virtual control approach. When the ball starts at a little higher location, $x_0 = 0.25$ m, it can be seen in Figure 4.6 that, for no augmented virtual spring/damper case (red dashed-line), the rotational angle needs to travel beyond the imposed constraint α_h and it requires much larger control effort in order to stabilize the motion of the ball, whereas with virtual mechanisms (blue solid-line), the constraint is met with less control effort. However, when the ball starts at even higher location, x_0= 0.345 m, the proposed constant virtual spring/damper fails to constrain α_b within ± 0.175 rad, as shown in Figure 4.7 (red dashed-line). Therefore, in what follows additional virtual control mechanisms with tunable profile parameters are considered and their performance assessed.

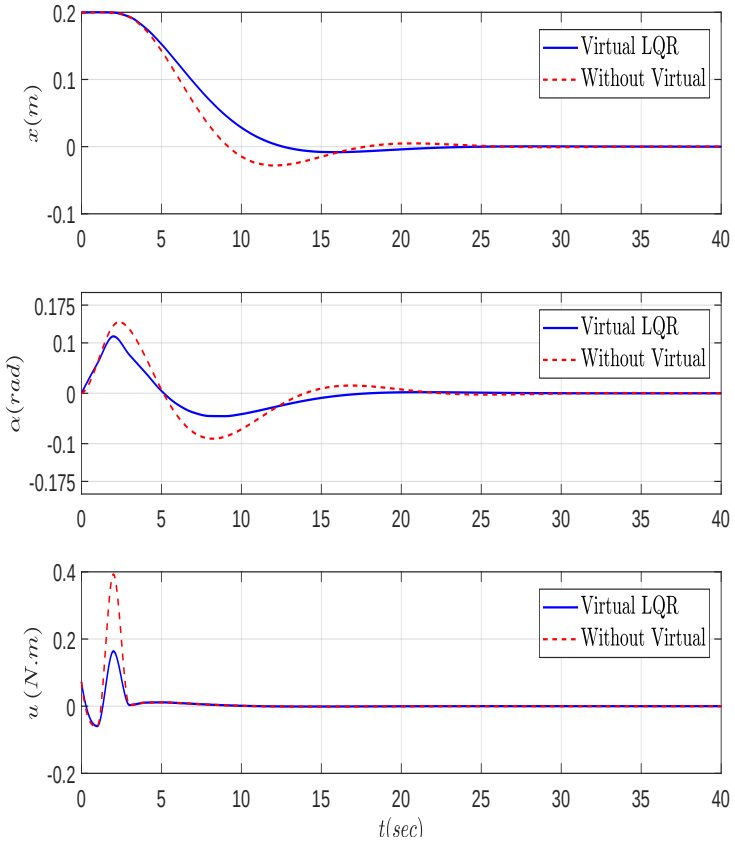


Figure 4.5: Comparison of with virtual control and without virtual control (x_0= 0.2m).

4.4.2 Virtual control with piece-wise linear profile

The piece-wise linear profile functions of the form presented in (4.12) and (4.13) are considered. Five sets of coefficients for (k_1, k_2, c_1, c_2) are chosen to study the effect of various linear shape functions on closed-loop responses. Table 4.2 provides the chosen coefficients, where LVC1, LVC2, LVC3, LVC4, and LVC5 denote the corresponding virtual receding horizon LQR controllers, respectively. It should be noted that in developing the virtual controllers, LVC1 to LVC5, the same weighting matrices and horizon step as defined in previous are used. The initial ball location is set at x_0= 0.345 m.

Figure 4.7 (blue solid-line) shows that the proposed piece-wise linear profile function LVC1 is able to successfully constrain the closed-loop response on α_b. In particular, the mid-point of LVC1 corresponds to the constant spring/damper pair used to generate Figure 4.7 (red dashed-line) that fails to satisfy the constraint. It should be noted that both LVC1 and constant spring/damper profile

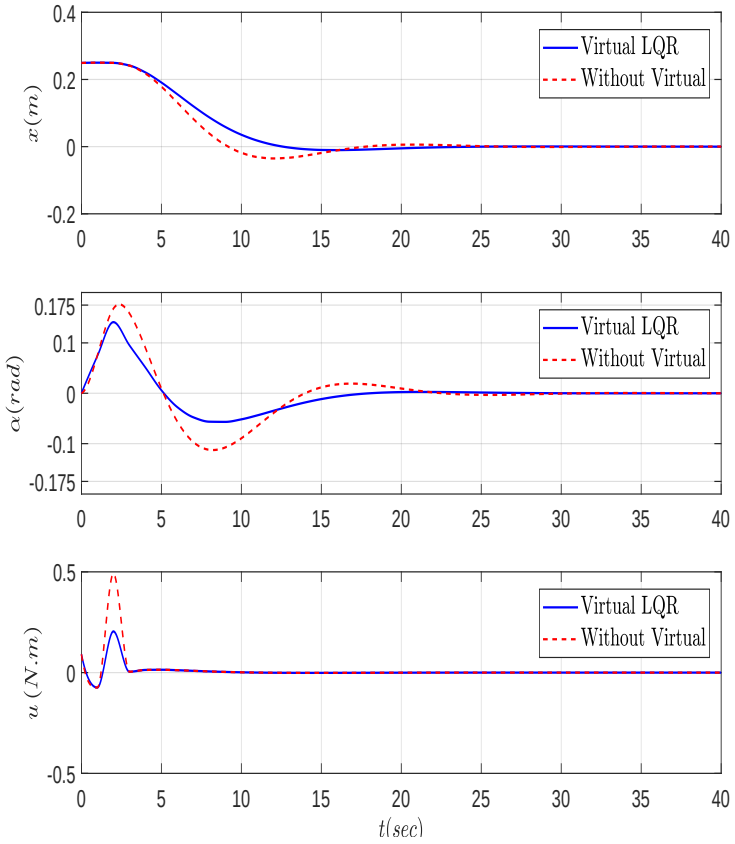


Figure 4.6: Comparison of with virtual control and without virtual control (x_0= 0.25m).

Table 4.2: Virtual spring and damping constants for piece-wise linear function

Parameter	LVC1	LVC2	LVC3	LVC4	LVC5
k_1 (N/rad)	400	800	1200	400	400
k_2 (N/rad)	500	900	1300	500	500
c_1 (N.s/rad)	20	20	20	30	40
c_2 (N.s/rad)	30	30	30	40	50

yield very agreeable control effort, this is critical for making an objective assessment about the two cases. Figures 4.8 and 4.9 show the closed-loop responses with augmented virtual piece-wise springs and dampers, and the constraint on α_b is met. Figure 4.8 shows that increasing spring constant reduces the maximum rotational angle α_b, but with increased control effort, whereas Figure 4.9 indicates that increasing damping constant is able to reduce maximum rotational angle with less control effort. However, when the ball starts at much higher location than x_0= 0.345 m, Figure 4.10 indicates that the proposed virtual piece-wise linear controllers are unable to constrain

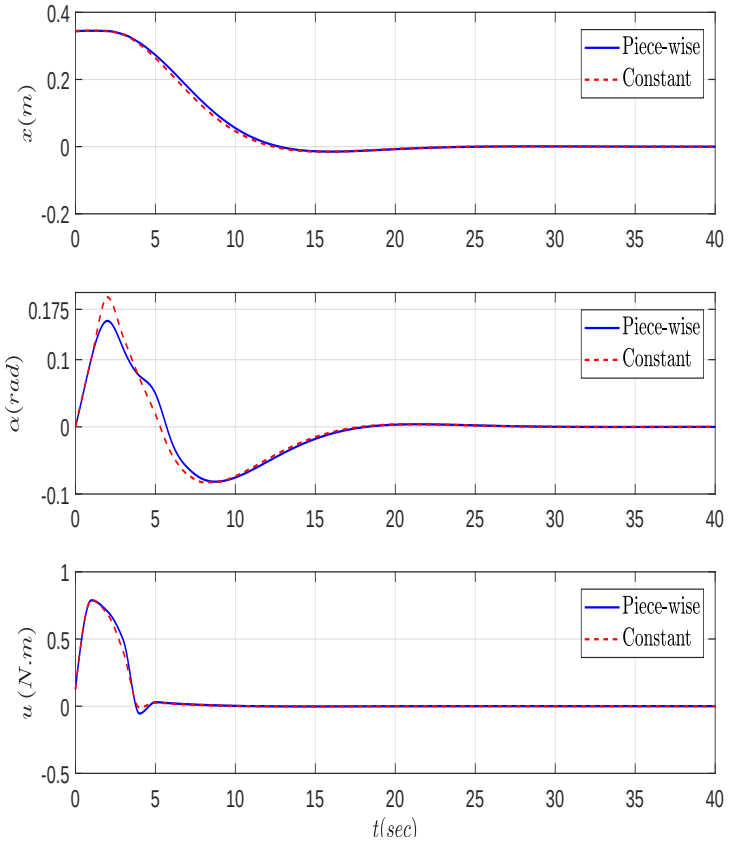


Figure 4.7: Comparison of constant spring/damper vs. piece-wise profile LVC1 (x_0= 0.345m).

α_b within ±0.175 rad (red dashed-line). Therefore, a much aggressive profile function is needed, especially when $|\alpha_b|$ gets closer to α_h.

4.4.3 Virtual control with tangent profile function

Here, the continuous tangent profile functions presented in (4.14) and (4.15) are utilized to better confine the balancing beam. In this simulation, five sets of scaling factors for (β, γ) are chosen to study their effect on closed-loop responses. These five sets of parameters are listed in Table 4.3, where TANVC1, TANVC2, TANVC3, TANVC4, and TANVC5 denote the associated virtual receding horizon LQR controllers, respectively. As shown in Figure 4.3 (black dashed-line), the effects of spring/damper become very aggressive as $|\alpha_b|$ approaches α_h, this would prevent the rotation of the beam from ever reaching ±0.175 rad. Therefore, if there are absolute hard constraints imposed on the system description, incorporating the proposed virtual tangent profile

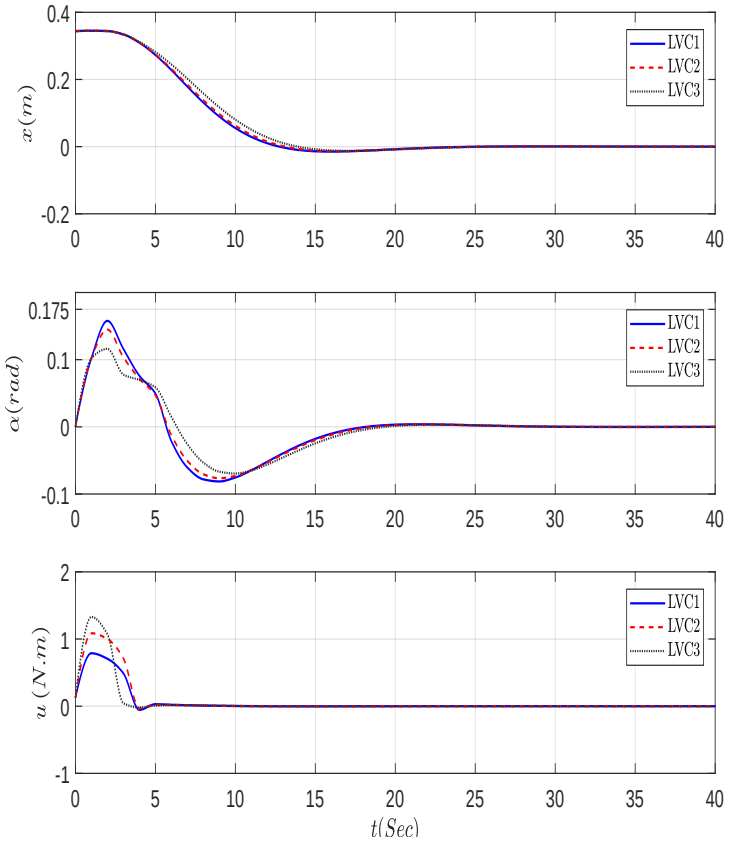


Figure 4.8: Effect of changing virtual spring constant for piece-wise profile (x_0= 0.345m).

function would be a viable approach. Again, the same weighting matrices and horizon step are used for constructing the virtual controllers. The initial ball location is set at x_0 = 0.4 m.

Table 4.3: Virtual nonlinear spring and damping constants for tangent function

Parameter	TANVC1	TANVC2	TANVC3	TANVC4	TANVC5
γ (N/rad)	300	600	900	300	300
β (N.s/rad)	30	30	30	35	40

Figure 4.10 (blue solid-line) shows that the tangent profile function TANVC1 is able to successfully constrain the closed-loop response on α_b, while the piece-wise profile function LVC1 failed. In addition, as shown in Figure 4.10 that TANVC1 and LVC1 yield similar control effort. The closed-loop responses with various tangent parameters presented in Table 4.3 are shown in Figures 4.11 and 4.12, and as shown, all responses meet the α_b constraint. The similar trends as shown in Section 4.4.2 are observed in Figures 4.11 and 4.12, in addition it can be clearly

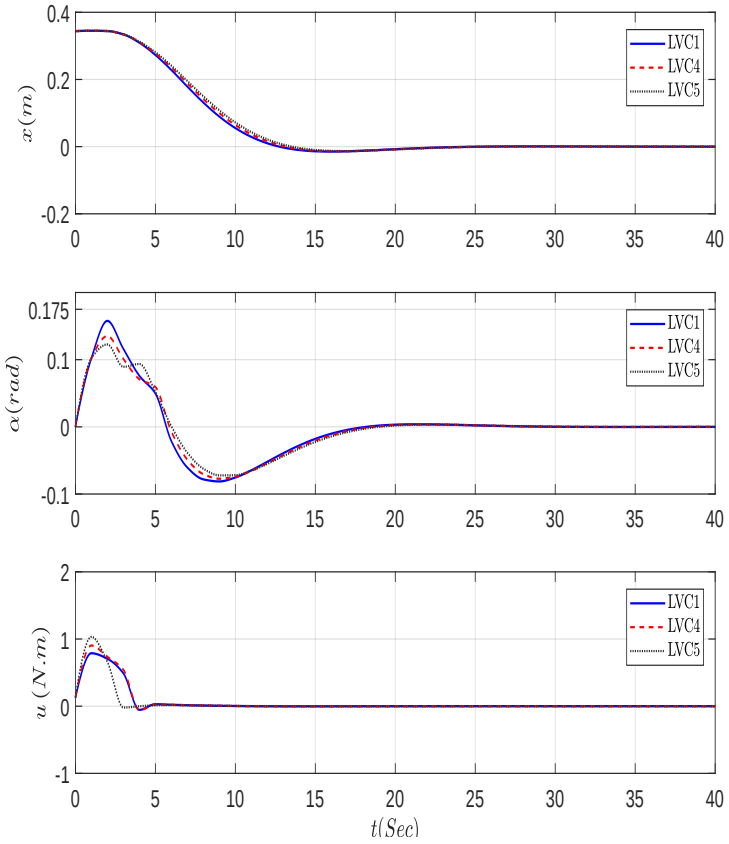


Figure 4.9: Effect of changing virtual damping constant for piece-wise profile (x_0= 0.345m).

demonstrated that, as in Figure 4.3 (black dashed-line), as $|\alpha_b|$ approaches α_h the spring/damper constant increases exponentially, and this aggressive encounter restricts the motion of the beam.

4.5 Summary

In this chapter, a novel virtual control method in dealing with hard constraints is proposed and a ball balancing problem is used to demonstrate the efficacy of the proposed concept. A pair of virtual spring and damper of various profile functions are introduced to accommodate apriorily identified state constraints and incorporate them into a virtual control model. Subsequently, the optimal receding horizon LQR-based MPC approach is utilized to design the stabilizing controllers for both with and without virtual mechanisms. The simulation results clearly demonstrate the benefits of the proposed virtual control concept, and depending on the nature of constraints, the virtual mechanisms can be tuned to achieve desired closed-loop system performance, while meeting

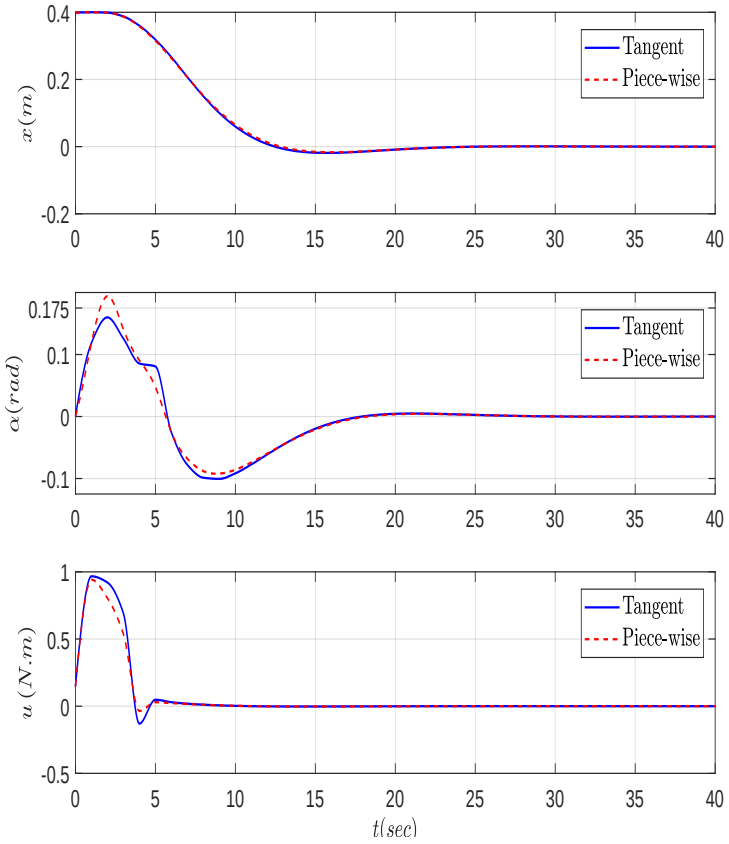


Figure 4.10: Comparison of piece-wise profile LVC1 and tangent profile TANVC1 (x_0 = 0.4m).

the constraint requirements. In particular, as shown in the simulations that as α_b is coming closer to its hard limits α_h the virtual spring/damper can be made progressively effective so as to prevent α_b from ever reaching the limit. This concept is especially relevant when implementing a physical hardware system, where one of the primary control design objectives is to safe-guard the hardware constraints during its operations.

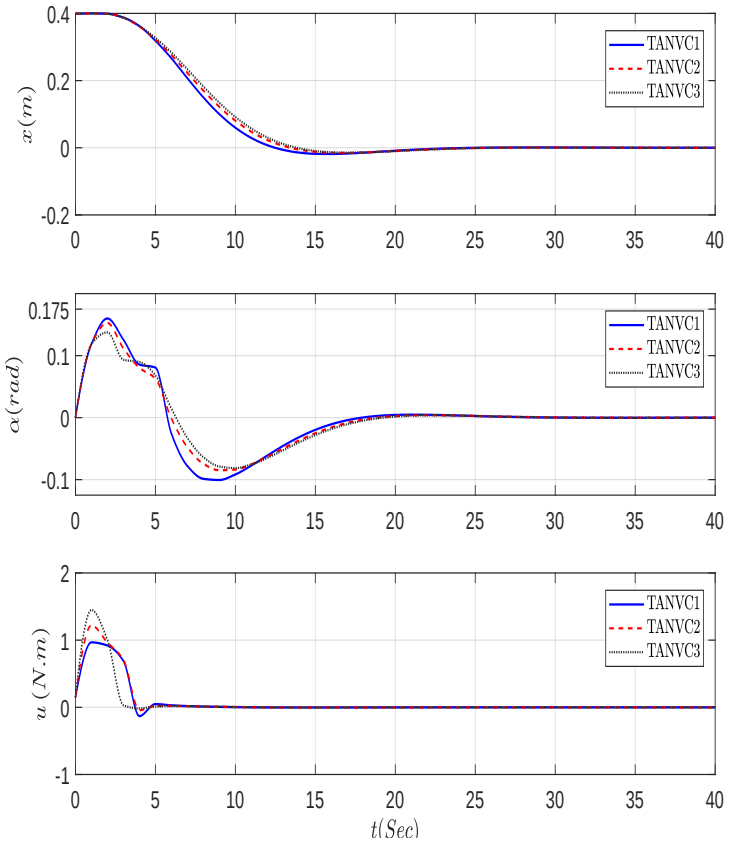


Figure 4.11: Effect of changing γ for tangent profile (x_0= 0.4m).

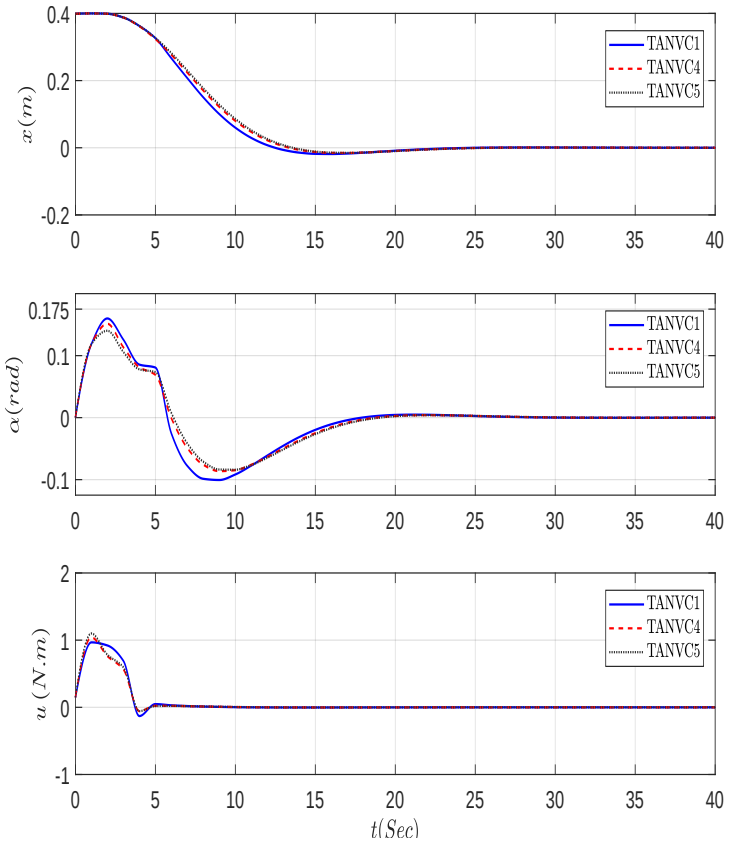


Figure 4.12: Effect of changing β for tangent profile (x_0= 0.4m).

CHAPTER 5

VIRTUAL LPV FLUTTER CONTROL OF A SMART AIRFOIL WITH HARD CONSTRAINTS

5.1 Introduction

As mentioned in previous chapters that the problems of hard constraints are very important topic in the industrial applications and control theory since it related to the actuator saturation problem and especially many real life applications are modeled and behave like an LPV systems. Examples are varied from Rocket angel of attack to be kept in specific range Rodriguez and Cloutier [57] and many other applications are considered under this topic.

The control problems with hard constraints exist in almost every physical systems. For instance, the motion of a linear actuator is always constrained by its mechanical limits. The control problems with hard constraints were a hot research topic for the last three decades in the control community. Many control approaches were used for the control problem ranging from antiwindup [83, 84, 85], conventional LPV [86, 87, 88, 89], switching LPV [90, 91, 92] to mention a few but not to limit. Most of investigations focus on a specific hard constrained control problem and some are computationally expensive. Since control design with hard constraints is complicated, there is a need for a simple and computationally sound control design method for a wide range of systems with hard constraints. This chapter proposes a new control synthesis approach by introducing the concept of virtual control for system with hard constraints. The idea is to introduce virtual mechanisms near hard constraints and to prevent the system from operating close to the hard constrained regions during the control design and incorporate the virtual mechanisms as part of the control dynamics during the control implementation.

From existing literature, it can be found that the virtual control notion used is quite different from what is used in this chapter. They can be divided in two main categories. The first one is

centered on Fault Tolerant Control (FTC) (see, for example, [93, 94, 95, 96]), where reconfigurable control is introduced for faulty systems and named as the virtual actuator. In [93], a fault tolerance control scheme, using model reference control as virtual actuator, is proposed for a linear parameter-varying (LPV) system subject to faulty actuators. More similar work can be found in [97, 98]. The second category is related to virtual simulation or controlling a virtual object; see [99] for virtual controlling a numerical machine, [100] for virtual testbed of industrial security systems, [101] for virtual controlling two layer hydraulic flow, [102] for virtual backstepping neural network control, and [103] virtual decomposition control (VDC) of large robotics by virtually decomposing and analyzing coupling force. As a summary, both virtual control categories are different from what is proposed in this chapter.

In this chapter, a novel LPV hard constrained virtual control (LPVVC) scheme is proposed for the Linear Parameter Varying (LPV) systems, where the virtual control are casted by introducing virtual mechanisms (such as springs and dampers) near the hard constraints to keep the system states from moving close to the posted hard constraints. The usage of term "virtual" is due to the fact that the introduced virtual mechanisms are not physical and they become part of controller dynamics when the designed LPV control is implemented physically. A state-feedback LPV gain scheduling controller is designed with guaranteed $\mathcal{H}_\infty$ performance and combined with the virtual mechanism dynamics. This control scheme is applied to suppressing the flutter of a smart airfoil. The target smart airfoil is an NACA0012 airfoil with a groove in the middle of its chored and a moving mass along the groove used to suppress the pitching motion caused by flutter phenomena; see [1, 2, 104] for details. The motion of the actuating moving mass is limited to the groove length. Virtual mechanisms in term of nonlinear springs and dampers are placed to both ends of the groove to prevent the mass from moving to both ends of the groove (hard constraints). Note that this approach is flexible and can be applied to a wide range of LPV systems with various hard constraints. In addition, the virtual mechanisms can be added to any number of actuation channels with any nonlinear dynamics; see details in the next two sections. To the best of authors' knowledge, the LPV virtual

control with hard constraints has never been reported before in the literature. Since most of physical systems are nonlinear with hard constraints and can be modeled as LPV systems, the proposed LPV virtual control with hard constraints bridges LPV control and practical hard constrained control problems. In addition, due to the flexibility of selecting the virtual mechanisms, the proposed method can be used for many kinds of control problems with hard constraints, and furthermore, it is fairly inexpensive in terms of computational load, comparing with nonlinear MPC approach [105].

The main contribution of this chapter are four-fold. Firstly, the LPV virtual control (LPVVC) is formulated for LPV systems with hard constraints. During the LPV control design process, virtual mechanisms are used to prevent certain states from moving outside of their hard constraints and during the control implementation phase, the designed LPV control strategy is combined with the virtual mechanical dynamics to form the LPVVC control law. Note that the proposed control scheme is radically different from other methods used to handle the hard constrained control problems such as anti-windup, nonlinear, MPC, etc. Secondly, the proposed method is applied to a smart airfoil flutter control problem with hard constraints, where the smart airfoil is modeled as an LPV system to accommodate the nonlinearities in air speed [1] and introduced virtual mechanisms. Thirdly, introducing virtual variable gains profile function. Where it allows the control designer to choose different virtual variable gain profile according to the application demands and introduce them in the controller to prevent actuator saturations. Finally, the performance of the designed LPVVC control is compared with that of nonlinear control (NLC) [2], conventional LPV control (named LPVN) with physical springs and dampers [1], LPV control (named LPVR) without considering hard constraints.

The reminder of the chapter is organised as follows. Section 5.2 presents the virtual control general problem formulation. The application of the proposed method to the smart airfoil flutter suppression example is introduced in Section 5.3 along with the control implementation. Section 5.4 presents the simulation results and discussion of comparisons with literatures and other control

methods. Finally, the chapter closes in Section 5.5 with conclusions and future work.

5.2 LPV Virtual Control Problem Formulation

In this section, we consider the affine LPV system described by

$$\begin{cases} \dot{x}(t) = A(\theta(t))x(t) + B_u(\theta(t))u(t) + B_w(\theta(t))w(t) \\ y(t) = C(\theta(t))x(t) + D_u(\theta(t))u(t) + D_w(\theta(t))w(t) \end{cases} \tag{5.1}$$

where $\theta(t) = (\theta_1(t), \theta_2(t), \ldots, \theta_q(t))$ denotes a vector of scheduling parameters, $x(t)$ the state, $u(t)$ the control input, $w(t)$ the disturbance input, and $y(t)$ the controlled output. All system matrices are assumed to be compatible in dimensions and they are in affine parameter-dependent form. For example, $A(\theta)$ is represented by

$$A(\theta(t)) = A_0 + \sum_{i=1}^{q} \theta(t)_i A_i \,, \tag{5.2}$$

where A_0 and A_i, $i = 1, 2, \ldots, q$, are constant matrices, and the scheduling parameters $\theta_i(t)$ are assumed to be measurable in real-time and their magnitude and variational rate are bounded as follows,

$$\underline{\nu}_i \leq \theta_i(t) \leq \bar{\nu}_i, \; i = 1, 2, \ldots, q, \tag{5.3}$$

$$-\sigma_i \leq \dot{\theta}_i(t) \leq \sigma_i, \; i = 1, 2, \ldots, q, \tag{5.4}$$

Furthermore, we assume that the pair $(A(\theta), B(\theta))$ is controllable and $B(\theta))$ has full column rank for all θ. In this work, the proposed gain-scheduling full state feedback control law is given by

$$u(t) = K(\theta(t))x(t), \tag{5.5}$$

where $K(\theta)$ is the parameter-dependent gain matrix to be determined.

5.2.1 A multi-simplex representation

Before presenting the affine to multi-simplex transformation, we introduce the following definitions.

Definition 1 [13, 14]. A unit simplex Ξ_s is a polytope of s vertices defined by

$$\Xi_s = \left\{ \rho = (\rho_1, \ldots, \rho_s) : \sum_{i=1}^{s} \rho_i = 1, \rho_i \geq 0, i = 1, 2, .., s \right\},$$

where the variable ρ_i varies inside the simplex Ξ_s.

Definition 2 [15]. A multi-simplex Ξ is the Cartesian product of q number of unit simplex, that is,

$$\Xi = \Xi_{N_1} \times \Xi_{N_2} \times \cdots \Xi_{N_q} = \prod_{i=1}^{q} \Xi_{N_i},$$

where the index $(N_1, N_2, \ldots, N_q)$ denotes the dimension of the multi-simplex Ξ, hence any $\rho \in \Xi$ can be decomposed into $(\rho_1, \rho_2, \ldots, \rho_q)$, and each ρ_i belonging to Ξ_{N_i} is represented by $(\rho_{i,1}, \rho_{i,2}, \ldots, \rho_{i,N_i})$ for $i = 1, 2, \ldots, q$.

In order to formulate the affine parameter dependent control problem in multi-simplex representation described in ρ-space, we need to first map every scheduling parameter θ_i into the corresponding unit simplex form with two vertices $(\rho_{i,1}, \rho_{i,2})$.

This can be achieved by utilizing the following general formula [40],

$$\begin{aligned} \rho_{i,1} &= \frac{\theta_i - \underline{\nu}_i}{\bar{\nu}_i - \underline{\nu}_i} \\ \rho_{i,2} &= 1 - \rho_{i,1} = \frac{\bar{\nu}_i - \theta_i}{\bar{\nu}_i - \underline{\nu}_i} \\ \rho_i &= (\rho_{i,1}, \rho_{i,2}) \in \Xi_{N_i}, \ \forall\, i = 1, 2, \cdots, q, \end{aligned} \tag{5.6}$$

For systems with two scheduling parameters, results in four corresponding vertices can be written as:

$$\begin{aligned} For \quad (\underline{\nu}_1, \underline{\nu}_2), \qquad \rho_1 &= \left(\frac{\bar{\nu}_1 - \theta_1}{\bar{\nu}_1 - \underline{\nu}_1}\right) \times \left(\frac{\bar{\nu}_2 - \theta_2}{\bar{\nu}_2 - \underline{\nu}_2}\right) \\ For \quad (\underline{\nu}_1, \bar{\nu}_2), \qquad \rho_2 &= \left(\frac{\bar{\nu}_1 - \theta_1}{\bar{\nu}_1 - \underline{\nu}_1}\right) \times \left(\frac{\theta_2 - \underline{\nu}_2}{\bar{\nu}_2 - \underline{\nu}_2}\right) \\ For \quad (\bar{\nu}_1, \underline{\nu}_2), \qquad \rho_3 &= \left(\frac{\theta_1 - \underline{\nu}_1}{\bar{\nu}_1 - \underline{\nu}_1}\right) \times \left(\frac{\bar{\nu}_2 - \theta_2}{\bar{\nu}_2 - \underline{\nu}_2}\right) \\ For \quad (\bar{\nu}_1, \bar{\nu}_2), \qquad \rho_4 &= \left(\frac{\theta_1 - \underline{\nu}_1}{\bar{\nu}_1 - \underline{\nu}_1}\right) \times \left(\frac{\theta_2 - \underline{\nu}_2}{\bar{\nu}_2 - \underline{\nu}_2}\right) \end{aligned} \tag{5.7}$$

where Ξ_{N_i} denotes a two-dimensional unit simplex defined by

$$\Xi_{N_i} = \left\{ \rho_i = (\rho_{i,1}, \rho_{i,2}) \in \mathbf{R}^2 : \sum_{j=1}^{2} \rho_{i,j} = 1, \rho_{i,j} \geq 0 \right\}, \; i = 1, 2, .., q, \tag{5.8}$$

hence, the two-dimensional unit simplex variable ρ_i is created. Similarly, the rate change of the scheduling parameter $\dot{\theta}_i(t)$ given in (5.4) can also be represented using a unit simplex variable. It follows from the condition that

$$\dot{\rho}_{i,1}(t) + \dot{\rho}_{i,2}(t) = 0\,, \; i = 1, 2, .., q,$$

where

$$\dot{\rho}_{i,1}(t) = \frac{\dot{\theta}_i - \underline{\nu}_i}{\bar{\nu}_i - \underline{\nu}_i}\,, \; \dot{\rho}_{i,2}(t) = -\dot{\rho}_{i,1}(t)\,.$$

Substituting the rate bounds (5.4) into above, we obtain

$$-\frac{\sigma_i}{\bar{\nu}_i - \underline{\nu}_i} \leq \dot{\rho}_{i,j} \leq \frac{\sigma_i}{\bar{\nu}_i - \underline{\nu}_i}\,, \; i = 1, 2, .., q; \; j = 1, 2. \tag{5.9}$$

To establish a two-dimensional unit simplex variable $\dot{\rho}_i$, we first introduce a 2×2 matrix H_i defined by

$$H_i = \begin{bmatrix} -\frac{\sigma_i}{\bar{\nu}_i - \underline{\nu}_i} & \frac{\sigma_i}{\bar{\nu}_i - \underline{\nu}_i} \\ \frac{\sigma_i}{\bar{\nu}_i - \underline{\nu}_i} & -\frac{\sigma_i}{\bar{\nu}_i - \underline{\nu}_i} \end{bmatrix}, i = 1, 2, \ldots, q\,, \tag{5.10}$$

hence the unit-simplex of dimension 2 can be defined by

$$\Omega_{N_i} = \left\{ \phi_i \in \mathbf{R}^2 : \phi_i = \sum_{j=1}^{2} \eta_{i,j} H_i^j, \eta_i \in \Xi_{N_i} \right\}, \; i = 1, 2, \ldots, q\,, \tag{5.11}$$

where $\eta_i = (\eta_{i,1}, \eta_{i,2})$ and H_i^j denotes the j^{th} column of matrix H_i. Therefore, the unit-simplex variable $\dot{\rho}_i \in \Omega_{N_i}$ is created. Finally, the scheduling parameters $(\theta, \dot{\theta})$ with prescribed bounds can now be transformed into multi-simplex domain through Cartesian product of defined unit-simplexes as follows,

$$(\rho, \dot{\rho}) \in \Xi \times \Omega = \prod_{i=1}^{q} \Xi_{N_i} \times \prod_{i=1}^{q} \Omega_{N_i}\,.$$

Note that $\rho = (\rho_1, \rho_2, \ldots, \rho_q) \in \Xi$ is defined inside the multi-simplex of dimension $(2, 2, \ldots, 2)$. Therefore, the affine parameter dependent system matrices in (5.1) can be represented in terms of

ρ. For instance, from the first equation in (5.6), we obtain that

$$\theta_i = \underline{\nu}_i + \rho_{i,1}(\bar{\nu}_i - \underline{\nu}_i)\,. \tag{5.12}$$

Substituting the above into (5.2) yields

$$A(\theta) = A_0 + \sum_{i=1}^{q} \underline{\nu}_i A_i + \sum_{i=1}^{q} (\bar{\nu}_i - \underline{\nu}_i) A_i \rho_{i,1} = A(\rho)\,.$$

By following the above parameter transformation, we assume that the LPV control systems (5.1) described in θ can be transformed into an equivalent system representation in terms of ρ as

$$\begin{cases} \dot{x}(t) = A(\rho(t))x(t) + B_u(\rho(t))u(t) + B_w(\rho(t))w(t) \\ y(t) = C(\rho(t))x(t) + D_u(\rho(t))u(t) + D_w(\rho(t))w(t) \end{cases} \tag{5.13}$$

The LPV control gain matrix defined in (5.5) will now be depending on the convex scheduling parameter ρ. Therefore, the closed-loop system in multi-simplex representation can be written as

$$\begin{aligned} \dot{x}(t) &= A_{cl}(\rho)x(t) + B_w(\rho)w(t) \\ y(t) &= C_{cl}(\rho)x(t) + D_w(\rho)w(t) \end{aligned} \tag{5.14}$$

where

$$A_{cl}(\rho) = A(\rho) + B_u(\rho)K(\rho)\,,\ C_{cl}(\rho) = C(\rho) + D_u(\rho)K(\rho)\,. \tag{5.15}$$

It should be noted that the proposed LPV control system design and analysis process will be performed based on (5.13), therefore the actual control implementation will require converting controller gain matrix from ρ-domain to θ-domain [3, 12, 13]. In particular, [3] paper presents explicitly the conversion of an LPV model with three scheduling parameters.

5.2.2 H_∞ control performance

The proposed LPV control problem can be now stated as follows: For system (5.13), a given $\gamma_\infty > 0$ and any given scheduling parameter pair $(\rho, \dot{\rho})$, design a gain-scheduling state-feedback controller of the form

$$u(t) = K(\rho)x(t),$$

that exponentially stabilizes the closed-loop system (5.14) with guaranteed $\mathcal{H}_\infty$ performance γ_∞, defined by

$$\sup_{(\rho_i,\dot{\rho}_i)} \sup_{\substack{w\in L_2 \\ w\neq 0}} \frac{||z||_2}{||w||_2} < \gamma_\infty \,. \tag{5.16}$$

Note that γ_∞ is a measure of system robustness. The next theorem provides the synthesis solution to the $\mathcal{H}_\infty$ control problem.

Theorem: [12, 106, 107]. The proposed $\mathcal{H}_\infty$ control problem is solvable for (5.13), if there exist a scalar $\varepsilon > 0$, symmetric positive-definite continuously differentiable matrix $P(\rho)$, and matrices $V(\rho)$ and $W(\rho)$, satisfying the following parameterized linear matrix inequality (PLMI)

$$\begin{bmatrix} \Gamma_1(\rho) & * & * & * \\ \Gamma_2(\rho) & -\varepsilon(V(\rho)V(\rho)') & * & * \\ \Gamma_3(\rho) & \Gamma_4(\rho) & -I & * \\ B_w(\rho)' & 0_{g\times n} & D_w(\rho)' & -\gamma_\infty^2 I \end{bmatrix} < 0 \tag{5.17}$$

where $*$ denotes the symmetric entry, and $\Gamma_1(\rho)$, $\Gamma_2(\rho)$, $\Gamma_3(\rho)$ and $\Gamma_4(\rho)$ are given by

$$\begin{aligned} \Gamma_1(\rho) =& A(\rho)V(\rho) + B_u(\rho)W(\rho) + V(\rho)'A(\rho)' + W(\rho)'B_u(\rho)' + \frac{\partial P(\rho)}{\partial \rho}\dot{\rho}, \\ \Gamma_2(\rho) =& P(\rho) - V(\rho) + \varepsilon(A(\rho)V(\rho) + B_u(\rho)W(\rho))', \\ \Gamma_3(\rho) =& C(\rho)V(\rho) + D_u(\rho)W(\rho), \\ \Gamma_4(\rho) =& \varepsilon C(\rho)V(\rho) + \varepsilon D_u(\rho)W(\rho)\,. \end{aligned}$$

Furthermore, if a solution exists to the above PLMI, then the gain-scheduling controller gain is given by $K(\rho) = W(\rho)V(\rho)^{-1}$.

Note that the PLMI (5.17) is an infinite dimensional LMI, which is generally difficult to solve. Few approaches are available in solving a PLMI; see for instance [41, 42, 43, 44], however this work adapts the method presented in [13].

5.2.3 Virtual control system modeling

In this section, we propose a novel modeling approach for a control system with state constraints. The goal of virtual control design is to develop an effective way to prevent violating the prescribed state constraints, so as to improve system performance while avoiding hardware to be saturated. It will be demonstrated that this can be accomplished by introducing augmented profile functions to the control systems.

Recall the LPV control system described in (5.1). Assuming that there are constrained states in the system, during the design phase, virtual mechanisms are proposed in the form of augmented profile function and introduced in the system dynamics where the constrained states are needed. The resulting system with virtual mechanism are as follows:

$$\dot{x}(t) = (A(\theta) + A_v(\theta))x(t) + B_u(\theta)u(t) + B_w(\theta)w(t) \tag{5.18}$$

where $A(\theta)$ contains the original system dynamics, and $A_v(\theta)$ the added virtual mechanism dynamics. In order to make it possible to implement the virtual dynamics into control law, the following condition needs to be satisfied for and x and u_v

$$A_v(\theta)x(t) = B_u(\theta)u_v(t)\,. \tag{5.19}$$

Or equivalently for any θ

$$Span(A_v(\theta)) \subseteq Span(B_u(\theta))\,. \tag{5.20}$$

In most practical control applications, the augmented profile function matrix $A_v(\theta)$ can be chosen in advance based on the nature of state constraints. Specifically, in this chapter, these profile functions are in the form of nonlinear springs and dampers so that the associated stiffness and damping coefficients increase dramatically as the constrained states move closer to their bounds. During the control implementation phase, the proposed virtual dynamics are augmented in to the control input $\hat{u}$ as follows

$$\hat{u} = u + (B_u^T(\theta)B_u(\theta))^{-1}B_u^T(\theta)A_v(\theta)x(t), \tag{5.21}$$

where $u_v(t) = (B_u^T B_u)^{-1} B_u^T A_v x(t)$ denotes the virtual control. Note that $B_u(B_u^T B_u)^{-1} B_u^T A_v x(t) = A_v x(t)$ due to (5.20) and u is the LPV control law to be designed.

5.3 LPV Modeling and Control of Smart Airfoil

This section presents the application of the proposed LPV virtual control approach to the flutter suppression problem for a smart airfoil; see Figure 5.1. Additional information about the smart airfoil can be found in [1, 2, 104]. As shown in Figure 5.1, the smart airfoil model is from the NACA0012 airfoil with a groove along its chord, in which a small moving mass is placed so that the vibrational motion of the airfoil is suppressed through dynamic coupling to the plunging and pitching motion of the airfoil.

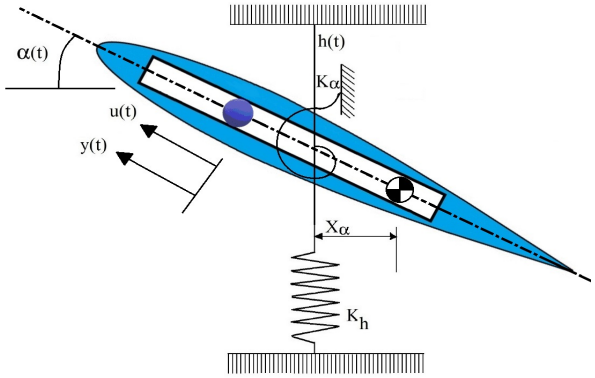


Figure 5.1: Smart airfoil.

The smart airfoil model in the state-space representation can be described in form of (5.1) (see [3]), where $x(t) = [\bar{h}, \alpha, \dot{\bar{h}}, \dot{\alpha}, \bar{y}, \dot{\bar{y}}]^T$ representing the normalized plunge, pitch attitude, normalized plunge rate, pitch rate, normalized control mass displacement and its rate, respectively. The system matrices are given by (22), where θ_1 (will be introduced in the virtual control phase) and θ_2 are two scheduling parameters. The detailed derivation of (22) can be found in [3].

$$A(\theta) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{\bar{r}_\alpha^2 \omega_h^2}{q\omega_\alpha^2} & \frac{\bar{r}_\alpha^2(6.68\theta_2-2.4)}{q\mu} + \frac{(6.68\theta_2-2.4)\bar{e}\bar{x}_\alpha}{q\mu} - \frac{\bar{r}_\alpha^2\bar{x}_\alpha}{q} & \frac{2\bar{r}_\alpha^2\theta_2}{q\mu} + \frac{2\theta_2\bar{e}\bar{x}_\alpha}{q\mu} & 0 & \frac{-\bar{x}_\alpha\beta}{q} & 0 \\ \frac{-\bar{x}_\alpha\omega_h^2}{q\omega_\alpha^2} & \frac{\bar{x}_\alpha(2.4-6.68\theta_2)}{q\mu} - \frac{(6.68\theta_2-2.4)\bar{e}(1+\beta)}{q\mu} - \frac{\bar{r}_\alpha^2(1+\beta)}{q} & \frac{-2\bar{x}\theta_2}{q\mu} - \frac{2\theta_2\bar{e}(1+\beta)}{q\mu} & 0 & \frac{\beta(1+\beta)}{q} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \bar{g} & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B_u(\theta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_w(\theta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}', \quad C(\theta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$D_u(\theta) = [0], \qquad D_w(\theta) = [0], \qquad q = \left[-[\bar{r}_\alpha^2(1+\beta) - \bar{x}_\alpha^2]\right] \tag{5.22}$$

5.3.1 Virtual control modeling

The idea of virtual control is to introduce virtual mechanisms into the model during the control design phase to prevent hard constrained states from moving close to the hard limits, and then in the implementation phase to transfer the added virtual dynamics (mechanisms) into the controller law. For the smart airfoil example, the virtual part is introduced in terms of springs and dampers at both ends of groove to prevents the control mass from hitting the groove boundaries. Note that the virtual mechanisms can be added freely as long as (5.20) is satisfied. Recalling (5.18), in the design phase the system model after introducing the virtual mechanisms is

$$\dot{x}(t) = \left[A(\theta) + A_v(\theta)\right] x(t) + B_u(\theta)u(t) + B_w(\theta)w(t) \tag{5.23}$$

Note that B_u has only one nonzero entry (the 6th). To satisfy (5.20), all entries of $A_v(\theta)$ should be zero except the last row. For the airfoil flutter control, A_v is selected as

$$A_v(\theta) = \begin{bmatrix} \mathbf{0}_{5\times4} & \mathbf{0}_{5\times2} \\ \mathbf{0}_{1\times4} & [-k(\theta_1) \quad -c(\theta_1)] \end{bmatrix} \tag{5.24}$$

Note that for the selected A_v, it is equivalent to adding virtual springs and dampers at the both ends of the groove if the nonlinear spring and damper coefficients are modeled as a functions of

the scheduling parameter θ_1 as shown in Figure 5.2. The spring stiffness $k(\theta_1)$ and the damping coefficient $c(\theta_1)$ are carefully designed to generate enough force to stop and reverse the motion of the control mass at both ends of the airfoil groove. According to Figure 5.2, the values of the $k(\theta_1)$ and $c(\theta_1)$ in the vicinity of the airfoil groove between -0.4 and 0.4 are equal to zero and hence the mass can move freely in this area. But near both ends, i.e., between 0.4 and 0.5 (between -0.5 and -0.4) their values increase by following a tangent profile to generate forces opposite to the moving direction and to stop the motion and/or reverse the moving direction.

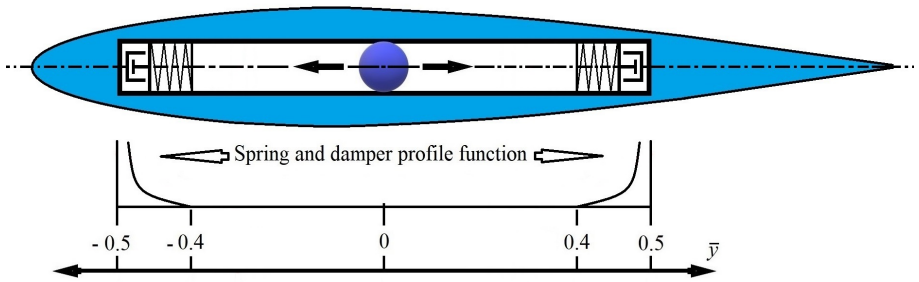


Figure 5.2: Virtual spring and damper

The virtual spring and damper coefficients were chosen as follows (also see Figure 5.2),

$$\begin{cases} k = 0, c = 0, & \text{when } |\bar{y}| \leq 0.4 \\ k(\theta_1) = 80 \tan(\frac{\pi(\theta_1)}{0.2}), & \text{when } |\bar{y}| > 0.4 \\ c(\theta_1) = 6 \tan(\frac{\pi(\theta_1)}{0.2}), & \text{when } |\bar{y}| > 0.4 \end{cases}$$

where $\theta_1 = |\bar{y}| - 0.4$. Noted that the functions $k(\theta_1)$ and $c(\theta_1)$ can be chosen in different ways, depending on applications.

Follow up of (5.6) the smart airfoil example has two scheduling parameters, i.e. $q = 2$, then variables in the multi-simplex domain can be depicted as

$$\rho_1 = (\rho_{1,1}, \rho_{1,2}), \rho_2 = (\rho_{2,1}, \rho_{2,2})$$

The LPV $\mathcal{H}_\infty$ design, based on Theorem 5.17, is performed over multi-simplex convex domain Ξ

and the resulting gain-scheduling controller is in terms of

$$\begin{aligned} W(\rho) &= \sum_{j_1=1}^{2}\sum_{j_2=1}^{2} \mathcal{W}_{j_1,j_2}\rho_{j_1,j_2}, \\ V(\rho) &= \sum_{j_1=1}^{2}\sum_{j_2=1}^{2} \mathcal{V}_{j_1,j_2}\rho_{j_1,j_2}, \end{aligned} \tag{5.25}$$

In order to calculate the affine feedback gain matrix in (5.5), the inverse transformation described in [1, 12, 13] are used. For example,

$$\begin{aligned} W_0 &= \frac{1}{16}\sum_{j_1=1}^{2}\sum_{j_2=1}^{2} \mathcal{W}_{j_1,j_2}, \\ W_i &= \frac{1}{16\bar{v}_i}\sum_{j_1=1}^{2}\sum_{j_2=1}^{2} (-1)^{j_i+i}\mathcal{W}_{j_1,j_2}, \quad i = 1,2 \end{aligned} \tag{5.26}$$

Hence, for two scheduling parameters, $W(\theta)$ and $V(\theta)$ can be written in the affine domain as

$$\begin{aligned} W(\theta) &= W_0 + \theta_1 W_1 + \theta_2 W_2, \\ V(\theta) &= V_0 + \theta_1 V_1 + \theta_2 V_2 \end{aligned} \tag{5.27}$$

where W_i $(i = 0, 1, 2)$ are constant matrices determined by (5.26) and V_i $(i = 0, 1, 2)$ can be obtained in a similar way. Finally, control gain matrix $K(\theta)$ defined in (5.5) can be calculated by

$$K(\theta) = W(\theta)V^{-1}(\theta) \tag{5.28}$$

For the case with more than two scheduling parameters, one may consult references [12, 13, 40]. For the solution of the convex optimization of (3.30), SeDuMi [46] solver with optimization packages ROLMIP [45] and YALMIP [11] are used. Then, the virtual control law in (5.21) can be formed as

$$\hat{u} = W(\theta)V^{-1}(\theta)x + (B_u^T(\theta)B_u(\theta))^{-1}B_u^T(\theta)A_v(\theta)x\,, \tag{5.29}$$

The virtual LPV controller synthesis solution steps generally can be considered as eight steps. Firstly, scaling scheduling parameter introduced. Secondly, introducing the virtual dynamics. Thirdly, change of variables from affine representation domain to the multi-simplex (convex) domain is performed. The fourth step is modeling the rates of variation of scheduling parameters in convex domain. In the fifth stage, performing the PLMI synthesis conditions. The six stage is to

perform PLMI relaxation to transfer the infinite dimension PLMI into finite dimension LMI to be solved using optimization package. Then the seven stage involve to utilize inverse transformation to transfer from the multi-simplex representation into the affine representation. Then finally the eight stage is the controller implementation with virtual dynamics compensation. The steps are depicted in the following diagram Figure 5.3.

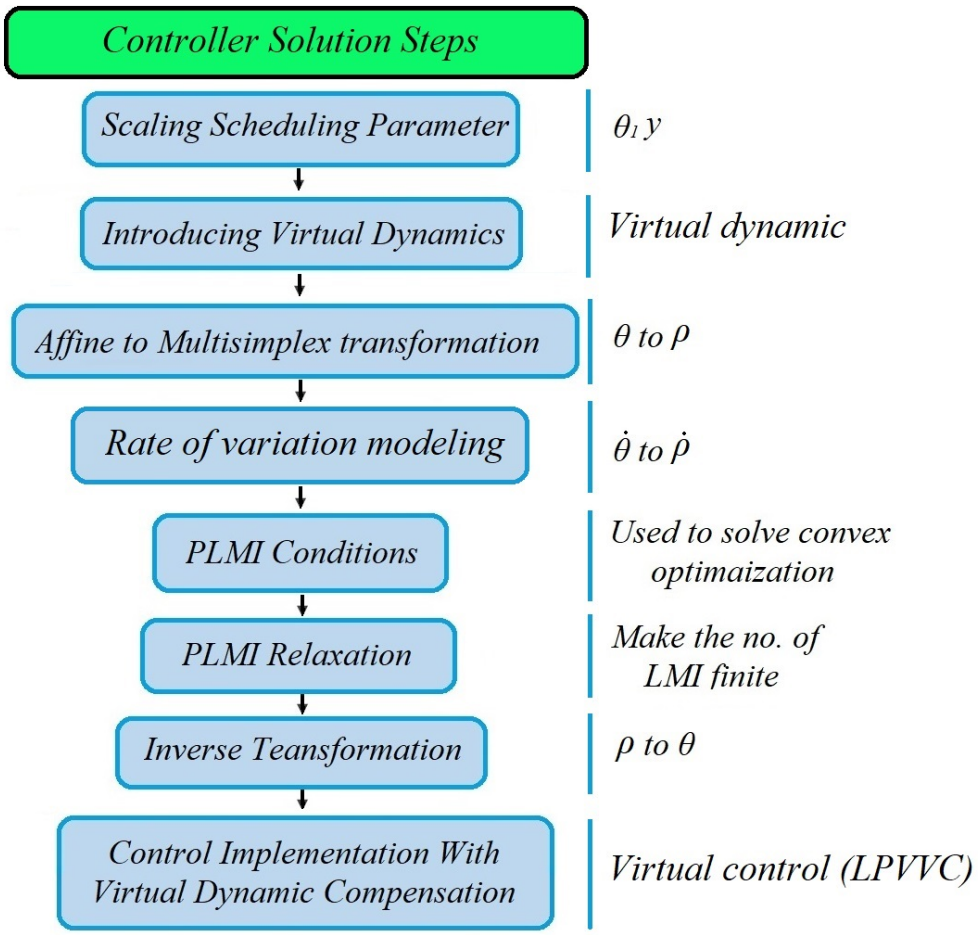


Figure 5.3: Controller steps.

5.4 Simulations investigation

5.4.1 Control design results

The parameters of the smart airfoil example used in this chapter are shown in Table 5.1 and the air speed θ_2 varies from 0.5 to 2.92 nondimensional speed which is equivalent to from $4m/s$ to $23.7m/s$. The virtual control idea has its unique advantage for designing controller with hard constraints on state variables. In order to assess the performance of the proposed LPV virtual controller (LPVVC) with hard constraints, numerical studies are conducted and compared with the existing results in the next subsection.

Table 5.1: [1, 2] Parameters used for the smart airfoil example.

Parameter	Value	Units	Parameter	Value	Units
b (m)	0.127	(m)	initial $\bar{h}$	0	Nondimensionalized
$\bar{e}$	0.35	Nondimensionalized	initial $\dot{\bar{h}}$	0	Nondimensionalized
$\bar{r}_\alpha^2$	0.388	Nondimensionalized	α_S	0.01	(rad)
$\bar{x}_\alpha$	0.25	Nondimensionalized	α_L	0.6	(rad)
airspeed $\bar{V}$	2.92	Nondimensionalized	initial $\dot{\alpha}$	0	(rad/sec)
ω_α	64.1	(rad/sec)	β	0.01	Nondimensional
ω_h	55.9	(rad/sec)	μ	152	Nondimensional

5.4.2 Simulation results

In order to highlight the significance of the proposed virtual control concept, three comparison studies are conducted in this section. The first comparison is with the method used in [3], where a conventional LPV control (LPVN) strategy is utilized. Two pairs of springs and dampers are physically installed at both ends of the airfoil groove from (|0.35| to |0.5|) [3].

Starting with some mild initial condition (α = 0.2 rad), one can see, from Figure 5.4, that both LPVN and proposed LPVVC controllers are able to stay within the groove hard constraint, but for the case of harsh initial conditions (for example, 0.6 rad) it can be seen, from Figure 5.5, that the LPVN control is not able to keep the moving mass within the groove hard constraints while the proposed LPVVC does. The system responses shown in Figures 5.4 and 5.5 confirm that the

proposed LPVVC virtual control is capable of keeping the actuation mass within its hard constraint and preventing actuator saturating under both small and large initial conditions.

Furthermore, recall Figure 5.2, where the black curve represents the proposed tangent profile function used in LPVVC virtual control. Therefore, the actuation mass moves freely between -0.4 and 0.4, but for the LPVN control [3], it reduces to be between -0.35 and 0.35. This indicates that utilizing the virtual tangent profile function increases free moving range for the control mass, making the hard constrained control more close to the case without hard constraints. This is benefited from the selected tangent profile function, which increases the linear region to be between -0.4 and 0.4 from between -0.35 and 0.35.

The second study is presented in Figure 5.6, where the LPVVC control is compared with a nonlinear controller (NLC) in [2]. The effectiveness of the proposed controller is clearly demonstrated, where the nonlinear controller is unable to satisfy the hard constraints for the given initial condition while the LPVVC does. In addition, the overall performance is also enhanced as well as the control effort is dramatically reduced from that of the nonlinear controller in [2].

The last comparison is with a regular LPV controller (LPVR) without considering the hard constraint requirement (see Figure 5.7). It can be clearly observed that since the LPVR control does not consider the hard constraints, it suppresses the airfoil flutter by regulating the control mass outside the hard constraints unrealistically. Again, the proposed LPVVC controller fulfills the hard constraint requirement effectively.

A parametric study is also conducted and the results are presented in Tables 5.2 and 5.3 below in terms of the signal $\mathcal{L}_\infty$ and $\mathcal{L}_2$ norms ($\|\bar{y}\|_\infty^2 = \sup_{t\geq 0}(\bar{y}^T\bar{y})$, $\|\bar{y}\|_2^2 = (\int_0^\infty \bar{y}^T\bar{y}dt)$) to show the control and response performance enhancement of the proposed method over LPVN, NLC, and LPVR. To make a fair comparison, initial conditions for all simulations are set to $\alpha = 0.6$. One can see a significant improvement of $\mathcal{L}_2$ norm (see Table 5.3). Note that the last two columns of Table 5.3 show percentages of improvement of the LPVVC control over the NLC [2] and LPVN [3], respectively. Also, the red boxes in Table 5.2 shows that the other methods violates the hard constraints limit.

It is very clear according to the above results that the proposed LPVVC control is a promising method for dealing with hard constrained control problems with the help of flexibly choosing various virtual mechanisms.

Table 5.2: $\|.\| \infty$ comparisons between LPVVC, LPVN [3], NLC [2] and LPVR.

$\|.\|_\infty$	LPVR	NLC [2]	LPVN[3]	LPVVC
$\bar{h}$	0.1429	0.2444	0.28736	0.2715
α	0.6	0.6	0.6	0.6
$\bar{y}$	7.8091	0.8683	0.9205	0.4319
$\bar{u}$	47.979	619.82	4.917	4.494

Table 5.3: $\|.\|_2$ comparisons between LPVVC, LPVN [3] and NLC [2].

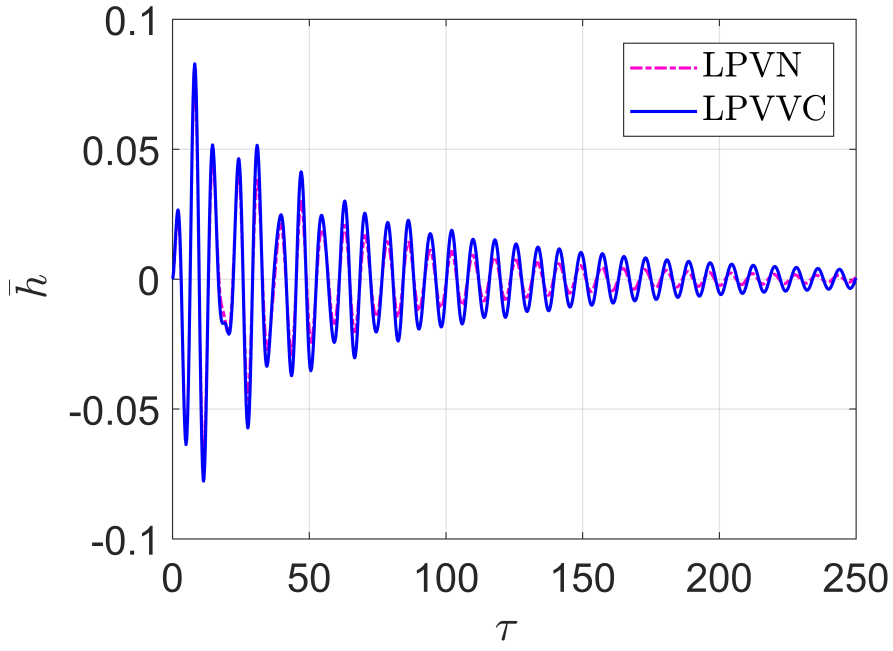
$\|.\|_2$	NLC[2]	LPVN[3]	LPVVC	[2] * %	[3]**%
$\bar{h}$	30.216	27.771	26.59	12	4.2
α	60.868	45.592	49.56	18.5	-8.7
$\bar{y}$	254.07	115.57	54.57	78.5	32
$\bar{u}$	6802.3	519.59	464.99	93.1	10.5

* Convergence rate with respect to NLC[2], ** Convergence rate with respect to LPVN[3]

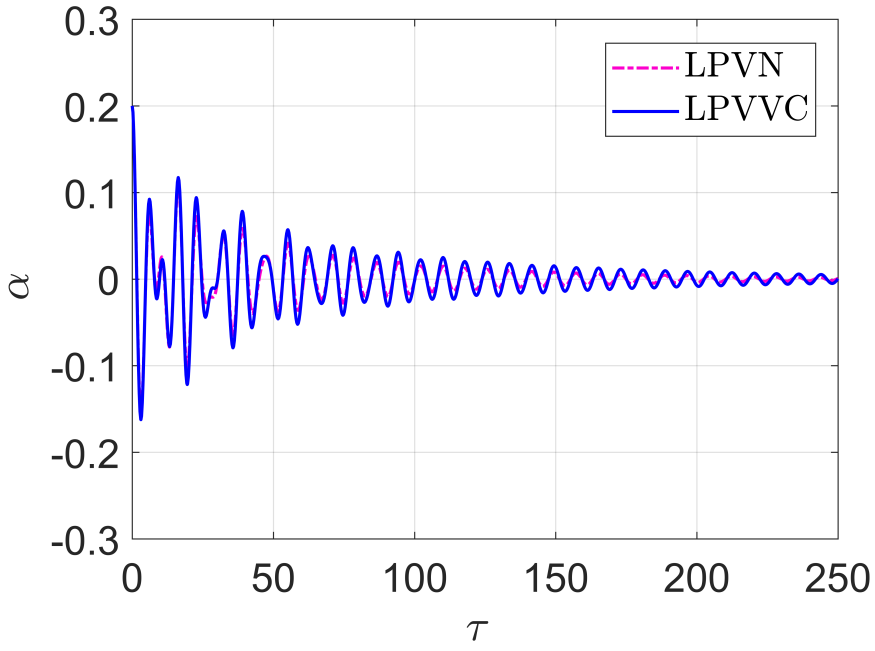
5.5 Summary

This chapter introduced and synthesized a novel LPV virtual control (LPVVC) scheme with hard constraints. The main idea of virtual control is to introduce virtual mechanisms near the state hard constraints to confine state movement into hard constrained neighborhood and hence prevent the states from violating the hard constraints, and during the control implementation, the virtual mechanisms are augmented with the designed LPV controller to form LPVVC. The proposed control scheme is applied to the flutter suppression of a smart airfoil to demonstrate its effectiveness and ability of performance enhancement. The state-feedback LPV (gain-scheduling) control with guaranteed $\mathcal{H}_\infty$ performance is used for designing controller based on the model with virtual mechanisms. The virtual mechanisms used in this chapter are in terms of springs and dampers located at both ends of the airfoil groove to prevent the control mass from moving outside

of the groove. Note that the virtual mechanisms are not limited to springs and dampers. The performance of the designed LPVVC controller is compared with the conventional LPV control with hard constraints (LPVN), nonlinear control (NLC), and regular LPV control (LPVR) and showed significant improvement. For instance, the control mass $\mathcal{L}_2$ norm is reduced by 77.5% over the NLC and 35% over the LPVN.

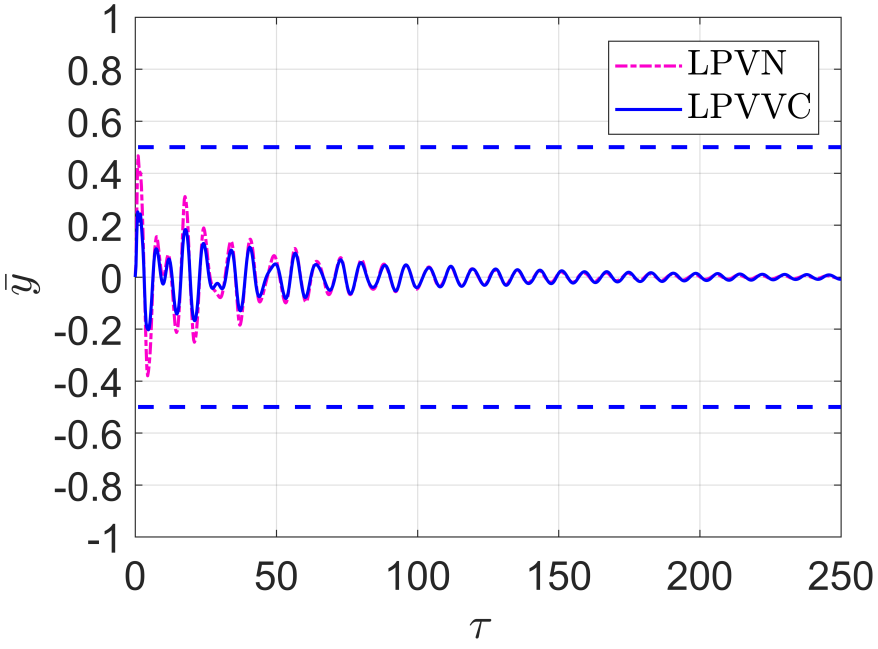


(a)

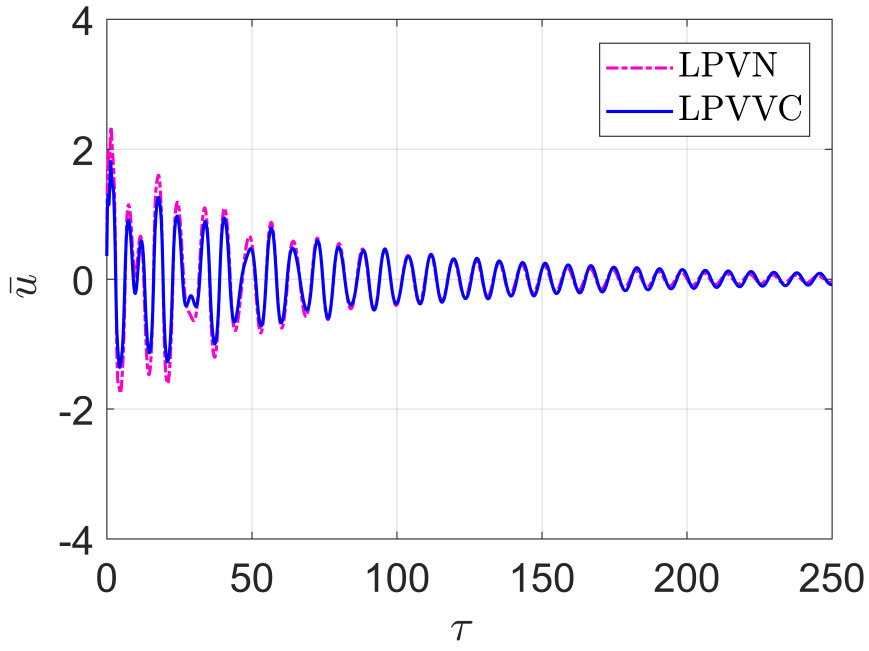


(b)

Figure 5.4: LPVVC vs LPVN, I.C($\alpha = 0.2$).

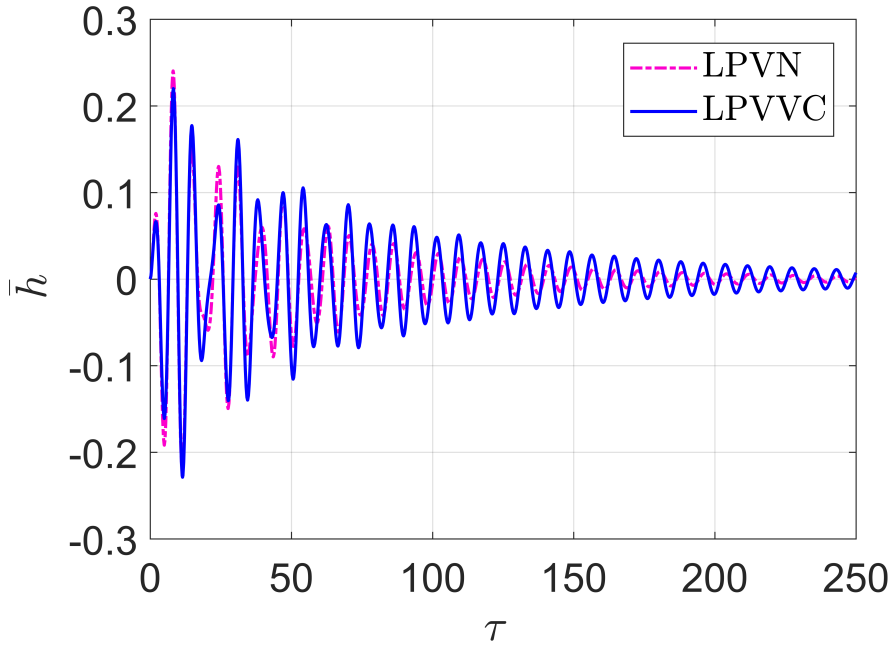


(c)

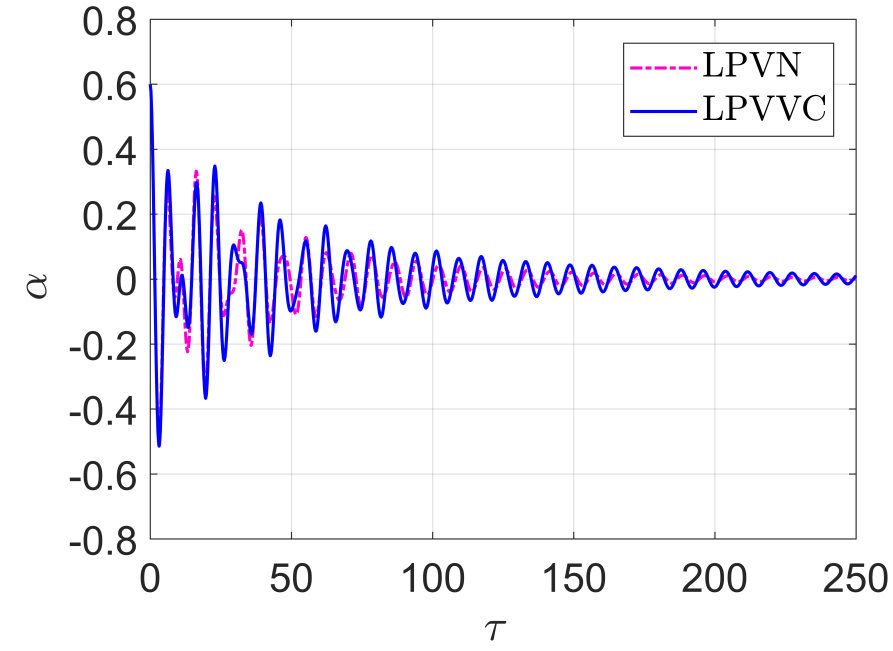


(d)

Figure 5.4: LPVVC vs LPVN, I.C(α = 0.2).

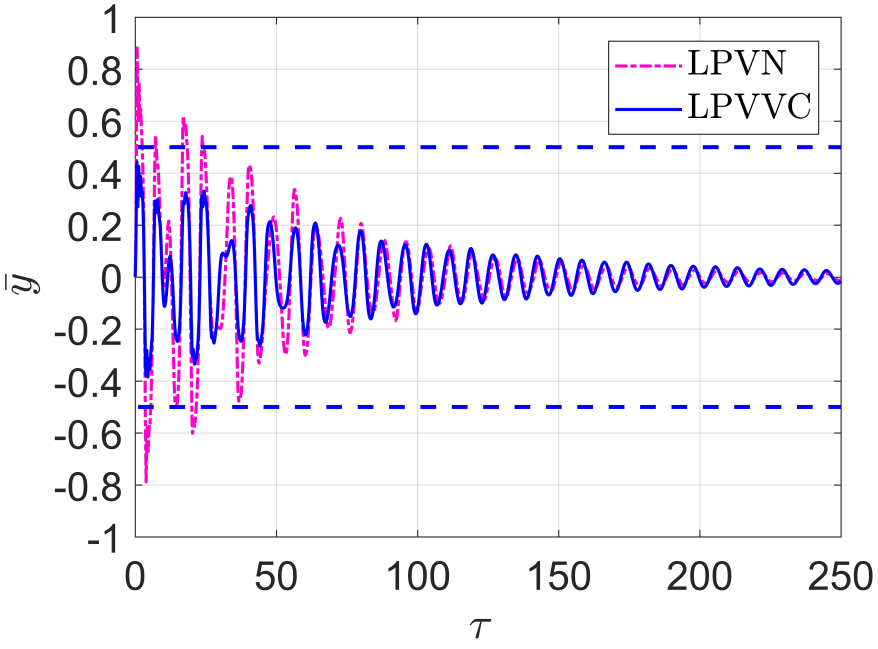


(a)

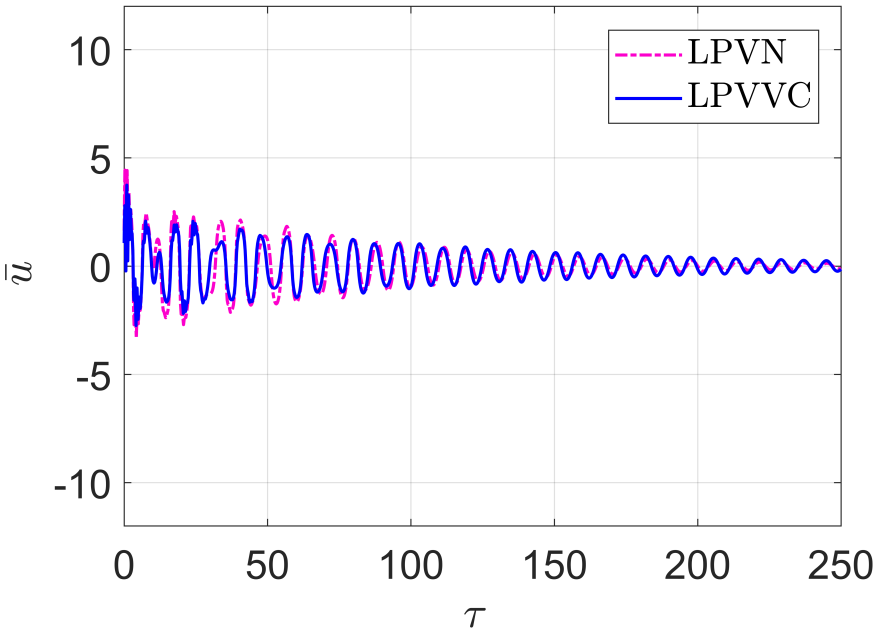


(b)

Figure 5.5: LPVVC vs LPVN, I.C($\alpha = 0.6$).

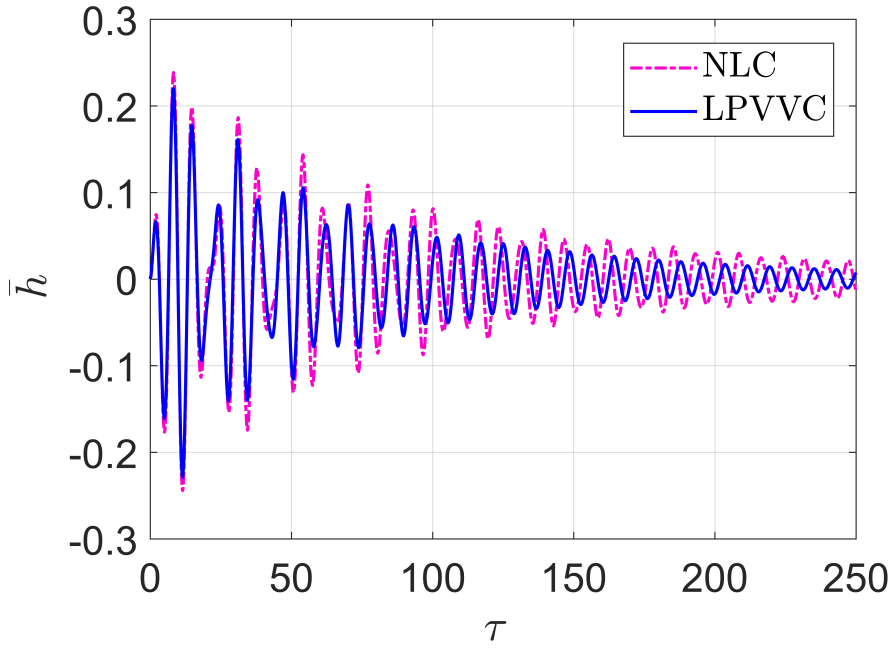


(c)

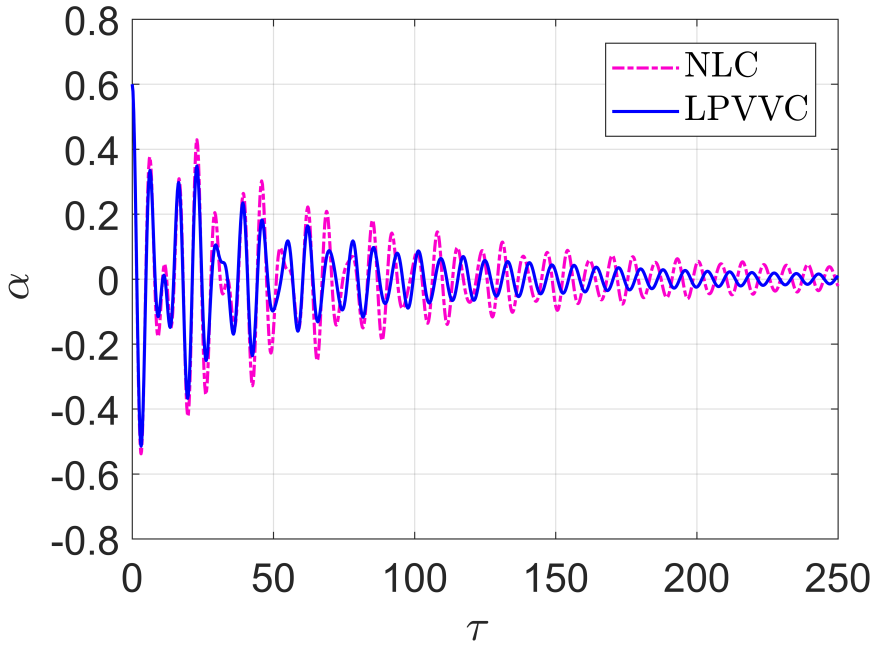


(d)

Figure 5.5: LPVVC vs LPVN, I.C($\alpha = 0.6$).

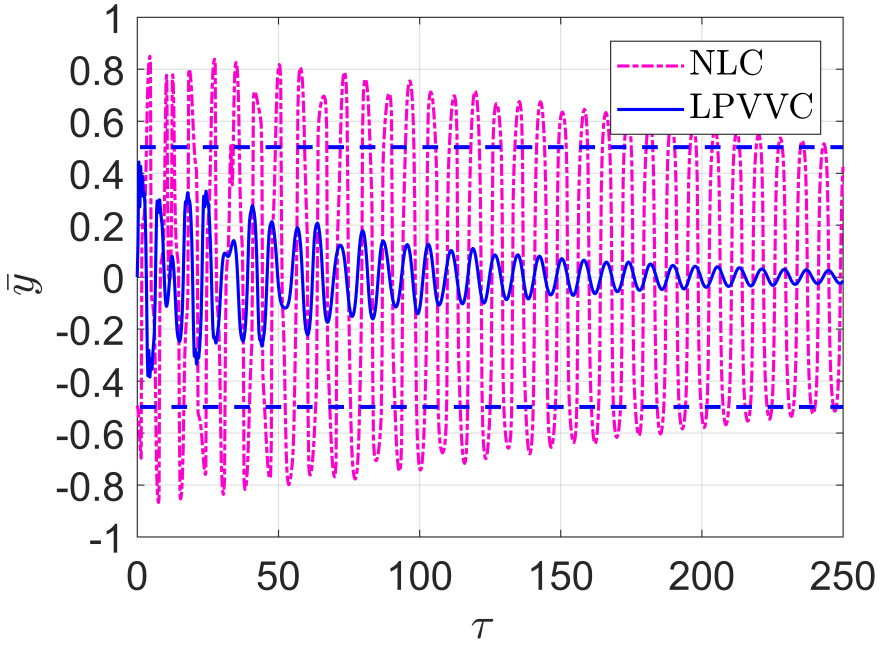


(a)

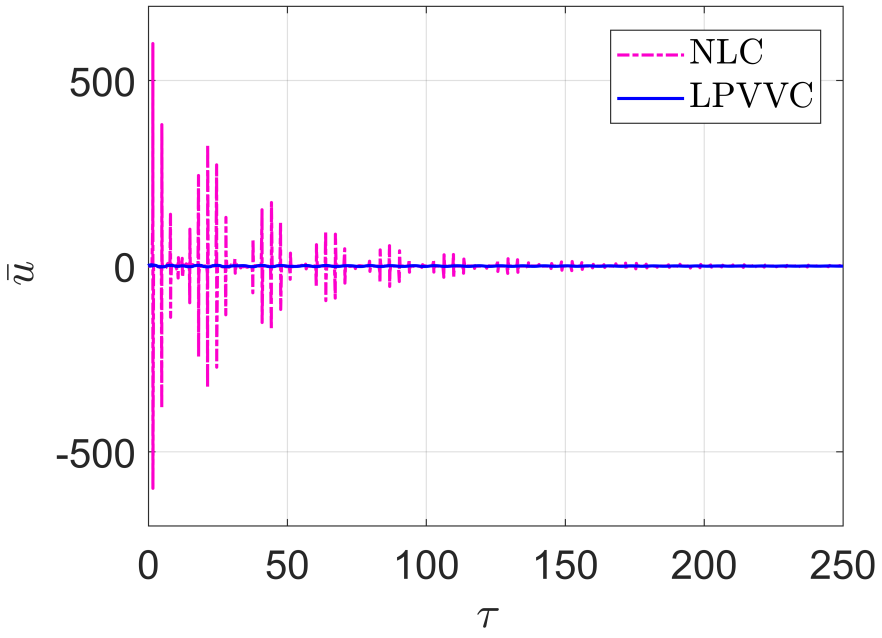


(b)

Figure 5.6: LPVVC vs NLC.

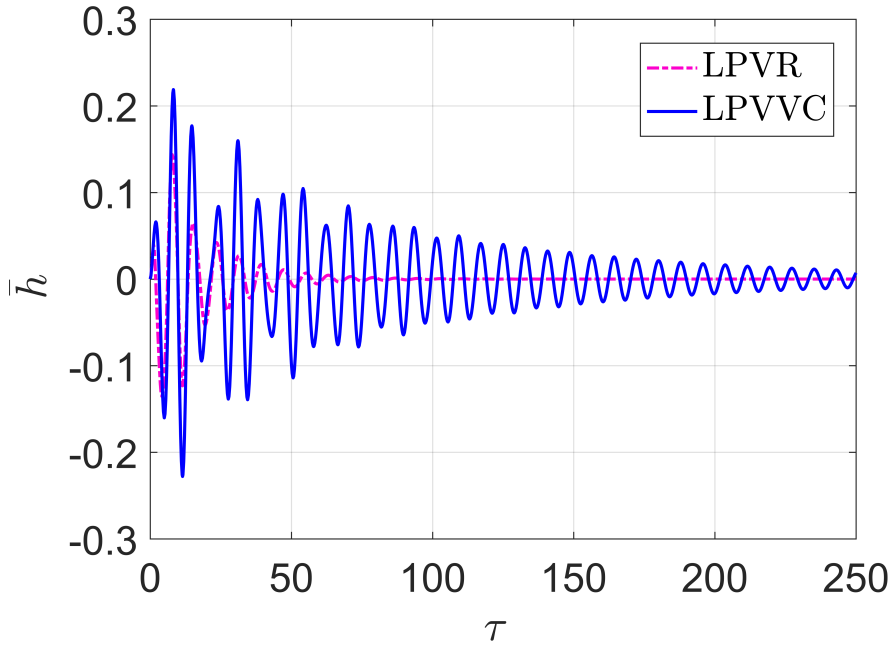


(c)

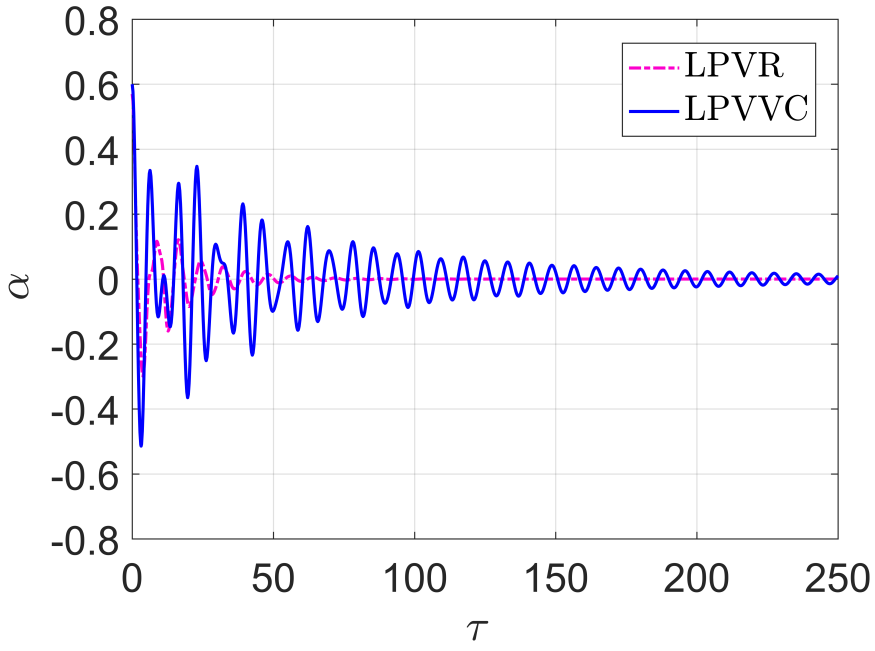


(d)

Figure 5.6: LPVVC vs NLC.

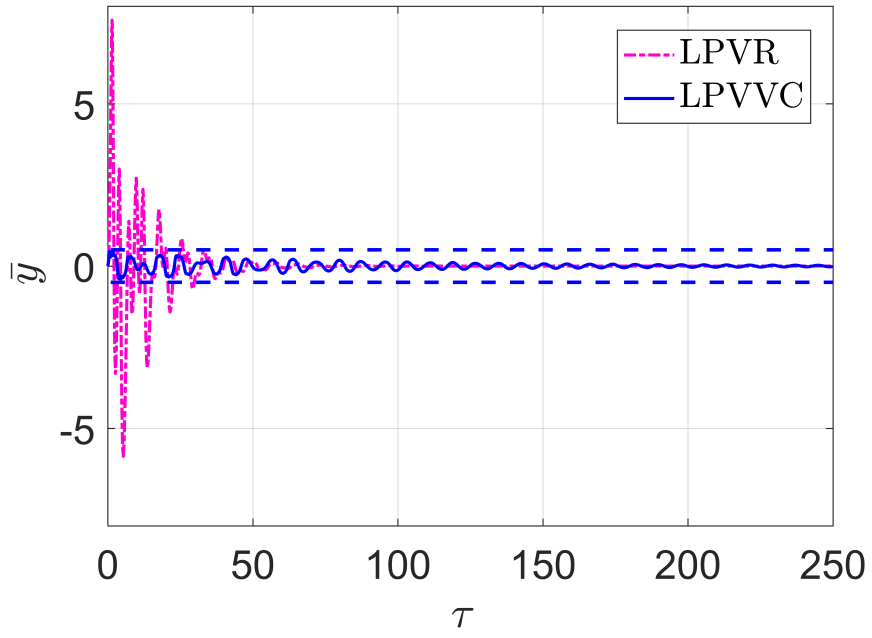


(a)

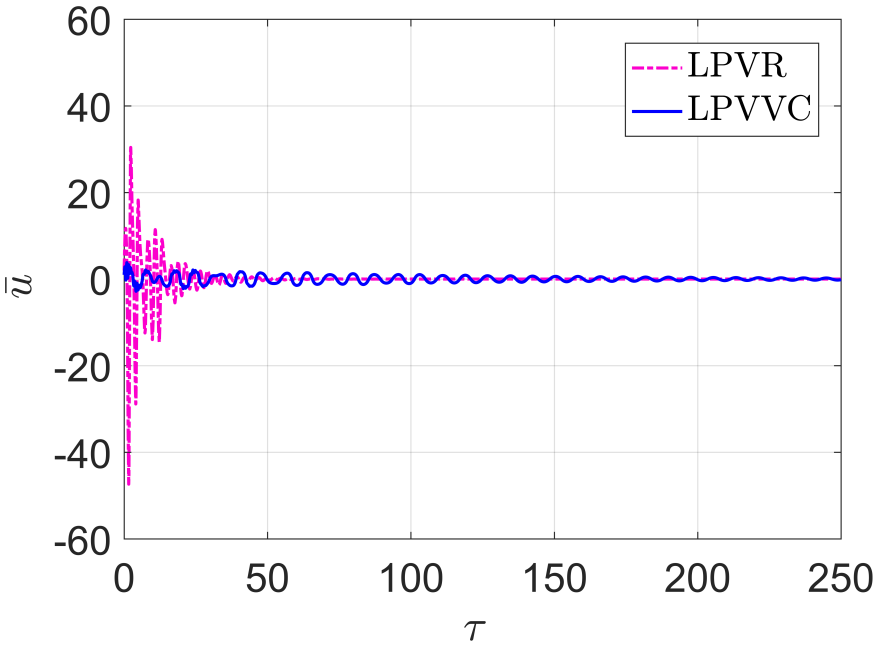


(b)

Figure 5.7: LPVVC vs LPVR.



(c)



(d)

Figure 5.7: LPVVC vs LPVR.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORKS

The dissertation introduced methods of handling actuator saturation for LTI and LPV systems. Numerical simulations, comparisons with other methods and parametric studies were introduced to show the effectiveness and the power of the proposed methods. This chapter provides some conclusions and proposes some future perspectives for the methods.

6.1 Conclusions

6.1.1 Scaling scheduling parameter LPV control with application to flutter suppression of smart airfoil

Chapter three presented two novel LPV modeling and control design techniques for a smart airfoil model that utilizes a moving mass for flutter suppression. The LPV gain-scheduling state feedback controllers based on the corresponding models were proposed. In the first proposed modeling and control technique (LPV-1), the moving mass position was used as the scaled scheduling parameter, whereas in the second modeling and control technique (LPV-2), nonlinear springs and dampers were added to the ends of the groove to constrain the moving mass. Moreover, a modeling and control technique to reduce the number of scheduling parameters (LPV-2A) hence to reduce conservativeness is introduced. The performance of the proposed LPV controllers was compared with an earlier nonlinear controller from the literature, and the results clearly demonstrated the advantages and the effectiveness of the proposed LPV modeling and control techniques in active flutter suppression. In addition, the proposed LPV control design was also able to significantly increase the flutter airspeed and keeps actuator movements inside its bounds.

6.1.2 Virtual control LTI with application to beam and ball system

In chapter four, novel hard constraints virtual controller is proposed and synthesized for LTI continuous systems as a solution for the saturation problem. The method is applied for Ball and Beam System with LQR-based MPC state feedback control to show the effectiveness of the proposed method. The simulations showed the effectiveness and the performance of the proposed method in handling the hard constraints without saturation of the actuator. The main advantages of the proposed method is utilizing the linear time invariant design to handle nonlinear hard constraints, presenting extra tuning parameters to the controller (spring and damper constants), and the simplicity in controller implementation. Also the method can be extended to the continuous LTI systems with minor effort. In addition, the method is aimed to be applied for many other applications with hard constraints and the other controller design theories.

6.1.3 Virtual control LPV with application to smart airfoil flutter suppression

In chapter five, novel hard constraints virtual controller is proposed and synthesized. The main idea of virtual control is to introduce virtual mechanisms near the state hard constraints to confine state movement into hard constrained neighborhood and hence prevent the states from violating the hard constraints, and during the control implementation, the virtual mechanisms are augmented with the designed LPV controller to form LPVVC. H_∞ state feedback guaranteed performance control is used. The method is applied for smart airfoil flutter suppression example to show the effectiveness of the proposed method. Parametric study and numerical simulation are presented. Comparisons studies with conventional LPV hard constrained control, nonlinear control, and regular LPV control without considering hard constraints are conducted to assess the performance of the proposed method and showed advantage over the existing methods. For instance, the control mass $\mathcal{L}_2$ norm is reduced by 77.5% over the nonlinear control and 35% over the conventional LPV control.

6.2 Future work

This section proposes some viable future research directions depending on the conclusions of the work.

It is tempting to extend the idea of scaling scheduling parameter introduced in chapter three which is applied for affine LPV system with hard constraints to cover other types of LPV system such as Quasi-LPV. Moreover, it can be applied to other system examples with more hard constraints. Further, it also can be utilized with other damping functions than the proposed in chapter three.

The promising virtual control idea in chapter four (which is applied to LTI beam and ball system) and chapter five (which is applied to LPV smart airfoil flutter suppression system) can be applied to control examples other than beam and ball system and smart airfoil system. Furthermore, it can be applied with other control methods such as adaptive control and nonlinear control with many different virtual functions.

Building experimental rig is interesting idea to show the proposed methods in beam and ball system and smart airfoil flutter suppression system or any other examples.

APPENDICES

APPENDIX A

SOME LMI PRELIMINARIES

A.1 Hurwitz Stability

In this section some LMI definitions and conditions will be presented. Most of formulation can be found in [108].

Definition 1 [108]: A polytope $P \subset R^n$is a set which is the convex hull of a nonempty finite set $z_1, z_2, \ldots z_m \subset R^n$, that is $P = conv z_1, z_2, \ldots z_m$.

Proposition 1 [108]: A continuous system

$$\dot{x}(t) = Ax(t) \tag{A.1}$$

is Hurwitz stable if and only if there exists a matrix $P \in S^n$, such that

$$\begin{cases} P > 0 \\ A^T P + PA < 0 \end{cases} \tag{A.2}$$

A.1.1 Family of systems

Consider the following system

$$\dot{x}(t) = A(\theta(t))x(t) \tag{A.3}$$

where the system coefficient matrix takes the form of

$$A(\theta(t)) = A_0 + \sum_{i=1}^{q} \theta(t)_{(i,...q)} A_{(i,...q)} \tag{A.4}$$

For $q = 3$

$$A(\theta(t)) = A_0 + \theta(t)A_1 + \theta(t)A_2 + \theta(t)A_3 \tag{A.5}$$

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{A.6}$$

where $x \in \mathcal{R}^n$ is the state vector,$y \in \mathcal{R}^m$ is the output vector and $x \in \mathcal{R}^r$ is the input vector. A,B,C, and D represents the system matrices with appropriate dimensions.

Definition 2 [108]: The system A.6 is said to be Hurwitz stabilizable if there exists a real matrix K such that $A + BK$ is Hurwitz stable.

Theorem 2 PBH [108] : The system A.6 is Hurwitz stabilizable if and only if

$$rank[sI - A \qquad B] = n, \qquad \forall s \in C, \qquad Re(s) \geq 0;$$

$$rank[sI - A \qquad B] = n, \qquad \forall s \in \lambda(A), \qquad Re(s) \geq 0$$

thats mean the system is Hurwitz stable if all its uncontrollable modes are Hurwitz stable.

Theorem 3 [108] : The system A.6, is Hurwitz stabilizable if and only if there exist a symmetric positive definite matrix P and a matrix W satisfying

$$AP + PA^T + BW + W^T B^T < 0 \tag{A.7}$$

Definition 3 : The system matrix is called bounded real if it satisfies

$$G^H(s)G(s) \leq I, \qquad \forall s \in C, \qquad Re(s) \geq \tag{A.8}$$

Preposition 3 : The bounded real of definition 3 is equivalent to

$$\|G(s)\|_\infty \leq I \tag{A.9}$$

Theorem 3 Bounded-Real Lemma: The system A.6 be controllable then it is non-expansive if and only if its transfer function is bounded real.

Theorem 4 H_∞ LMI conditions: For a system A.6, then $\|G(s)\|_\infty < \gamma$ if and only if there exists a matrix $P > 0$ such that the following inequality holds

$$\begin{bmatrix} A^T P + PA & PB & C^T \\ B^T P & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0 \tag{A.10}$$

APPENDIX B

SCHEDULING PARAMETER MATRICES

B.1 Scheduling Parameter Matrices

In this section, the constant matrices (Z_0, G_0) and (Z_i, G_i) for constructing LPV-0, LPV-1, LPV-2 and LPV-2A controllers of chapter three are presented.

For LPV-0 controller: (Z_0^0, G_0^0) and (Z_i^0, G_i^0)

$$
\begin{aligned}
Z_0^0 &= [-0.051788000,\ \ 0.2853700, -0.261390,\ \ 0.476160,\ \ 0.045888, -8.81260],\\
Z_2^0 &= [-0.000028579,\ \ 0.0095015, -0.036397,\ \ 0.083472, -0.018931, -0.63590],\\
Z_3^0 &= [\ \ 0.008960500, -0.1393600,\ \ 0.006209, -0.042522, -0.117880,\ \ 0.31722]\,.
\end{aligned}
\tag{B.1}
$$

$$
G_0^0 = \begin{bmatrix}
1.0292e-05 & -7.0926e-06 & -2.0329e-07 & 3.6702e-06 & 7.1505e-09 & -9.4204e-05\\
-7.032e-06 & 3.6004e-05 & -2.4069e-06 & -2.6686e-06 & -4.575e-06 & 3.9664e-04\\
-4.1334e-07 & -2.1419e-06 & 1.0487e-05 & -1.3243e-05 & 9.1849e-05 & 4.0045e-05\\
3.9355e-06 & -3.5161e-06 & -1.3181e-05 & 4.2533e-05 & -3.7746e-04 & -1.4701e-04\\
-1.6558e-06 & 2.3264e-06 & 9.6502e-05 & -3.961e-04 & 4.9923e-02 & -3.2631e-01\\
-9.4264e-05 & 3.9815e-04 & 1.1691e-05 & -3.0011e-05 & -3.288e-01 & 104.92e-01
\end{bmatrix}
\tag{B.2}
$$

$$
G_2^0 = \begin{bmatrix}
1.0241e-05 & -6.8766e-06 & -1.9139e-07 & 3.642e-06 & -7.7403e-08 & -9.4036e-05\\
-6.8168e-06 & 3.54e-05 & -2.4457e-06 & -2.648e-06 & -3.9416e-07 & 3.8971e-04\\
-4.0415e-07 & -2.17e-06 & 1.063e-05 & -1.3519e-05 & 9.2068e-05 & 2.5043e-05\\
3.9103e-06 & -3.5054e-06 & -1.3456e-05 & 4.2983e-05 & -3.792e-04 & -9.1409e-05\\
-2.0406e-06 & 7.7154e-06 & 9.6801e-05 & -3.982e-04 & 5.1528e-02 & -2.3771e-01\\
-9.4179e-05 & 3.9065e-04 & 4.2126e-06 & -5.6076e-06 & -2.52e-01 & 29.561e-01
\end{bmatrix}
\tag{B.3}
$$

$$G_3^0 = \begin{bmatrix} 1.0241e-05 & -6.8766e-06 & -1.9139e-07 & 3.6419e-06 & 3.0065e-08 & -9.4368e-05 \\ -6.8168e-06 & 3.54e-05 & -2.4457e-06 & -2.648e-06 & -8.4001e-07 & 3.9108e-04 \\ -4.0415e-07 & -2.17e-06 & 1.063e-05 & -1.3519e-05 & 9.2005e-05 & 2.5621e-05 \\ 3.9103e-06 & -3.5054e-06 & -1.3456e-05 & 4.2983e-05 & -3.7894e-04 & -9.4228e-05 \\ -2.0393e-06 & 7.7183e-06 & 9.681e-05 & -3.9823e-04 & 5.2281e-02 & -2.4613e-01 \\ -9.4195e-05 & 3.9049e-04 & 3.9802e-06 & -4.9266e-06 & -2.6559e-01 & 30.102e-01 \end{bmatrix} \tag{B.4}$$

For LPV-1 controller: (Z_0^1, G_0^1) and (Z_i^1, G_i^1)

$$\begin{aligned} Z_0^1 &= [\ 0.548420, -0.00019372, -0.15522000, \ 0.71419000, \ 1.08060000, -109.9000], \\ Z_1^1 &= [\ 0.553410, -0.00019562, -0.13229000, \ 0.70671000, \ 1.08980000, -110.8800], \\ Z_2^1 &= [-0.014035, -0.03191200, \ 0.08397600, -0.05061400, -0.00059914, -0.048978], \\ Z_3^1 &= [-0.287940, -0.00022911, \ 0.00022877, -0.00082886, \ 0.02613200, -0.121390]\,. \end{aligned} \tag{B.5}$$

$$G_0^1 = \begin{bmatrix} 7.8896e-06 & -4.88e-06 & -2.1003e-07 & 3.0824e-06 & -4.144e-05 & -3.9649e-05 \\ -4.8329e-06 & 2.7367e-05 & -1.6313e-06 & -3.1912e-06 & 1.3371e-04 & 2.3919e-04 \\ -3.6471e-07 & -1.4518e-06 & 7.9214e-06 & -9.6416e-06 & 8.1299e-05 & -5.8287e-05 \\ 3.2593e-06 & -3.784e-06 & -9.5948e-06 & 3.2516e-05 & -3.6529e-04 & 7.8805e-05 \\ -4.2384e-05 & 1.3887e-04 & 8.1301e-05 & -3.6761e-04 & 1.8855e-02 & -5.0645e-01 \\ -2.8195e-05 & 1.7853e-04 & -5.6427e-05 & 1.6936e-04 & -5.1254e-01 & 507.95e-01 \end{bmatrix} \tag{B.6}$$

$$G_1^1 = \begin{bmatrix} 7.8884e-06 & -4.8753e-06 & -2.0747e-07 & 3.0788e-06 & -4.1397e-05 & -3.9695e-05 \\ -4.8288e-06 & 2.7344e-05 & -1.6295e-06 & -3.1935e-06 & 1.3359e-04 & 2.3879e-04 \\ -3.6188e-07 & -1.4515e-06 & 7.9138e-06 & -9.6322e-06 & 8.1284e-05 & -5.8183e-05 \\ 3.2555e-06 & -3.7846e-06 & -9.5862e-06 & 3.2506e-05 & -3.6529e-04 & 7.8771e-05 \\ -4.2359e-05 & 1.388e-04 & 8.1281e-05 & -3.6761e-04 & 1.8855e-02 & -5.0643e-01 \\ -2.8291e-05 & 1.7829e-04 & -5.63e-05 & 1.6924e-04 & -5.1253e-01 & 507.95e-01 \end{bmatrix} \tag{B.7}$$

$$G_2^1 = \begin{bmatrix} 7.8555e-06 & -4.7094e-06 & -2.0663e-07 & 3.0568e-06 & -4.0963e-05 & -2.6925e-06 \\ -4.6633e-06 & 2.6788e-05 & -1.6547e-06 & -3.1241e-06 & 1.3399e-04 & 8.7094e-05 \\ -3.6405e-07 & -1.4649e-06 & 8.0632e-06 & -9.8762e-06 & 8.1091e-05 & -6.0543e-05 \\ 3.2368e-06 & -3.726e-06 & -9.8281e-06 & 3.2861e-05 & -3.6601e-04 & 1.9392e-04 \\ -4.1803e-05 & 1.3851e-04 & 8.111e-05 & -3.6749e-04 & 1.398e-02 & -1.305e-02 \\ -2.1645e-06 & 8.5489e-05 & -6.2007e-05 & 1.9973e-04 & -1.3281e-02 & 2.3088e-02 \end{bmatrix} \tag{B.8}$$

$$G_3^1 = \begin{bmatrix} 7.8543e-06 & -4.7047e-06 & -2.0406e-07 & 3.0532e-06 & -4.092e-05 & -2.7393e-06 \\ -4.6591e-06 & 2.6765e-05 & -1.6529e-06 & -3.1264e-06 & 1.3387e-04 & 8.6695e-05 \\ -3.6122e-07 & -1.4646e-06 & 8.0557e-06 & -9.8668e-06 & 8.1075e-05 & -6.0439e-05 \\ 3.2329e-06 & -3.7267e-06 & -9.8195e-06 & 3.2851e-05 & -3.66e-04 & 1.9388e-04 \\ -4.1778e-05 & 1.3845e-04 & 8.109e-05 & -3.6748e-04 & 1.398e-02 & -1.3033e-02 \\ -2.2601e-06 & 8.525e-05 & -6.188e-05 & 1.9961e-04 & -1.3264e-02 & 2.305e-02 \end{bmatrix} \tag{B.9}$$

For LPV-2 controller: (Z_0^2, G_0^2) and (Z_i^2, G_i^2)

$$Z_0^2 = [-2.4213e-06,\ \ 4.2664e-06, 6.1342e-06, -3.2752e-05, -1.9581e-02, -5.5480e-01],$$
$$Z_1^2 = [-5.9946e-06,\ \ 2.3481e-05, 4.3778e-05, -1.8995e-04, -5.1250e-02, -4.0719e-01],$$
$$Z_2^2 = [\ \ 2.2156e-06, -2.9162e-05, 1.1126e-04, -3.9439e-04, -4.1207e-02, -5.2469e-01],$$
$$Z_3^2 = [-2.7218e-06, -8.0204e-06, 2.2373e-04, -8.4153e-04, -8.6814e-02, -6.8411e-01]\,.$$

(B.10)

$$G_0^2 = \begin{bmatrix} 8.6375e-06 & -5.8541e-06 & -1.7052e-07 & 3.061e-06 & 1.9876e-06 & -8.5609e-05 \\ -5.8037e-06 & 2.9948e-05 & -1.9898e-06 & -2.3014e-06 & -1.1892e-05 & 3.8407e-04 \\ -3.4681e-07 & -1.7695e-06 & 8.7657e-06 & -1.0947e-05 & 8.0774e-05 & 1.3285e-04 \\ 3.2827e-06 & -3.0092e-06 & -1.0893e-05 & 3.5244e-05 & -3.3881e-04 & -4.2294e-04 \\ 3.2859e-07 & -4.7744e-06 & 8.6082e-05 & -3.6101e-04 & 5.3732e-02 & -4.4081e-01 \\ -8.6032e-05 & 3.9208e-04 & 2.2815e-04 & -8.0485e-04 & -3.9729e-01 & 71.034e-01 \end{bmatrix}$$

(B.11)

$$G_1^2 = \begin{bmatrix} 8.6356e-06 & -5.8486e-06 & -1.6711e-07 & 3.0549e-06 & 3.458e-06 & -9.3032e-05 \\ -5.7989e-06 & 2.9927e-05 & -1.989e-06 & -2.3008e-06 & -1.6895e-05 & 4.168e-04 \\ -3.4266e-07 & -1.7703e-06 & 8.7587e-06 & -1.0942e-05 & 8.0278e-05 & 2.285e-04 \\ 3.2751e-06 & -3.0071e-06 & -1.0891e-05 & 3.5264e-05 & -3.3025e-04 & -9.0546e-04 \\ 2.0057e-06 & -9.9109e-06 & 8.5962e-05 & -3.5333e-04 & 5.539e-02 & -3.4871e-01 \\ -9.6388e-05 & 4.2388e-04 & 1.476e-04 & -5.6966e-04 & -3.7782e-01 & 67.63e-01 \end{bmatrix}$$

(B.12)

$$G_2^2 = \begin{bmatrix} 8.6022e-06 & -5.6961e-06 & -1.4859e-07 & 2.9851e-06 & 3.8124e-07 & -7.9024e-05 \\ -5.6459e-06 & 2.951e-05 & -2.0785e-06 & -2.0486e-06 & 8.2653e-07 & 3.3505e-04 \\ -3.2917e-07 & -1.8423e-06 & 8.9527e-06 & -1.1358e-05 & 6.6592e-05 & 8.7906e-05 \\ 3.2152e-06 & -2.7843e-06 & -1.13e-05 & 3.6103e-05 & -2.8264e-04 & -3.1086e-04 \\ -1.4043e-06 & 7.7799e-06 & 7.1544e-05 & -3.0434e-04 & 2.7715e-02 & -1.6798e-01 \\ -7.7589e-05 & 3.3782e-04 & 1.6989e-04 & -6.2933e-04 & -1.3177e-01 & 30.6e-01 \end{bmatrix}$$

(B.13)

$$G_3^2 = \begin{bmatrix} 8.6002e-06 & -5.6906e-06 & -1.4519e-07 & 2.9789e-06 & 1.8516e-06 & -8.6448e-05 \\ -5.6411e-06 & 2.9489e-05 & -2.0777e-06 & -2.048e-06 & -4.1759e-06 & 3.6778e-04 \\ -3.2503e-07 & -1.8431e-06 & 8.9457e-06 & -1.1354e-05 & 6.6096e-05 & 1.8355e-04 \\ 3.2076e-06 & -2.7822e-06 & -1.1298e-05 & 3.6122e-05 & -2.7409e-04 & -7.9338e-04 \\ 2.7281e-07 & 2.6434e-06 & 7.1424e-05 & -2.9666e-04 & 2.9374e-02 & -7.5872e-02 \\ -8.7945e-05 & 3.6962e-04 & 8.9333e-05 & -3.9413e-04 & -1.123e-01 & 27.196e-01 \end{bmatrix} \tag{B.14}$$

For LPV-2A controller: (Z_0^{2A}, G_0^{2A}) and (Z_i^{2A}, G_i^{2A})

$$\begin{aligned} Z_0^{2A} &= [9.5650e-05, 7.7015e-05, -1.1704e-04, \quad 1.4560e-04, -1.3835e-03, -28.959e-01], \\ Z_1^{2A} &= [5.2380e-05, 9.1385e-06, \quad 1.0587e-04, -5.9202e-04, -5.5273e-02, -3.0144e-01], \\ Z_2^{2A} &= [1.6166e-05, 4.9277e-05, \quad 1.2832e-04, -7.2469e-04, -1.9366e-01, \quad 1.6593e-01]. \end{aligned} \tag{B.15}$$

$$G_0^{2A} = \begin{bmatrix} 1.2067e-05 & -5.7913e-06 & -1.4604e-07 & 2.7411e-06 & 6.1678e-07 & -8.6683e-05 \\ -5.7468e-06 & 3.6872e-05 & -1.7063e-06 & -2.5482e-06 & -3.6149e-06 & 3.4392e-04 \\ -3.8476e-07 & -1.4536e-06 & 1.2038e-05 & -1.3235e-05 & 7.9318e-05 & 1.2268e-04 \\ 3.0024e-06 & -3.4095e-06 & -1.3189e-05 & 4.405e-05 & -3.2221e-04 & -4.8872e-04 \\ -1.1886e-06 & 3.4651e-06 & 8.7408e-05 & -3.5454e-04 & 4.0479e-02 & -2.4278e-01 \\ -8.9113e-05 & 3.5178e-04 & 1.2295e-04 & -4.9701e-04 & -2.2854e-01 & 43.08e-01 \end{bmatrix} \tag{B.16}$$

$$G_1^{2A} = \begin{bmatrix} 1.206e-05 & -5.7651e-06 & -1.3122e-07 & 2.7185e-06 & 7.9486e-07 & -9.5248e-05 \\ -5.7205e-06 & 3.6796e-05 & -1.7858e-06 & -2.4327e-06 & -2.4595e-06 & 3.912e-04 \\ -3.7252e-07 & -1.5181e-06 & 1.2054e-05 & -1.3257e-05 & 7.6751e-05 & 1.5131e-04 \\ 2.9831e-06 & -3.3113e-06 & -1.3209e-05 & 4.405e-05 & -3.1524e-04 & -6.3172e-04 \\ -7.6883e-07 & 4.1088e-06 & 8.0012e-05 & -3.2804e-04 & 4.1129e-02 & -1.6123e-01 \\ -9.0988e-05 & 3.7241e-04 & 9.6802e-05 & -3.966e-04 & -1.7717e-01 & 37.591e-01 \end{bmatrix} \tag{B.17}$$

$$G_2^{2A} = \begin{bmatrix} 1.2067e-05 & -5.7914e-06 & -1.4607e-07 & 2.7412e-06 & 6.7547e-07 & -8.7462e-05 \\ -5.7468e-06 & 3.6872e-05 & -1.7061e-06 & -2.5486e-06 & -3.9118e-06 & 3.4587e-04 \\ -3.8481e-07 & -1.4533e-06 & 1.2038e-05 & -1.3235e-05 & 7.9304e-05 & 1.1497e-04 \\ 3.0026e-06 & -3.4104e-06 & -1.3189e-05 & 4.405e-05 & -3.2238e-04 & -4.5687e-04 \\ -1.183e-06 & 3.3699e-06 & 8.7379e-05 & -3.5453e-04 & 4.0815e-02 & -2.4882e-01 \\ -8.8986e-05 & 3.5128e-04 & 1.2343e-04 & -4.9817e-04 & -2.2927e-01 & 43.775e-01 \end{bmatrix} \tag{B.18}$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Ali M Alhajjar, Ali Al-jiboory, Sean Shan-Min Swei, and Guoming G Zhu. Lpv modeling and control for active flutter suppression of a smart airfoil. In *AIAA Guidance, Navigation, and Control Conference*, pages 2018–1342, 2018.

[2] Sean Shan-Min Swei and Yi-Tsann Jiang. On the efficacy of the smart airfoil model in active flutter suppression. In *Second international Conference on Motion and Vibration Control*, pages 593–598. Yokohama, 1994.

[3] Ali MH Al-Hajjar, Sean Shan-Min Swei, and Guoming Zhu. Novel linear parameter–varying modeling and flutter suppression control of a smart airfoil. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, page 0959651818804377, 2018.

[4] José A De Doná, Graham C Goodwin, and SO Reza Moheimani. Combining switching, over-saturation and scaling to optimise control performance in the presence of model uncertainty and input saturation. *Automatica*, 38(7):1153–1162, 2002.

[5] Yonggang Chen, Shumin Fei, Kanjian Zhang, and Lei Yu. Control of switched linear systems with actuator saturation and its applications. *Mathematical and Computer Modelling*, 56(1-2):14–26, 2012.

[6] Chengzhi Yuan and Fen Wu. A switching control approach for linear systems subject to asymmetric actuator saturation. In *33rd Chinese Control Conference (CCC).*, pages 3959–3964. IEEE, 2014.

[7] Chengzhi Yuan and Fen Wu. Switching control of linear systems subject to asymmetric actuator saturation. *International Journal of Control*, 88(1):204–215, 2015.

[8] Kenneth R Muske and James B Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993.

[9] Christopher V Rao and James B Rawlings. Steady states and constraints in model predictive control. *AIChE Journal*, 45(6):1266–1278, 1999.

[10] Jan Marian Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.

[11] Johan Löfberg. Yalmip: A toolbox for modeling and optimization in matlab. In *International Symposium on Computer Aided Control Systems Design*, pages 284–289. IEEE, 2004.

[12] Ali Khudhair Al-Jiboory. *Guaranteed performance robust gain-scheduling control with uncertain scheduling parameters*. Michigan State University, 2016.

[13] Ricardo CLF Oliveira, Pierre-Alexandre Bliman, and Pedro LD Peres. Robust lmis with parameters in multi-simplex: Existence of solutions and applications. In *47th Conference on Decision and Control, CDC .*, pages 2226–2231. IEEE, 2008.

[14] Ricardo CLF Oliveira, Maurício C De Oliveira, and Pedro LD Peres. Robust state feedback lmi methods for continuous-time linear systems: Discussions, extensions and numerical comparisons. In *International Symposium on Computer-Aided Control System Design (CACSD).*, pages 1038–1043. IEEE, 2011.

[15] Ricardo CLF Oliveira, Pierre-Alexandre Bliman, and Pedro LD Peres. Selective gain-scheduling for continuous-time linear systems with parameters in multi-simplex. In *European Control Conference (ECC).*, pages 213–218. IEEE, 2009.

[16] D Subbaram Naidu. *Optimal control systems*. CRC press, 2002.

[17] L Chesta. A parametric study of wing store flutter. In *AGARD Specialists Meeting on Wing-With-Stores Flutter 12 p(SEE N 75-28011 19-02),*, 1975.

[18] Mordechay Karpel. *Design for active and passive flutter suppression and gust alleviation*, volume 3482. National Aeronautics and Space Administration, Scientific and Technical Information Branch, 1981.

[19] Wilmer H Reed III, FW Cazier Jr, and Jerome T Foughner Jr. Passive control of wing/store flutter. *National Aeronautics and Space Administration, Technical Report*, 1980.

[20] Zhi-Chun Yang and Ling-Cheng Zhao. Wing-store flutter analysis of an airfoil in incompressible flow. *Journal of Aircraft*, 26(6):583–587, 1989.

[21] Jae-Hung Han, Junji Tani, and Jinhao Qiu. Active flutter suppression of a lifting surface using piezoelectric actuation and modern control theory. *Journal of Sound and Vibration*, 291(3):706–722, 2006.

[22] C De Marqui, EM Belo, and FD Marques. A flutter suppression active controller. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 219(1):19–33, 2005.

[23] K Zhang and A Behal. Continuous robust control for aeroelastic vibration control of a 2-d airfoil under unsteady flow. *Journal of Vibration and Control*, 22(12):2841–2860, 2016.

[24] William E Triplett, Hans-Peter F Kappus, and Robert J Landy. Active flutter suppression systems for military aircraft. a feasibility study. Technical report, DTIC Document, 1973.

[25] Christina Rubillo, Piergiovanni Marzocca, and Erik Bollt. Active aeroelastic control of lifting surfaces via jet reaction limiter control. *International Journal of Bifurcation and Chaos*, 16(09):2559–2574, 2006.

[26] Hak-Tae Lee, Ilan Kroo, and Stefan Bieniawski. Flutter suppression for high aspect ratio flexible wings using microflaps. In *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference,*, page 1717, 2002.

[27] Jan-Willem Van Wingerden, Pieter Gebraad, and Michel Verhaegen. Lpv identification of an aeroelastic flutter model. In *49th Conference on Decision and Control (CDC),*, pages 6839–6844. IEEE, 2010.

[28] Jeffrey M Barker and Gary J Balas. Comparing linear parameter-varying gain-scheduled control techniques for active flutter suppression. *Journal of Guidance, Control, and Dynamics*, 23(5):948–955, 2000.

[29] E Lau and AJ Krener. Lpv control of two dimensional wing flutter. In *Proceedings of the 38th Conference on Decision and Control,*, volume 3, pages 3005–3010. IEEE, 1999.

[30] Gang Chen, Yueming Li, Sun Jian, Zuo Yingtao, and Patrick Hu. Linear parameter varying control for active flutter suppression based on adaptive reduced order model. In *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 19th AIAA/ASME/AHS Adaptive Structures Conference,*, page 1773, 2011.

[31] Gary Balas, Claudia Moreno, and Peter Seiler. Robust aeroservoelastic control utilizing physics-based aerodynamic sensing. In *AIAA Guidance, Navigation, and Control Conference,*, page 4897, 2012.

[32] Ali Khudhair Al-Jiboory, Guoming G Zhu, Sean Shan-Min Swei, Weihua Su, and Nhan T Nguyen. Lpv modeling of a flexible wing aircraft using modal alignment and adaptive gridding methods. *Aerospace Science and Technology*, 66:92–102, 2017.

[33] Andrew P White, Guoming G Zhu, and Jongeun Choi. *Linear parameter-varying control for engineering applications*. Springer, 2013.

[34] Xiaobao Han, Zhenbao Liu, Huacong Li, and Xianwei Liu. Output feedback controller design for polynomial linear parameter varying system via parameter-dependent lyapunov functions. *Advances in Mechanical Engineering*, 9(2):1687814017690327, 2017.

[35] Damiano Rotondo, Fatiha Nejjari, and Vicenç Puig. Quasi-lpv modeling, identification and control of a twin rotor mimo system. *Control Engineering Practice*, 21(6):829–846, 2013.

[36] Ilhan Polat, Esref Eskinat, and IE Kose. Dynamic output feedback control of quasi-lpv mechanical systems. *IET Control Theory & Applications*, 1(4):1114–1121, 2007.

[37] Fatiha Nejjari, Damiano Rotondo, Vicenç Puig, and Mario Innocenti. Quasi-lpv modelling and non-linear identification of a twin rotor system. In *20th Mediterranean Conference on Control & Automation (MED),*, pages 229–234. IEEE, 2012.

[38] Mordechay Karpel. Design for active flutter suppression and gust alleviation using state-space aeroelastic modeling. *Journal of Aircraft*, 19(3):221–227, 1982.

[39] Andreas Kwiatkowski and Herbert Werner. Pca-based parameter set mappings for lpv models with fewer parameters and less overbounding. *IEEE Transactions on Control Systems Technology*, 16(4):781–788, 2008.

[40] Márcio J Lacerda, Eduardo S Tognetti, Ricardo CLF Oliveira, and Pedro LD Peres. A new approach to handle additive and multiplicative uncertainties in the measurement for lpv filtering. *International Journal of Systems Science*, (ahead-of-print):1042–1053, 2016.

[41] Carsten W Scherer. Lmi relaxations in robust control. *European Journal of Control*, 12(1):3–29, 2006.

[42] Ricardo CLF Oliveira and Pedro LD Peres. Parameter-dependent lmis in robust analysis: characterization of homogeneous polynomially parameter-dependent solutions via lmi relaxations. *Transactions on Automatic Control.*, 52(7):1334–1340, 2007.

[43] Carsten W Scherer and Camile WJ Hol. Matrix sum-of-squares relaxations for robust semi-definite programs. *Mathematical programming*, 107(1-2):189–211, 2006.

[44] Dimitri Peaucelle and Masayuki Sato. Lmi tests for positive definite polynomials: Slack variable approach. *IEEE Transactions on Automatic Control*, 54(4):886–891, 2009.

[45] Cristiano M Agulhari, RCLF de Oliveira, and Pedro LD Peres. Robust lmi parser: a computational package to construct lmi conditions for uncertain systems. In *XIX Brazilian Conference on Automation (CBA)*, pages 2298–2305, 2012.

[46] Jos F Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.

[47] Pierre Apkarian and Hoang Duong Tuan. Parameterized lmis in control theory. *SIAM journal on control and optimization*, 38(4):1241–1264, 2000.

[48] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

[49] Pablo A Parrilo and Sanjay Lall. Semidefinite programming relaxations and algebraic optimization in control. *European Journal of Control*, 9(2-3):307–321, 2003.

[50] Jean B Lasserre. A semidefinite programming approach to the generalized problem of moments. *Mathematical Programming*, 112(1):65–92, 2008.

[51] Fen Wu, Xin Hua Yang, Andy Packard, and Greg Becker. Induced l2-norm control for lpv systems with bounded parameter variation rates. *International Journal of Robust and Nonlinear Control*, 6(9-10):983–998, 1996.

[52] Greg Becker and Andy Packard. Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback. *Systems & Control Letters*, 23(3):205–215, 1994.

[53] Jie Yu and Athanasios Sideris. HâĹđ control with parametric lyapunov functions. *Systems & Control Letters*, 30(2-3):57–69, 1997.

[54] Yoshio Ebihara and Tomomichi Hagiwara. A dilated lmi approach to robust performance analysis of linear time-invariant uncertain systems. *Automatica*, 41(11):1933–1941, 2005.

[55] Pascal Gahinet, Pierre Apkarian, and Mahmoud Chilali. Affine parameter-dependent lyapunov functions and real parametric uncertainty. *IEEE Transactions on Automatic control*, 41(3):436–442, 1996.

[56] Guoming G Zhu, Karolos M Grigoriadis, and Robert E Skelton. Covariance control design for hubble space telescope. *Journal of Guidance, Control, and Dynamics*, 18(2):230–236, 1995.

[57] Armando A Rodriguez and James R Cloutier. Control of a bank-to-turn (btt) missile with saturating actuators. In *American Control Conference.*, volume 2, pages 1660–1664. IEEE, 1994.

[58] Yoram Koren. Control of machine tools. *Journal of Manufacturing Science and Engineering*, 119(4B):749–755, 1997.

[59] Dennis S Bernstein and Anthony N Michel. A chronological bibliography on saturating actuators. *International Journal of robust and nonlinear control*, 5(5):375–380, 1995.

[60] Vikram Kapila and Karolos Grigoriadis. *Actuator saturation control*. CRC Press, 2002.

[61] J Lozier. A steady state approach to the theory of saturable servo systems. *IRE Transactions on Automatic Control*, 1(1):19–39, 1956.

[62] John C Doyle, Roy S Smith, and Dale F Enns. Control of plants with input saturation nonlinearities. In *American Control Conference.*, pages 1034–1039. IEEE, 1987.

[63] Hongchao Li, Zhiqiang Zuo, and Yijing Wang. Dynamic output feedback control for systems subject to actuator saturation via event-triggered scheme. *Asian Journal of Control*, 20(1):207–215, 2018.

[64] Elmer G Gilbert and K Tin Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic control*, 36(9):1008–1020, 1991.

[65] Eric N Johnson and Anthony J Calise. Pseudo-control hedging: A new method for adaptive control. In *Advances in navigation guidance and control technology workshop*. Alabama, USA Alabama, USA, 2000.

[66] Xinquan Zhang, Jun Zhao, and Xiaoyin Li. Stability analysis and design of uncertain discrete-time switched systems with actuator saturation using antiwindup and multiple lyapunov functions approach. *Asian Journal of Control*, 19(1):325–331, 2017.

[67] Xinquan Zhang. L 2-gain analysis and anti-windup design of switched linear systems subject to input saturation. *Asian Journal of Control*, 19(2):672–680, 2017.

[68] Han Li and Airong Wei. Stabilization and hâĹđ control of nonlinear switched hamiltonian systems subject to actuator saturation. *Asian Journal of Control*, 19(3):951–960, 2017.

[69] Robert D Gregg, Tommaso Lenzi, Levi J Hargrove, and Jonathon W Sensinger. Virtual constraint control of a powered prosthetic leg: From simulation to experiments with transfemoral amputees. *IEEE Transactions on Robotics*, 30(6):1455–1471, 2014.

[70] J Anthony Rossiter. *Model-based predictive control: a practical approach*. CRC press, 2003.

[71] GC Goodwin, MM Serón, and JA De Doná. Constrained control & estimation–an optimization perspective. *London: SpringerVerlag*, 2005.

[72] Wook Hyun Kwon and Soo Hee Han. *Receding horizon control: model predictive control for state models*. Springer Science & Business Media, 2006.

[73] Liuping Wang. *Model predictive control system design and implementation using MATLAB*. Springer Science & Business Media, 2009.

[74] Langwen Zhang, Wei Xie, and Jingcheng Wang. Robust mpc for linear systems with structured time-varying uncertainties and saturating actuator. *Asian Journal of Control*, 19(3):1197–1204, 2017.

[75] Ridvan Berber and Costas Kravaris. *Nonlinear model based process control*, volume 353. Springer Science & Business Media, 2012.

[76] James B Rawlings and David Q Mayne. Model predictive control: Theory and design. 2009.

[77] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[78] Jianhua Wang, Yan Song, Sunjie Zhang, Shuai Liu, and Abdullah M Dobaie. Robust model predictive control for linear discrete-time system with saturated inputs and randomly occurring uncertainties. *Asian Journal of Control*, 20(1):425–436, 2018.

[79] Haofei Xie, Jun Wang, and Xiaoming Tang. Robust constrained model predictive control for discrete-time uncertain system in takagi-sugeno's form. *Asian Journal of Control*, 20(4):1566–1581, 2018.

[80] Priyank Jain and MJ Nigam. Real time control of ball and beam system with model reference adaptive control strategy using mit rule. In *International Conference on Computational Intelligence and Computing Research (ICCIC).*, pages 1–4. IEEE, 2013.

[81] MK Choudhary and G Naresh Kumar. Eso based lqr controller for ball and beam system. *IFAC-PapersOnLine*, 49(1):607–610, 2016.

[82] Stephen Pace and Guoming G Zhu. Optimal lq transient air-to-fuel ratio control of an internal combustion engine. In *Dynamic Systems and Control Conference*, pages 763–768. American Society of Mechanical Engineers, 2011.

[83] Fen Wu, Karolos M Grigoriadis, and Andy Packard. Anti-windup controller design using linear parameter-varying control methods. *International Journal of Control*, 73(12):1104–1114, 2000.

[84] Fernando A Inthamoussou, Fernando D Bianchi, Hernán De Battista, and Ricardo J Mantz. Lpv wind turbine control with anti-windup features covering the complete wind speed range. *IEEE Transactions on Energy Conversion*, 29(1):259–266, 2014.

[85] Sophie Tarbouriech and Matthew Turner. Anti-windup design: an overview of some recent advances and open problems. *IET control theory & applications*, 3(1):1–19, 2009.

[86] FEN Wu and Karlos M Grigoriadis. Lpv-based control of systems with amplitude and rate actuator saturation constraints. In *Proceedings of the American Control Conference,*, volume 5, pages 3191–3195. IEEE, 1999.

[87] Anh-Lam Do, Olivier Sename, and Luc Dugard. An lpv control approach for semi-active suspension control with actuator constraints. In *American Control Conference (ACC),*, pages 4653–4658. IEEE, 2010.

[88] Andrew White, Guoming Zhu, and Jongeun Choi. Optimal lpv control with hard constraints. *International Journal of Control, Automation and Systems*, 14(1):148–162, 2016.

[89] Manh Quan Nguyen, Olivier Sename, and Luc Dugard. An lpv fault tolerant control for semi-active suspension-scheduled by fault estimation. *IFAC-PapersOnLine*, 48(21):42–47, 2015.

[90] Tianyi He, Ali Khudhair Al-Jiboory, Sean Shan-Min Swei, and Guoming G Zhu. Switching state-feedback lpv control with uncertain scheduling parameters. In *American Control Conference (ACC),*, pages 2381–2386. IEEE, 2017.

[91] Tianyi He, Guoming G Zhu, Sean Shan-Min Swei, and Weihua Su. Simultaneous design of smooth switching state-feedback lpv control. In *Annual American Control Conference (ACC)*, pages 3368–3373. IEEE, 2018.

[92] Tianyi He, Guoming G Zhu, Sean Shan-Min Swei, and Weihua Su. Smooth-switching lpv control for vibration suppression of a flexible airplane wing. *Aerospace Science and Technology*, 84:895–903, 2019.

[93] Damiano Rotondo. *Advances in gain-scheduling and fault tolerant control techniques*. Springer, 2017.

[94] Damiano Rotondo, Jean-Christophe Ponsart, Fatiha Nejjari, Didier Theilliol, and Vicenç Puig. Virtual actuator-based ftc for lpv systems with saturating actuators and fdi delays. Institute of Electrical and Electronics Engineers, 2016.

[95] Damiano Rotondo, Jean-Christophe Ponsart, Didier Theilliol, Fatiha Nejjari, and Vicenç Puig. A virtual actuator approach for the fault tolerant control of unstable linear systems subject to actuator saturation and fault isolation delay. *Annual Reviews in Control*, 39:68–80, 2015.

[96] María M Seron, José A De Doná, and Jan Richter. Fault tolerant control using virtual actuators and set-separation detection principles. *International Journal of Robust and Nonlinear Control*, 22(7):709–742, 2012.

[97] Seyed Mojtaba Tabatabaeipour, Jakob Stoustrup, and Thomas Bak. Control reconfiguration of lpv systems using virtual sensor and actuator. *IFAC Proceedings Volumes*, 45(20):818–823, 2012.

[98] Saúl de Oca, Vicenç Puig, Marcin Witczak, and Łukasz Dziekan. Fault-tolerant control strategy for actuator faults using lpv techniques: Application to a two degree of freedom helicopter. *International Journal of Applied Mathematics and Computer Science*, 22(1):161–171, 2012.

[99] Toly Chen, Yi-Chi Wang, and Zhirong Lin. Predictive distant operation and virtual control of computer numerical control machines. *Journal of Intelligent Manufacturing*, 28(5):1061–1077, 2017.

[100] Bradley Reaves and Thomas Morris. An open virtual testbed for industrial control system security research. *International Journal of Information Security*, 11(4):215–229, 2012.

[101] Larry Pratt and Larry Armi. Two-layer rotating hydraulics: strangulation, remote and virtual controls. *pure and applied geophysics*, 133(4):587–617, 1990.

[102] Dan Wang and Jie Huang. Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form. *IEEE Transactions on Neural Networks*, 16(1):195–202, 2005.

[103] Wen-Hong Zhu. *Virtual decomposition control: toward hyper degrees of freedom robots*, volume 60. Springer Science & Business Media, 2010.

[104] Sean Shan-Min Swei and Mohammad A Ayoubi. Lmi-based fuzzy optimal variance control of airfoil model subject to input constraints. In *International Conference on Fuzzy Systems*, pages 1–6. IEEE, 2017.

[105] Frank Allgöwer and Alex Zheng. *Nonlinear model predictive control*, volume 26. Birkhäuser, 2012.

[106] Corentin Briat. Linear parameter-varying and time-delay systems. *Analysis, observation, filtering & control*, 3, 2014.

[107] Masayuki Sato. Design method of gain-scheduled controllers not depending on derivatives of parameters. *International Journal of Control*, 81(6):1013–1025, 2008.

[108] Guang-Ren Duan and Hai-Hua Yu. *LMIs in control systems: analysis, design and applications*. CRC press, 2013.