ROBUST AND MULTI-OBJECTIVE MODEL PREDICTIVE CONTROL DESIGN FOR NONLINEAR SYSTEMS

Amir Hajiloo

A thesis
in
The Department
of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy
Concordia University
Montréal, Québec, Canada

February 2016

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Amir Hajiloo**

Entitled: **Robust and Multi-Objective Model Predictive Control Design for Nonlinear Systems**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

_____ Dr. Wei-Ping Zhu, Chair

_____ Dr. Zheng Hong Zhu, External Examiner

_____ Dr. Shahin Hashtrudi Zad, Examiner

_____ Dr. Chun-Yi Su, Examiner

_____ Dr. Javad Dargahi, Examiner

_____ Dr. Wen-Fang Xie, Supervisor

Approved _____
Chair of Department or Graduate Program Director

_____ 2016 _____ _____
Amir Asif, Ph.D., Interim Dean
Faculty of Engineering and Computer Science

Abstract

Robust and Multi-Objective Model Predictive Control Design for Nonlinear Systems

Amir Hajiloo, Ph.D.
Concordia University, 2016

The multi-objective trade-off paradigm has become a very valuable design tool in engineering problems that have conflicting objectives. Recently, many control designers have worked on the design methods which satisfy multiple design specifications called multi-objective control design. However,the main challenge posed for the MPC design lies in the high computation load preventing its application to the fast dynamic system control in real-time. To meet this challenge, this thesis has proposed several methods covering nonlinear system modeling, on-line MPC design and multi-objective optimization. First, the thesis has proposed a robust MPC to control the shimmy vibration of the landing gear with probabilistic uncertainty. Then, an on-line MPC method has been proposed for image-based visual servoing control of a 6 DOF Denso robot. Finally, a multi-objective MPC has been introduced to allow the designers consider multiple objectives in MPC design.

In this thesis, Tensor Product (TP) model transformation as a powerful tool in the modeling of the complex nonlinear systems is used to find the linear parameter-varying (LPV) models of the nonlinear systems. Higher-order singular value decomposition (HOSVD) technique is used to obtain a minimal order of the model tensor. Furthermore, to design a robust MPC for nonlinear systems in the presence of uncertainties which degrades the system performance and can lead to instability, we consider the parameters of the nonlinear systems with probabilistic uncertainties in the modeling using TP transformation. In this thesis, a computationally efficient methods for MPC design of image-based visual servoing, i.e. a fast dynamic system has been proposed. The controller is designed considering the robotic visual servoing system's input and output constraints, such as robot physical limitations and visibility constraints.

The main contributions of this thesis are: (i) design MPC for nonlinear systems with probabilistic uncertainties that guarantees robust stability and performance of the systems;

(ii) develop a real-time MPC method for a fast dynamical system; (iii) to propose a new multi-objective MPC for nonlinear systems using game theory. A diverse range of systems with nonlinearities and uncertainties including landing gear system, 6 DOF Denso robot are studied in this thesis. The simulation and real-time experimental results are presented and discussed in this thesis to verify the effectiveness of the proposed methods.

Acknowledgments

I would like to express my deepest appreciation and gratitude to my supervisor Dr. Wen-Fang Xie. Without her guidance, continuous support, and encouragement throughout my studies and research, this dissertation would have been impossible. For this, I am utmost grateful.

Also, I cannot be thankful enough to my family, friends and those who contributed to the successful completion of my studies at Concordia University for self-devotion, encouragement, and support in the past years. I appreciate the support of Concordia University that covered my financial needs during my studies.

I dedicate this thesis to the loving memory of my father, my best friend.
I also dedicate it to my mother for her constant, unconditional love and support.

Contents

List of Figures

List of Tables

List of Symbols

MPC	Model Predictive Control
RHC	Receding Horizon Control
LPV	Linear Parameter-Varying
LTI	Linear Time-Invariant
LQR	Linear Quadratic Regulator
ARE	Algebraic Riccati Equation
HOSVD	Higher Order Singular Value Decomposition
LMI	Linear Matrix Inequality
IBVS	Image Based Visual Servoing
TP	Tensor Product
HSS	Hammersley Sequence Sampling
RMPC	Robust Model Predictive Control
$f(.), g(.), \text{and } h(.)$	Nonlinear functions
x	State vector
y	System output
u	System input
$\mathbb{R}$	Real coordinate space
$\mathbf{S}$	System matrix
$\mathbf{p}$	Vector of uncertain parameters
$A, B, C, \text{and } D$	State space matrices

ω	Weighting function
$\times_n$	n-mode product
$\bigotimes_{n=1}^{N}$	n-mode product of a tensor
$\succ$, $\prec$, $\succeq$, and $\preceq$	Elementwise inequalities
σ	Singular value
k	Time instance
Q, R	Positive definite weighting matrices
V	Lyapunov function
$\Phi(.)$	Terminal cost function
$\mathcal{X}_f$	Terminal constraint set
$*$	Symmetric entries of a symmetric matrix
Ω	Close hypercube
L_s	Interaction matrix
s_m	Image feature vector
V_c	Camera velocity screw
$\mathcal{P}$	Pareto set
$\mathcal{P}_f$	Pareto front
d	Disagreement point
N	Nash Bargaining solution

Chapter 1

Introduction

Model Predictive Control (MPC) or *Receding Horizon Control* (RHC) has been recently attracting more attention than ever, because of the fact that the MPC can take the constraints on inputs, outputs, and state variables directly into account through an optimization problem. During the last decades, much effort has been paid on enhancing the optimization procedure in order to obtain a fast and robust MPC scheme which can be applied on-line and handle uncertainties.

The underlying goal of this thesis is to incorporate uncertainty model and multi-objective trade-off paradigm, which is a very valuable design tool in engineering problems, in the MPC for the complex nonlinear systems. This thesis focuses on linearization techniques, which can reduce the nonlinear system complexity and can model the time-varying and parametric uncertainties. Thus it leads to solve non-convex and intractable nonlinear optimization problems as the linear convex optimization problems.

1.1 Overview

The development of high-performance controllers, which provide acceptable stability, robustness, and optimal behavior of nonlinear systems, has been a major research activity among the control engineering practitioners in recent years. Although, the classical control strategies such as Proportional-Integral-Derivatives (PID) controllers have been adopted in many industrial processes, the highly nonlinear behavior of many dynamical and industrial systems rises the need for more precise controllers [1]. Model predictive control (MPC) or receding horizon control (RHC) is an advanced control strategy which is developed from last three

decades and its applications have been increased in a wide variety range of control engineering [2–4]. The main application of the MPC was in the industrial process [5] and its practical application is very successful [6]; however, nowadays due to the development of multi-core computers, it can be used for controlling the dynamical systems with fast responses such as robotics and aerospace industries [7–10].

Model predictive control is used for both linear and nonlinear systems. The controller design is based on the optimization technique in which the current control action is obtained by minimizing a cost function of the constrained finite-horizon control *on-line* [8, 11]. MPC is an optimal open-loop control strategy; therefore, according to Rawlings [10]: "The difficulty that MPC introduces into the robustness question is the open-loop nature of the optimal control problem and the implicit feedback produced by the receding horizon implementation." In order to handle this issue, the considered objective function should be minimized for finite horizons, that is, the measurement and optimization procedures are accomplished from the current time to some future time instant.

The cost function includes the current measurement and the prediction of the future states, also the current and future control signals based on a discrete-time model of the system [3]. The purpose of considering the finite-horizon objective function and measuring the states at each time step is to compensate for unmeasured disturbances and model uncertainties [12]. The unique characteristic of the MPC is that it can take the constraints on inputs, outputs, and state variables directly into account. Therefore, it has become a popular control method both in theory and practice [3].

The major drawback of MPC is large computational time required to solve the optimization problem which often exceeds sampling interval in real time situation [13]. This computational time, in general, depends on the time that an internal model, called predictor, takes to calculate future response of a system in the optimization procedure [14]. The predictor can be linear or nonlinear model. The advantage of using linear model in MPC is that the related optimization problems are convex and efficiently solvable, even the problem has the large number of design variables [5]. On the contrary, using the nonlinear models results in non-convex optimization problems which cannot be solved in real-time [15]. Recently, much attention has been paid on the nonlinear system modelling using a set of linear models. This method relies on the embedding of a nonlinear system into a polytopic Linear Paramete-Varying (LPV) uncertain system associated with time-varying and state-dependent weighting functions (also called scheduling parameter) [16]. Therefore, the

nonlinear system is approximated by a model which consists of a number of linear models assigned to the trajectories of the original plant [17]. During the last decade, many research activities have been carried out on the control method development for LPV systems [18].

One of the methods in approximating a complex nonlinear system is the well-known Takagi-Sugeno (T-S) fuzzy modelling [19]. By using T-S fuzzy modelling method, the nonlinear system can be modelled with N linear models at each time step, and the system behavior is approximated as a *Convex Combination* of the N linear systems [17]. In fact, such model contains a polytope described by N linear models (vertices) or superposition of linear time-invariant (LTI) models. However, one drawback remains; i.e. N vertices, membership functions and fuzzy rule bases need to be decided for constructing the T-S fuzzy model. They can be regarded as the subjective design parameters, which are inherently difficult to design.

Another method based on Higher Order Singular Value Decomposition (HOSVD) is called the Tensor Product (TP) model transformation. This model can extract the fully orthonormal and singular value ordered structure of the given function [20]. Compared with T-S fuzzy model, TP model has the universal approximation property [21], that is, vast range of dynamical system can be modeled with TP model transformation. Therefore, we are able to use numerical techniques, such as *Linear Matrix Inequality* (LMI) approach to design the controllers providing the stability and minimizing some design specifications.

The vast ranges of the control problems can be formulated as convex optimization problems with LMIs [16], and also there are powerful numerical methods to solve LMI problems [22]. Also, the robust optimization technique presented in the modelling YALMIP [23, 24] toolbox can be used to solve the optimization problem as *on-line*. In this thesis, the primary objective is to provide an efficient method for controlling the complex nonlinear system using TP model and LMI techniques.

In practical control problem, the designers should consider several design specifications which are normally in conflict with each other. Therefore, the optimal control design problems can be formulated as a multi-objective optimization problem, which can be solved by LMI formulation efficiently [25]. The only problem of this approach is that in MPC design at each sampling interval only one optimal solution is needed. However, in multi-objective optimization, a set of non-dominated solutions, called *Pareto Frontier*, are obtained. In this thesis, to cope with this drawback of multi-objective optimization, *Nash Bargaining Solution*

will be used to obtain a trade-off point among Pareto solutions [26]. Therefore, the combination of LMI, robust convex programming, and Nash solution provides an efficient on-line method to solve multi-objective convex optimization problem presented in this thesis.

1.2 Research Objectives and Scopes

The specific objectives and scopes of this thesis can be summarized as follows

1. To propose a computationally efficient robust MPC approach for linear parameter-varying system whose parameters have probabilistic uncertainties. Although there are many different MPC approaches which can handle uncertainties, there is scant MPC method for the systems with probabilistic uncertainties. In this thesis, a robust MPC scheme is proposed to control a system with probabilistic uncertainty.

2. To design a real-time MPC for fast dynamic systems, which is usually considered as one of the challenging control problems among control engineers. In this thesis, based on the TP model transformation and reduced LMI conditions, a real-time MPC scheme is proposed to handle the parametric uncertainties.

3. To consider multi-objective design criteria in MPC design for nonlinear systems, a new MPC scheme using with multi-objective optimization technique is proposed to take into account different objective functions in designing procedure.

1.3 Contributions

The main contributions of the thesis are listed as follows

1. A sufficient stability criteria for exponential stability of MPC has been proposed for uncertain nonlinear systems. The Polytopic LPV models are used to address the stability analysis for a class of nonlinear systems.

2. A robust MPC has been designed to handle the probabilistic uncertainty. Considering probabilistic uncertainty for the uncertain parameters of the system decreases the conservativeness of the conventional robust control methods which are based on the worst-case scenario.

3. A computationally efficient and fast MPC scheme has been developed for fast dynamic systems. The Image Based Visual Servoing (IBVS) system is chosen to assessment the proposed method. To the best of the authors knowledge, this is the first time that MPC method has been successfully applied to IBVS system on-line. All the reported MPC methods applied on the IBVS systems are confined the simulation results. In this thesis, the developed MPC scheme is executed on the real-time experimental set-up consisting of a 6 DOF Denso robot and eye-in-hand camera.

4. A novel MPC scheme has been proposed based on the multi-objective optimization techniques. The multi-objective optimization discovers a set of non-dominated points in the design variables space. To find a single control signal at each sampling time, Nash bargaining solution has been used to select the trade-off point.

1.4 Publications

The presented research work is documented in a number of journals and conference proceedings. The following is the list of author's publications.

Journal Papers

1. A. Hajiloo, M. Keshmiri, W.F. Xie, T.T. Wang,"Robust On-Line Model Predictive Control for a Constrained Image Based Visual Servoing," IEEE Transactions on Industrial Electronics, In Press, 2016.

2. A. Hajiloo, W.F. Xie, "The Stochastic Robust Model Predictive Control of Shimmy Vibration in Aircraft Landing Gears," Asian Journal of Control, Vol. 17, No. 2, pp. 476-485, March 2015,

3. A. Hajiloo, W.F. Xie, S.V. Hoa, S. Khan, "Thermal control design for an automated fiber placement machine," Science and Engineering of Composite Materials, Vol. 21, Issue 3, pp. 427-434, 2014,

4. A. Hajiloo, W.F. Xie, "Multi-Objective Optimal Fuzzy Fractional-Order PID Controller Design," Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 18, Issue 3, pp. 262-270, 2014.

5. A. Hajiloo, W.F. Xie, "Model Predictive Control Based on the Multi-Objective Optimization and Nash Bargaining Solution," Submitted to Expert Systems with Applications, 2016.

6. A. Hajiloo, W.F. Xie, "Robust T-S Fuzzy Modeling and Control Design using Multi-Objective Optimization," Submitted to Journal of Advanced Computational Intelligence and Intelligent Informatics, 2016.

Conference Papers

1. A. Hajiloo, W.F. Xie, and X. Ren, "Multi-Objective Robust Model Predictive Control Using Game Theory," IEEE International Conference on Information and Automation, pp.2026-2030, China, August 2015,

2. A. Hajiloo, M. Keshmiri, W.F. Xie, "On-Line Model Predictive Control for Constrained Image Based Visual Servoing of a Manipulator," SIAM Conference on Control and Its Application, pp. 220-227, Paris, France, July 2015,

3. A. Hajiloo, W.F. Xie, "Robust model predictive control of shimmy vibration in aircraft landing gears with probabilistic uncertainty," Control Applications (CCA), 2014 IEEE Conference on, pp. 146-151, France, August 2014,

4. A. Hajiloo, W.F. Xie, "Multi-objective control design of the nonlinear systems using genetic algorithm," Innovations in Intelligent Systems and Applications (INISTA) Proceedings, 2014 IEEE International Symposium on, IEEE, pp. 27-34, Italy, June 2014,

5. A. Hajiloo, W.F. Xie, "Fuzzy fractional-order PID controller design using multi-objective optimization," IFSA World Congress and NAFIPS Annual Meeting, IEEE, pp. 1445-1450, June 2013.

6. A. Hajiloo, W.F. Xie, "Nash Bargaining Approach to Design Multi-Objective MPC," Submitted to 2016 IEEE Congress on Evolutionary Computation.

1.5 Outline

The thesis starts with an introductory chapter on TP model transformation in Chapter 2. This chapter presents the methods to deal with nonlinear systems including parameters with probabilistic uncertainties. Also, the preliminary notions of MPC are given in Chapter 2. The robust MPC for an LPV system with probabilistic uncertainty is presented in Chapter 3. Chapter 4 is devoted to the design and application of a real-time MPC on an experimental system. As a further designed MPC, a multi-objective MPC scheme is proposed in Chapter 5. Finally, concluding remarks and future work are presented in Chapter 6.

Chapter 2

Literature Review

The main objective of this chapter is to present a literature review on LPV models and MPC design. The major ideas and the methodology that will be used throughout this thesis are addressed here. To this end, the LPV systems modeling, stability analysis and controller synthesis of LPV systems using LMIs techniques will be addressed in this chapter.

2.1 Introduction

With the advent of very powerful algorithms for convex optimization, many convex optimization problems including linear matrix inequalities (LMIs) can be solved very rapidly and efficiently, even for those that have no analytical solutions. As a result, a large number of control engineering problems, which cannot be solved using Lyapunov function, Riccati equations, or other classical techniques, can be reformulated in terms of LMIs so that a handful of standard convex and quasi-convex optimization techniques can be used to solve them[22].

The main idea of the LMI method is to formulate a given problem as an optimization problem with convex objective functions and LMI constraints [16, 22]. A variety of design specification and constraints from control theory such as Lyapunov and Riccati inequalities can be expressed as LMIs [16]. For example, Lyapunov–based stability analysis and the design of the linear systems can be considered as the quadratic matrix inequality.

Some of the problems addressed by LMI techniques includes the following

1. Robust stability of the uncertain systems [16, 21, 22],

2. Multi-objective state feedback controller design [22, 25],

3. Optimal Linear Quadratic Gaussian control [16],

4. Stability analysis of Takagi–Sugeno (T–S) fuzzy system [27, 28].

The use of polytopic model in control system design results in solving the design procedure as convex optimization which involves LMIs [21]. Therefore, a lot of attention has been recently paid to the nonlinear system modelling using a set of linear plants, which allows the control engineers to use LMI methods to design the controllers for intricate nonlinear systems. This method relies on the embedding of a nonlinear system into a polytopic *Linear Parameter-Varying* (LPV) uncertain system associated with time–varying and state–dependent weighting functions (also called scheduling parameter) [16]. Hence, the nonlinear system can be modeled with N linear models, and the system behavior is approximated as a *Convex Combination* of the N linear systems [29]. In fact, such model can be described as a polytope described by N linear models (vertices). In the following sections, some preliminary definitions about LPV models and LMI techniques are presented. Also, the basic concept and notations of model predictive control will be presented in this chapter.

2.2 Robust Nonlinear System Modeling

In real control engineering problems, there exist a variety of uncertainty sources which should be considered through modeling and control system design. Normally, two types of uncertainty, namely, structured and unstructured uncertainties are used in uncertain system classification. Those uncertainties include plant parameter variations due to environmental condition, incomplete knowledge of the parameters, and unmodelled high frequency dynamics.

Consider the following nonlinear system representation as

$$\dot{x}(t) = f(x(t), \mathbf{q}) + g(x(t), \mathbf{q})u(t), \tag{2.1a}$$

$$y(t) = h(x(t), \mathbf{q}), \tag{2.1b}$$

where, $f(.)$, $g(.)$ and $h(.)$ are nonlinear functions, $x = [x_1, x_2, ..., x_n]^T \in \mathbb{R}^n$ is the state vector of the system, $u \in \mathbb{R}^m$ is the input, and $y \in \mathbb{R}^p$ is the output of the system. Also,

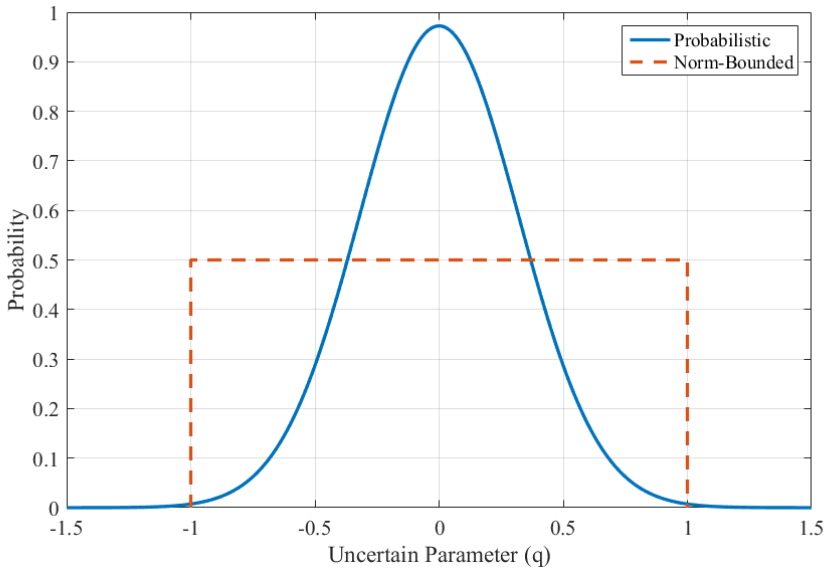


Figure 2.1: Norm-bounded uncertainty vs. Probabilistic uncertainty.

q is the vector of probabilistic uncertain parameters with known probability distribution function (PDF).

One of the conventional methods which can be used for modeling the system with parametric uncertainties is norm-bounded perturbation, and control system design for this type of systems are based on the worst case scenario. In this method, the most possible pessimistic value of the performance for the set of uncertain models is considered. Consequently, the performance of the robust controllers designed based on such norm-bounded model uncertainties is deteriorated for the most likely models of the uncertain systems which may probabilistically happen in the reality [30]. Therefore, in order to reduce the conservatism or accounting more for the most likely plants with respect to uncertainties, in this thesis, the probabilistic uncertainties are propagated through the system model parameters. Figure 2.1 illustrates the difference between norm-bounded and probabilistic uncertainties. It can be seen that for probabilistic uncertainty the likelihood of parameter values is considered, while for norm-bounded uncertainties, the likelihood of each value is the same.

There are different methods for modeling the system uncertainties [16]. In this thesis, the polytopic model is used for modeling the system uncertainties. To find the polytope model of the nonlinear systems (2.1) with parametric uncertainties, the first step is describing them

as a family of linear time-varying system. Every trajectory of the system (2.1) satisfies [16]

$$\dot{x}(t) = A(\mathbf{p}(t), \mathbf{q})x(t) + B(\mathbf{p}(t), \mathbf{q})u(t) \tag{2.2a}$$

$$y(t) = C(\mathbf{p}(t), \mathbf{q})\mathbf{x}(t) + D(\mathbf{p}(t), \mathbf{q})u(t), \tag{2.2b}$$

where the nonlinear system matrix is

$$\mathbf{S}(\mathbf{p}(t), \mathbf{q}) = \begin{pmatrix} A(\mathbf{p}(t), \mathbf{q}) & B(\mathbf{p}(t), \mathbf{q}) \\ C(\mathbf{p}(t), \mathbf{q}) & D(\mathbf{p}(t), \mathbf{q}) \end{pmatrix} \in \mathbb{R}^{(n+p)\times(n+m)} \tag{2.3}$$

and $\mathbf{p}(t) \in \Omega$ is the vector of time-varying parameters. It may include some elements of $x(t)$ or $u(t)$. The set Ω is a closed hypercube which describes the time-varying and parametric uncertainties in this system.

In general, the system presented in (2.2) belongs to the polytopic LPVs. In polytopic models, system matrix, $\mathbf{S}$ varies within a fixed polytope of matrices, that is, Ω is described by a list of its vertices as

$$\Omega = \left\{ \begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix}, ..., \begin{pmatrix} A_L & B_L \\ C_L & D_L \end{pmatrix} \right\}. \tag{2.4}$$

In other words, one has

$$\mathbf{S} := \left\{ \sum_{i=1}^{L} w_i \mathbf{S}_i; \quad w_i \geq 0, \quad \sum_{i=1}^{L} w_i = 1 \right\}, \tag{2.5}$$

where, w_i, $i = 1, \ldots, L$ are weighting functions, also, are the functions of uncertain variables, which imply that the system in (2.5) belongs to convex hull. In other words, any admissible system matrix $\mathbf{S}$ can be written as an unknown convex combination of L vertices $\mathbf{S}_i$, $i = 1, ..., L$, given by

$$\mathbf{S}_i = \begin{pmatrix} A_i & B_i \\ C_i & D_i \end{pmatrix}, \tag{2.6}$$

where A_i, B_i, C_i, and D_i, $i = 1, ..., L$, are constant matrices with appropriate dimensions, which construct i^{th} linear time-invariant (LTI) system [16, 20].

In (2.3), $\mathbf{q}$ is the vector of probabilistic uncertain parameters whose elements are time-independent and have constant values. However, the exact values are unknown. It should

be noted that this kind of uncertainty satisfies the convex condition in (2.5). Suppose q is a random variable with predefined probability function, $\mathbf{Pr}$, which satisfies $\mathbf{Pr}(q) \geq 0$ for all $q \in C$ and $\int_C \mathbf{Pr}(q) q dq = 1$, then $C \subseteq \mathbb{R}^n$ is convex [31].

It should be noted that the dimension of Ω depends on the elements of the uncertain vector (for the sake of notational simplicity, hence, $\mathbf{p}$ is used for both uncertainties related to states and parameters).

The system given in (2.2) can now be written as

$$\begin{pmatrix} \dot{x} \\ y \end{pmatrix} = \sum_{i=1}^{L} w_i(\mathbf{p}) \mathbf{S}_i \begin{pmatrix} x \\ y \end{pmatrix}. \tag{2.7}$$

In order to have an accurate model, the number of LTI models, $\mathbf{S}_i$, should be large. It means in pursuit of a good approximation, one has to face high computational cost. In order to lighten the mentioned contradictory features, Baranyi [21] presented a way in which tensor product (TP) model transformation based on the higher-order singular value decomposition (HOSVD) is used to model an uncertain system with minimal number of vertices. After using this method, the LMI-tools can be used to solve the control design problem.

2.3 Tensor Product Model Transformation

During the last decade, the models representation and identification has changed significantly. The TP model transformation is a recently proposed numerical method to transform LPV models into the parameter varying convex combination of the finite number of LTI components [21]. The main advantage of the TP model transformation is that the LMI based control design techniques can be applied on polytopic TP models.

The following briefly introduces the basic operators of tensor algebra and the main concept of HOSVD.

2.3.1 Definitions and Properties of Tensor Algebra

In literatures, tensor algebra is considered as a follow-up technique on linear algebra, that is, it is a generalization of classical linear algebra. In other words, tensors are the universalities of vectors and matrices [32]. For example, scalars are the tensors of the order 0, vectors are the tensors of the order 1, and matrices are the tensors of the order 2. It is obvious that the

tensors of the order greater than 2 not only is it hard to visualize but also it is difficult to formulate them. The following definitions are adopted from [33] to present the tensors

Definition 2.3.1 (n–mode matrix of tensor $\mathcal{A}$)
Assume an N^{th}-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, then the n-mode matrix of tensor $\mathcal{A}$, which is denoted by $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_{n+1} I_{n+2} \ldots I_N I_1 I_2 \ldots I_{n-1})}$ contains the element $a_{i_1, i_2, \ldots, i_N}$ where the row number i_n and column number r equal

$$\begin{aligned}(i_{n+1}-1)I_{n+2}I_{n+3}\ldots I_N I_1 I_2 \ldots I_{N-1} + (i_{n+2}-1)I_{n+3}I_{n+4}\ldots I_N I_1 I_2 \ldots I_{n-1} + \ldots \\ +(i_N - 1)I_1 I_2 \ldots I_{n-1} + (i_1 - 1)I_2 I_3 \ldots I_{n-1} + (i_2 - 1)I_3 I_4 \ldots I_{n-1} + \cdots + I_{n-1}.\end{aligned} \tag{2.8}$$

This method is called *ordering*, and the purpose of that is to present a higher-order tensor as a matrix. Then, the singular value decomposition can be applied on the tensors. Figure 2.2 illustrates a third-order tensor $\mathcal{A}$ and different n-mode matrices $\mathbf{A}_{(n)}$, $n = 1, 2, 3$. It can be seen that $\mathbf{A}_{(1)}$ is the $I_1 \times I_2 I_3$ matrix, also, $\mathbf{A}_{(2)}$ and $\mathbf{A}_{(3)}$ are the $I_2 \times I_3 I_1$ and $I_3 \times I_1 I_2$ matrices, respectively.

In order to apply HOSVD technique on higher-order tensors, the following definitions regarding the tensors are needed.

Definition 2.3.2 (Scalar product)
Consider two tensors $\mathcal{A}$, $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the scalar product of them is defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} b_{i_1 i_2 \ldots i_N} a_{i_1 i_2 \ldots i_N}. \tag{2.9}$$

Using the notations given in [33], the matrix product can be presented as $\mathbf{H} = \mathbf{U}.\mathbf{G}.\mathbf{V}^T$ in which $\mathbf{G} \in \mathbb{R}^{I_1 \times I_2}$, $\mathbf{U} \in \mathbb{R}^{J_1 \times I_2}$, $\mathbf{V} \in \mathbb{R}^{J_2 \times I_2}$, and $\mathbf{H} \in \mathbb{R}^{J_1 \times J_2}$ by virtue of the $\times_n$ symbol as $\mathbf{H} = \mathbf{G} \times_1 \mathbf{U} \times_2 \mathbf{V}$. This expression means that $\mathbf{G}$ is being multiplied along its rows (the first dimension) by $\mathbf{U}$, and along its columns (the second dimension) by $\mathbf{V}$.

Definition 2.3.3 (n-mode product of a tensor by a matrix)
The n-mode product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$ is an $(I_1 \times I_2 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N)$ tensor whose elements are given by [33]

$$(\mathcal{A} \times_n \mathbf{U})_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} a_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} u_{j_n i_n}. \tag{2.10}$$

The multiple n-mode product of a tensor, such as $\mathcal{A} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \cdots \times_N \mathbf{U}_N$ can be written as $\mathcal{A} \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_n$.

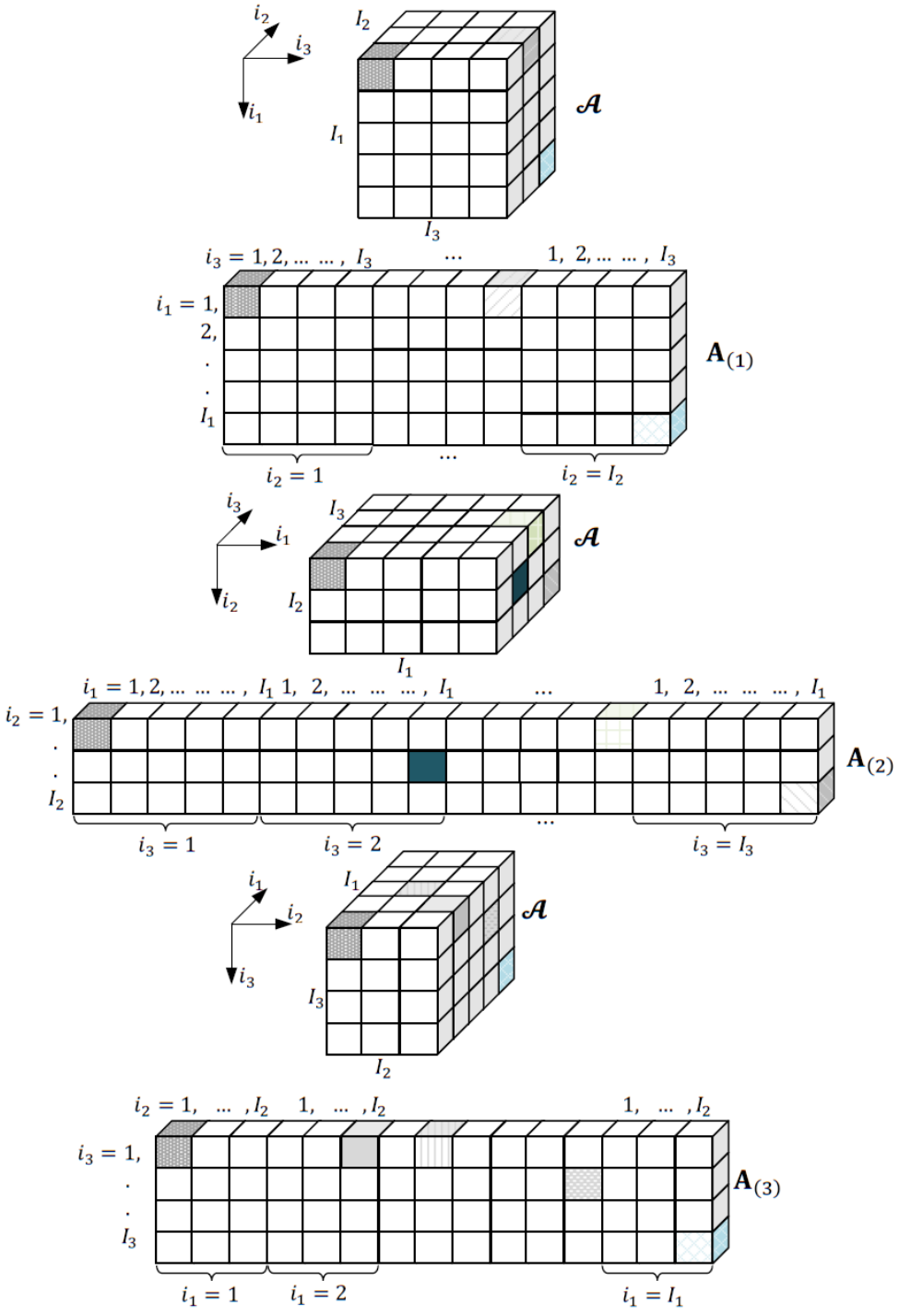


Figure 2.2: Formation of 3-mode matrices of a 3rd-order tensor $\mathcal{A}$.

2.3.2 Higher Order Singular Value Decomposition

A $I_1 \times I_2$ matrix $\mathbf{A}$ can be factored as

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \tag{2.11}$$

where $\mathbf{U}$ is a $I_1 \times I_2$ orthogonal matrix, $\mathbf{V}$ is a $I_2 \times I_2$ orthogonal matrix, and $\mathbf{S}$ is a $I_1 \times I_2$ *pseudodiagonal* matrix with the following nonzero singular values

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0, \tag{2.12}$$

where $r = \min(I_1, I_2)$, that is, the number of nonzero singular values equal the rank of matrix $\mathbf{A}$. Using Definition 2.3.1, (2.11) can be rewritten as follows

$$\mathbf{A} = \mathbf{S}\times_1\mathbf{U}\times_2\mathbf{V}. \tag{2.13}$$

Now, consider a $(I_1 \times I_2 \times \cdots \times I_N)$ tensor $\mathcal{A}$. It can be written as the following form [33]

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times 3 \cdots \times_N \mathbf{U}_N = \mathcal{S} \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_n, \tag{2.14}$$

where $\mathbf{U}_n = (\mathbf{u}_1^n\, \mathbf{u}_2^n\, \ldots \mathbf{u}_{I_n}^n)$, $n = 1 \ldots N$ is a unitary $(I_n \times I_n)$ matrix. Here, $\mathbf{u}_i^n$ is an i^{th}-mode singular vector, and σ_i^n are n-mode singular value of $\mathcal{A}$ which can be ordered for all possible n as follows

$$\sigma_1^{(n)} \geq \sigma_2^{(n)} \geq \cdots \geq \sigma_{I_n}^{(n)} > 0. \tag{2.15}$$

This method is called Higher Order Singular Value Decomposition (HOSVD). Any high order tensor can be approximated using HOSVD method. The minimal or compact form of HOSVD is defined as follows [33]

Definition 2.3.4 (Compact HOSVD)
By applying SVD on each n-mode matrix of $\mathcal{A}$, $n = 1, \ldots, N$, and discarding zero singular values and the related singular vectors, the compact form of HOSVD is obtained as

$$\mathcal{A} = \mathcal{D} \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_n, \tag{2.16}$$

where $\mathbf{U}_n \in \mathbb{R}^{I_n \times R_n}$ and $\mathcal{A} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ with $R_n \leq I_n$.

In many applications, especially in real-time control system design, one should make a trade-off between complexity and accuracy. In order to reduce the system complexity, some nonzero singular values, also, the zero ones can be discarded [18, 21]. This method is called reduced higher order SVD, which is defined as follows

Definition 2.3.5 (Reduced HOSVD)
Discarding the nonzero singular values,$\sigma_{R'_n+1}^{(n)}$, $\sigma_{R'_n+2}^{(n)}$, $\ldots$, $\sigma_{R_n}^{(n)}$, and the corresponding singular vectors, a reduced tensor will be obtained as

$$\hat{\mathcal{A}} = \mathcal{D} \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_n, \tag{2.17}$$

where $\mathbf{U}_n \in \mathbb{R}^{I_n \times R'_n}$ and $\mathcal{A} \in \mathbb{R}^{R'_1 \times \cdots \times R'_N}$ with $R'_n < R_n$. The difference between the exact tensor, $\mathcal{A}$, and the reduced one, $\hat{\mathcal{A}}$, can be approximated as follows [18]

$$\epsilon = \|\mathcal{A} - \hat{\mathcal{A}}\|_2 = \sum_{i_1=R'_1+1}^{R_1} (\sigma_{i_1}^{(1)})^2 + \sum_{i_2=R'_2+1}^{R_2} (\sigma_{i_2}^{(2)})^2 + \cdots + \sum_{i_n=R'_n+1}^{R_n} (\sigma_{i_n}^{(n)})^2. \tag{2.18}$$

In the following section, a brief introduction to linear matrix inequalities is given.

2.4 Linear Matrix Inequalities

Linear Matrix Inequalities (LMIs) are a useful tool for solving a wide variety of optimization and control problems [16]. The major advantage of LMI techniques in control problems is to reduce the original problem to a convex optimization problem. According to Boyd et al. [16]: "The LMIs that arise in system and control theory can be formulated as *convex optimization problems* that are amenable to computer solution."Therefore, the most significant advantage of which is that many control problems involving LMIs can be solved for which there is no analytical solutions [34].

2.4.1 LMI Definition

An LMI has the form [16]

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \succ 0, \tag{2.19}$$

where, $x \in \mathbb{R}^m$ is called decision variables, and $F_i \in \mathbb{R}^{n\times n}$, $i = 0, \ldots, m$, are real fixed symmetric matrices. The inequality symbol in (2.19) means that $F(x)$ is positive definite matrix, that is, $u^T F(x)u > 0$ for all nonzero $u \in \mathbb{R}^n$. Linear inequalities such as those in (2.26), convex quadratic inequalities, matrix norm inequalities, and various constraints from control theory such as Lyapunov and Riccati inequalities can all be presented as LMIs [34]. Also, nonrestrict LMIs are represented in the form $F(x) \succeq 0$.

The LMIs in (2.19) is a convex constraint on x. It means that the set $\{x|F(x) \succ 0\}$ is convex. It can be readily shown that this set is convex. $F(x)$ is a affine function, i.e., $F(x)$ is a linear function which satisfies the following condition

$$F(\theta x_1 + (1-\theta)x_2) = \theta F(x_1) + (1-\theta)F(x_2), \quad \theta \in (0,1),$$

i.e., the LMI (2.19) forms a convex constraint on x.

The multiple LMIs $F_1(x) \succ 0, \ldots, F_p(x) \succ 0$ can be expressed as a single LMI in the following form

$$F(x) = \begin{pmatrix} F_1(x) & 0 & \cdots & 0 \\ 0 & F_2(x) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & F_p(x) \end{pmatrix}. \tag{2.20}$$

Also, a class of convex nonlinear inequality constraints can be converted to an LMI using Schur complement lemma. The convex nonlinear inequalities are in the following form

$$R(x) \succ 0, \quad Q(x) - S(x)R(x)^{-1}S(x)^T \succ 0, \tag{2.21}$$

where $Q(x) = Q(x)^T$, $R(x) = R(x)^T$, and $S(x)$ are affine functions of x. It can be converted into the equivalent LMI as

$$\begin{pmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{pmatrix} \succ 0. \tag{2.22}$$

2.4.2 LMI Applications

As mentioned, many optimization and control problems can be written in terms of a set of LMIs. Normally, LMI problems are divided into two major classes, feasibility and optimization.

The LMI feasibility test leads to finding x^{feas} such that $F(x^{feas}) \succ 0$. Such as Lyapunov stability test in which a matrix P satisfying the LMI should be found

$$P \succ 0, \quad A^T P + PA \prec 0, \tag{2.23}$$

where $A \in \mathbb{R}^{n \times n}$ is given and $P = P^T$ is the variable (in many control problems the variables are matrices).

In the case of optimization problem with an LMI constraint, the goal is to find x^{opt} which optimize a cost function J which satisfies $F(x^{opt}) \succ 0$.

2.4.3 Solving LMI Problems

One of the useful methods in solving the convex optimization problem that include inequality constraints is interior-point method [34]. This method has been developed during the last years and became of true interest in th context of LMI problems [16, 21, 22]. In this method, the constraints are used to define a *barrier function* which is similar to the penalty function in some way [31]. This method allows the constrained optimization problem to be changed into an unconstrained optimization problem which can be solved using Newton's method [34]. In this thesis, the interior-point method is used to solve the optimization problem. Software for solving convex optimization using interior-point methods can be found in commercial software such as MATLAB.

2.5 Model Predictive Control

The mid-seventies to mid-eighties is considered as true birth of MPC. Originally, the MPC strategy developed and became popular in the power plants and petroleum refiners industries, and now it can be found in a vast variety of industrial applications including food processing, chemicals, automotive, and aerospace [5, 35]. The MPC is a control paradigm with a great amount of variants. The variation is mainly based on an explicit process model using to predict the future response of a system, so-called model predictor. Without going into details, the finite impulse response (FIS) model-based MPC and step response model-based MPC are the first algorithms of MPC. Dynamic Matrix Control (DMC), Generalized Predictive Control (GPC) are the latter algorithms. Despite the vast number of algorithms, they all share the same structural features, i.e. based on the optimization of the future system

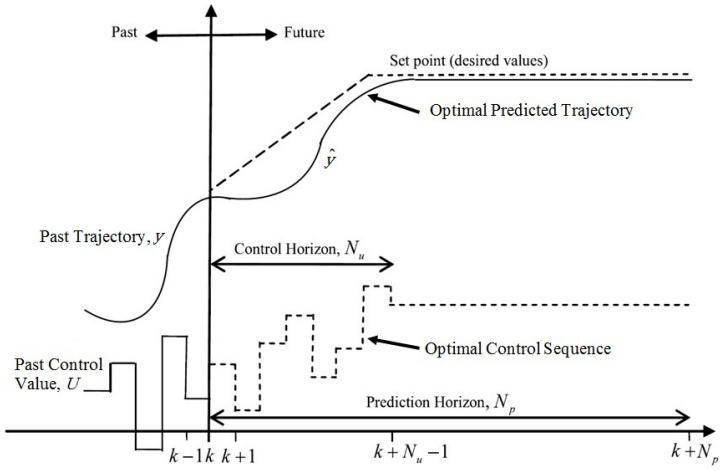


Figure 2.3: Model Predictive Control strategy (Adopted from [14]).

behavior a sequence of future control variables are obtained. The first part of the obtained optimal sequence is then applied to the system, and the entire procedure is repeated after a short time interval [5].

Fig. 2.3 illustrates the principle of the MPC. It can be seen that the optimization is done based on the measurement at time instant k. The optimization procedure leads to obtaining control input vector over *Control Horizon*, $T_u = [k, k+1, ..., k+N_u]$. The model predictor predicts the behavior of the system over an interval which is called *Prediction Horizon*, $T_p = [k, k+1, ..., k+N_p]$. In MPC, N_p is usually equal or greater than N_u [2, 12, 36, 37]. It should be noted that $u_{k+i|k} = u_{k+N_u-1|k}$ for $i \geq N_u$ [37].

In both tracking and regulating control design, the control signal is calculated as the system output is kept as close as possible to the desired trajectory or reference value [2]. As mentioned before, the first control signal is used to control the process and at the next sampling time the optimization is repeated. It should be noted that at the sampling time $k+1$, the current control signal $u_{k+1|k+1}$ will be different to the $u_{k+1|k}$ [2].

2.6 System Representation

In this thesis, a discrete time state-space model is considered. A linear discrete-time system used throughout the thesis can be written as follows

$$x_{k+1} = Ax_k + Bu_k, \tag{2.24a}$$

$$y_k = Cx_k, \tag{2.24b}$$

where, $x \in \mathbb{R}^n$ is the state vector of the system, $u \in \mathbb{R}^p$ is the input, and $y \in \mathbb{R}^m$ is the output of the system. Also, in MPC, the state and control signal must satisfy the following constraints at each sample time k

$$x_k \in \mathbb{X}, \tag{2.25a}$$

$$u_k \in \mathbb{U}, \tag{2.25b}$$

where, $\mathbb{X} \subseteq \mathbb{R}^n$ is a convex and closed set, and $\mathbb{U} \subseteq \mathbb{R}^m$ is considered as a convex and compact set which contains the equilibrium point (origin) in their interior [5]. As they are convex set, they can be, also, described by linear inequalities

$$\mathbf{E}_x x \leq \mathbf{F}_x, \tag{2.26a}$$

$$\mathbf{E}_u u \leq \mathbf{F}_u. \tag{2.26b}$$

In order to obtain the control signal at each sample time, an open-loop optimization problem is introduced. The performance measure or cost function is typically a quadratic function based on the $l_2 - norms$ of the input and output or states deviations from the desired values and includes separate weight matrices for each part to allow the designers to make trade-offs [35]. The quadratic cost function leads to solving efficient optimization problems which are quadratic programming (QP) optimization [13]. By defining positive definite weight matrices $Q \succ 0$ and $R \succ 0$, the control signal is obtained by minimizing the following infinite horizon cost function at sample time k

$$J_k = \sum_{n=0}^{\infty} y_{k+n|k}^T Q y_{k+n|k} + u_{k+n|k}^T R u_{k+n|k}. \tag{2.27}$$

It is obvious when there are no constraints, the control design problem is the same as linear quadratic regulator (LQR) problem in which the control signal is obtained by the solution of an algebraic Riccati equation (ARE) [5].

A finite prediction and control horizons must be chosen to define the finite horizon cost

function. Then the cost function can be written as follows

$$J_k = \sum_{n=0}^{N-1} y_{k+n|k}^T Q y_{k+n|k} + u_{k+n|k}^T R u_{k+n|k}, \tag{2.28}$$

where N is the prediction horizon. In this thesis, N_p and N_u are considered equal N. Throughout this thesis, as an assumption, the exact states are available at each sample time, that is, $x_{k|k} = x_k$.

The basic MPC design can be summarized as follows [35]

1. Measure $x_{k|k}$,
2. Solve the optimization problem given in 2.28 to obtain $u_{.|k}$,
3. Apply the first element of the obtained control signal $u_k = u_{k|k}$,
4. Wait for the new sampling time $k+1$, then go to (1.).

2.7 Quadratic Programming for MPC Design

According to [31], an optimization problem is called a quadratic programming (QP) if the objective function is quadratic, and the constraints are affine. A quadratic program can be written as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T H x + q^T x + r \\ \text{subject to} & Gx \preceq h \\ & Fx = b, \end{array} \tag{2.29}$$

where, H, G, and F are compatible matrices, q, h, and b are vectors. Also, r is an scalar. Without loss of generality, H is assumed to be positive definite matrix ($H = H^T \succ 0$).

To make the appropriate form of the optimization problem in (2.28) for quadratic programming, assuming the current state $x_{k|k} = x_k$ is available, then the future states, control inputs and outputs are denoted as

$$X = [x_{k|k}^T,\, x_{k+|k}^T,\, \ldots,\, x_{k+N-1|k}^T]^T, \tag{2.30a}$$

$$U = [u_{k|k}^T,\, u_{k+|k}^T,\, \ldots,\, u_{k+N-1|k}^T]^T, \tag{2.30b}$$

$$Y = [y_{k|k}^T,\, y_{k+|k}^T,\, \ldots,\, y_{k+N-1|k}^T]^T. \tag{2.30c}$$

The compact matrix form of the predicted states and output are given as

$$X = \mathcal{A}x_{k|k} + \mathcal{B}U, \tag{2.31a}$$

$$Y = \mathcal{C}X, \tag{2.31b}$$

where $\mathcal{A} \in \Re^{nN \times n}$, $\mathcal{B} \in \Re^{pN \times mN}$, and $\mathcal{C} \in \Re^{pN \times nN}$ are defined as [38]

$$\mathcal{A} = \begin{pmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{N-1} \end{pmatrix}, \tag{2.32a}$$

$$\mathcal{B} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{N-2}B & \cdots & AB & B & 0 \end{pmatrix}, \tag{2.32b}$$

$$\mathcal{C} = \begin{pmatrix} C & 0 & \cdots & 0 \\ 0 & C & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & C \end{pmatrix}. \tag{2.32c}$$

Now by substituting the predicted outputs and inputs in (2.28), we can have a convex optimization problem in the QP form. To this end, the weighting matrices are defined as the block diagonal matrices with N blocks. They are given as

$$\mathcal{Q} = \begin{pmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & Q \end{pmatrix}, \tag{2.33a}$$

$$\mathcal{R} = \begin{pmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & R \end{pmatrix}. \tag{2.33b}$$

Now, the optimization problem in (2.28) can be as

$$\begin{aligned} \underset{U}{\text{minimize}} \quad & Y^T \mathcal{Q} Y + U^T \mathcal{R} U \\ \text{subject to} \quad & \mathcal{E}_u U \leq \mathcal{F}_u \\ & \mathcal{E}_x X \leq \mathcal{F}_x, \end{aligned} \tag{2.34}$$

where, $\mathcal{F}_u = (\mathbf{F}_u^T, \mathbf{F}_u^T, \ldots, \mathbf{F}_u^T)^T$ and $\mathcal{F}_x = (\mathbf{F}_x^T, \mathbf{F}_x^T, \ldots, \mathbf{F}_x^T)^T$ represent the imposed input and output constraints whose number of rows equal the number of constraints and number of columns equal the prediction horizon, N.

By substituting (2.31) into (2.34), the final quadratic programming can be obtained as follows

$$\begin{aligned} \underset{U}{\text{minimize}} \quad & U^T(\mathcal{A}^T \mathcal{C}^T \mathcal{Q} \mathcal{C}^T \mathcal{A} + \mathcal{R})U + 2U^T \mathcal{B}^T \mathcal{Q} \mathcal{C} \mathcal{A} x_{x|x} \\ \text{subject to} \quad & \mathcal{E}_u U \leq \mathcal{F}_u \\ & \mathcal{E}_x \mathcal{B} \leq \mathcal{F}_x - \mathcal{E}_x \mathcal{A} x_{k|k}, \end{aligned} \tag{2.35}$$

It is interesting to note that the optimization problem in (2.35) will be solved analytically if it is an unconstrained problem or has the equality constraints. For the unconstrained case the global optimal solution that will give a minimum of the cost function can be readily obtained as

$$U = -(\mathcal{A}^T \mathcal{C}^T \mathcal{Q} \mathcal{C}^T \mathcal{A} + \mathcal{R})^{-1} \mathcal{B}^T \mathcal{Q} \mathcal{C} \mathcal{A} x_{k|k}. \tag{2.36}$$

By considering the first m rows of the optimal solution in (2.36), the optimal state feedback control law can be written as follows

$$u_k = F x_{k|k}. \tag{2.37}$$

where $F\ in \mathbb{R}^{m \times n}$ is state-feedback gain matrix. In this case by considering $N = \infty$, the solution is the optimal linear quadratic controller.

2.8 MPC Stability Analysis

Although the model used in (2.35) is linear, the constraints introduce the nonlinearities which necessitate the use of Lyapunov stability. A lot of research works have been devoted to MPC stability analysis. However, there is no general and unique stability theory developed for MPC. A comprehensive survey on stability theory for MPC can be found in [11]

To ensure the closed-loop stability for MPC, one has to define a terminal cost function, $\Phi(x)$, terminal constraint set, $\mathcal{X}_f$, and nominal controller, $L(x) = Lx$ so that they satisfy the following assumptions [11]

1. $\mathcal{X}_f \subset \mathcal{X}$, $\mathcal{X}_f$ is closed and contains origin (0),
2. $\forall x \in \mathcal{X}_f$, $L(x) \in \mathcal{U}$,
3. $\forall x \in \mathcal{X}_f$, $Ax + BL(x) \in \mathcal{X}_f$,
4. $\Phi(0) = 0$, and $\forall x \neq 0$, $\Phi(x) \geq 0$,
5. $\forall x \in \mathcal{X}_f$, $\Phi(Ax + BL(x)) - \Phi(x) \leq -x^T Q x - L(x)^T(x) R L(x)$.

Assumptions 1 and 2 imply that the state constraint and the control constraint are satisfied in $\mathcal{X}_f$. Assumption 3 means that $\mathcal{X}_f$ is a positively invariant set under $L(.)$. Also, Assumptions 4 and 5 signify that $\Phi(.)$ is a Lyapunov function.

Using the above-mentioned assumptions, the following optimization problem will provide asymptotic stability for an MPC controller

$$\begin{aligned} \underset{U}{\text{minimize}} \quad & \sum_{n=0}^{N-1} x_{k+n|k}^T Q x_{k+n|k} + u_{k+n|k}^T R u_{k+n|k} + \Phi(x_{k+N|k}), \\ \text{subject to} \quad & u_{k+n|k} \in \mathcal{U} \\ & x_{k+n|k} \in \mathcal{X} \\ & x_{k+N|k} \in \mathcal{X}_f. \end{aligned} \tag{2.38}$$

Providing that the states start in $\mathcal{X}_f$, and $u_k = L(x_k)$ is used as the controller, it is obvious that $u_k \in \mathcal{U}$, and $x_{k+1} \in \mathcal{X}_f \subset \mathcal{X}$. And Assumption 5 can be written as

$$\Phi(x_{k+1}) - \Phi(x_k) \leq -x_k^T Q x_k - u_k^T R u_k. \tag{2.39}$$

Summing up both sides of this inequality from time instant k to infinity

$$\Phi(x_\infty) - \Phi(x_k) \leq \sum_{n=0}^{\infty} -x_{k+n}^T Q x_{k+n} - u_{k+n}^T R u_{k+n}, \tag{2.40}$$

and letting $x_\infty = 0$, based on the terminal cost definition $\Phi(x_\infty) = 0$ yield the upper bound of the quadratic function as

$$\sum_{n=0}^{\infty} x_{k+n}^T Q x_{k+n} + u_{k+n}^T R u_{k+n} \leq \Phi(x_k). \tag{2.41}$$

Now, it can be concluded that the terminal cost function, $\Phi(x)$ is the upper bound for infinite horizon cost function at sample time k.

There are a variety of different methods to define a terminal cost function, terminal constraint set, and nominal controller [5, 11, 39–41]. Ideally, the terminal cost $\Phi(.)$ can be chosen as a quadratic terminal cost $\Phi(x) = x^T P x$, where $P \in \mathbb{R}^n$, and $P \succ 0$. In the case of stable system, a stabilizing controller is $F(x) = 0$ which satisfies the control constraints for all states, that is, $\mathcal{X}_f = \mathbb{R}^n$, and then Assumption 5 simplifies as

$$x_k^T A^T P A x_k - x_k^T P x_k \leq -x_k^T Q x_k. \tag{2.42}$$

Then the following statement can be obtained as

$$A^T P A - P \preceq -Q. \tag{2.43}$$

Therefore, to design the stable MPC, a Lyapunov equation is solved to find P at each sampling time.

2.9 LMI Based Robust MPC Design

When a control system is robust it means that it should provide the acceptable level of stability and performance in the presence of uncertainties. Designing a controller for a system with both constraints and uncertainties is usually challenging. In this case, the robust optimization methods must be used to design MPC. When an optimization problem includes uncertainty in cost function and constraints, it is called robust optimization which

can be defined as follows

$$\begin{aligned} &\underset{x}{\text{minimize}} \max_{w} f(x,w) \\ &\text{subject to} \qquad g(x,w) \preceq 0 \quad \forall w \in \mathcal{W}, \end{aligned} \tag{2.44}$$

where x is the design variable vector, and w is the uncertain variable. In the literatures, this optimization problem is also known as *Minimax* optimization problem which leads to minimizing the worst-case value of the cost function[23].

In order to design the MPC for polytopic model, the goal is to find a linear state feedback $u_k = Fx_k$ that minimizes an upper bound of the worst-case infinite horizon quadratic cost in (2.41), which is the function $x_k^T P x_k$ [35]. Now (2.39) can be rewritten as

$$x_{k+1}^T P x_{k+1} - x_k^T P x_k \leq -x_k^T Q x_k - u_k^T R u_k, \tag{2.45}$$

by adding up both left and right-hand side from 0 to ∞ and inserting a linear state feedback $u_k = Lx_k$, the following matrix inequality will be obtained as

$$(A_k + B_k F)^T P (A_k + B_k F) - P \preceq -Q - F^T R F, \tag{2.46}$$

where A_k and B_k are system matrices obtaining by TP model transformation from all possible vertices at sampling time. Now, by converting (2.46) to LMIs matrices P and F can be attained.

2.10 Summary

In this chapter, a brief literature review on LPV models, MPC design and stability analysis are presented. Also, the LMIs technique as a useful method to solve the convex optimization problem is addressed in this chapter.

In the following chapters, a robust and real-time methods will be introduced to design the MPC for nonlinear systems using robust optimization and LMI techniques.

Chapter 3

Shimmy Vibration Control Using Robust MPC

The development of the robust control systems is owing to the fact that there is usually a source of uncertainty in the system modeling, e.g. disturbance, unmodeled dynamics, parameter uncertainty and measurement noise. Some characteristics such as stability, sensitivity, and robustness are defined to appraise how a control system can deal with uncertainty. A control system is called robust when the stability is maintained and the performance specifications are met in the presence of a specified uncertainty range [42]. The main objective of this chapter is to propose sufficient Lyapunov stability and stabilization criteria, based on the mentioned methods in Chapter 2, for LPV systems with probabilistic uncertainty.

3.1 Introduction

Robust control is a synthesis that optimizes worst-case performance specification and identifies worst-case parameters while the plant remains varying in some specified set. The uncertain model appears when the system parameters are not precisely known, or may vary over a given range. In [43], a Robust MPC (RMPC) synthesis is proposed that allows explicit incorporation of the description of plant uncertainties. In addition, the problem of minimizing an upper bound on the worst-case is reduced to a convex optimization involving LMIs. It has been proved that the solution of the LMI optimization problem, which is the receding horizon state-feedback control, can robustly stabilize the set of uncertain plants. Motivated by [43], Cuzzola *et al.*,[44] presented a new approach based on the use of several

Lyapunov functions, each of which corresponds to a different vertex of the uncertainty's polytope. Wada *et al.*, [45] proposed a method for synthesizing the MPC law for Linear Parameter Varying (LPV) systems by using the Parameter Dependent Lyapunov Function (PDLF) and claimed less conservative control performance. Also, the computational load involved in RMPC is a challenging problem in an on-line RMPC application. To deal with this drawback, a modified strategy is proposed here to stabilize the LPV system with reduced computational load.

To investigate the stability and robustness of the proposed method, the RMPC is used to suppress the shimmy vibration which is one of the major concerns in the aircraft landing gear design. The kinetic energy of the forward motion of the aircraft provides the energy for this type of vibration, which leads to the self-excited torsional oscillation of tires about the vertical axis [46]. Also, shimmy vibration may result in instability. This oscillatory motion may also be induced by the forces produced by runway surface irregularities and non-uniformities of the tires. The shimmy vibration typically has a frequency range between 10 to 30 Hz [47]. The analysis of shimmy formation can be found in [46–48].

In this chapter, the aircraft landing gear shimmy dynamics model presented in [49, 50] is studied with the following variable parameters; caster length, taxiing velocity and spring stiffness. The considered linearized landing gear system is a typical Linear Parameter Varying system, whose state-space matrices are functions of those varying parameters. Using the TP model transformation method given in Chapter 2, a discrete polytopic LPV model for the aircraft landing gear system is easily obtained.

The control objective is to steer the yaw angle to zero in order to suppress the shimmy when the landing gear system is subjected to uncertainties, which are varying taxiing velocity, and wheel caster length during landing; also, torsional spring stiffness is considered as the probabilistic uncertain parameter. Therefore, both time-varying and probabilistic uncertain parameters are considered in this study. The performance of the designed controller is verified by simulation results, which shows that the proposed RMPC using the LMI approach leads to finding a solution at each sample time with guaranteed closed-loop stability, high computational efficiency and strong disturbance rejection ability. Compared with the designed RMPC dealing with only one time-varying parameter (taxiing velocity) [49], the proposed RMPC can handle two time-varying parameters, i.e. caster length, taxiing velocity and one probabilistic uncertain parameter, i.e. spring stiffness.

The chapter is organised as follows. Section 3.2 is devoted to describing a modified

RMPC algorithm for a LPV system. TP model transformation method is given in Section 3.3. Section 3.4 presents shimmy model and analysis for typical landing gear. The simulation results of the proposed control strategies in shimmy suppression are presented to demonstrate the control performance and the computational efficiency in Section 3.5. Concluding remarks are presented in Section 3.6.

3.2 Robust MPC Design

Consider the LPV system (2.2) subjected to the input and output constraints

$$||u_{k+i|k}||_2^2 \leq u_{max}, \tag{3.1a}$$

$$||y_{k+i|k}||_2^2 \leq y_{max}, \tag{3.1b}$$

where k is the time instant, u_{max} is the upper bound on input, y_{max} is the upper bound on output, and $i = 0, 1, ...$ is the future sample time. The control objective is to find a control law $u(k)$, so that the state variables, $x(k)$, can be steered to zero in desirable time. The control signal is updated at each sampling instant via the minimization of the finite robust objective performance with respect to the uncertain parameters and constraints (3.1) at each sampling instant k. Using *Minimax* optimization method, the objective function can be written as follows

$$\underset{u}{\text{minimize}} \max_{\mathbf{p}} \sum_{i=0}^{N-1} x_{k+i|k}^T Q x_{k+i|k} + u_{k+i|k}^T R u_{k+i|k}, \tag{3.2}$$

where, $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are positive definite weighting matrices. In the RMPC design, the Lyapunov function is chosen as the terminal cost function (see Section 2.8) defined as $V(x) = x_{k|k}^T P x_{k|k}$. Using Assumption 5 given in Section 2.8, one can achieve the optimal performance objective (3.2) as follows

$$V(x_{k+i|k}) - V(x_{k|k}) \leq -x_{k|k}^T Q x_{k|k} - u_{k|k}^T R u_{k|k}. \tag{3.3}$$

Summing up the above inequality from time instant k to ∞ and $x_{\infty|k} = 0$ yields the

following constraint as

$$\sum_{i=0}^{\infty} x_{k+i|k}^T Q x_{k+i|k} + u_{k+i|k}^T R u_{k+i|k} \leq V(x_{x|x}), \tag{3.4}$$

i.e., the Lyapunov function is the upper bound for infinite horizon cost function at sample time k. Based on (2.46), the problem of designing the controller is to find P_k and F_k where the control law is $u_{k+i|k} = F_k x_{k+i|k}$, $i \geq 0$. This control law, indeed, minimizes the upper bound on the robust performance objective function at sampling time, k.

By performing a congruence transformation with $X_k = P_k^{-1} \succ 0$ defining

$$F_k = Y_k X_k^{-1}, \tag{3.5}$$

and applying Schur compliment, the following linear objective minimization problem [43] will be obtained as

$$\underset{\gamma_k, X_k, Y_k}{\text{minimize}}\, \gamma_k \tag{3.6a}$$

subject to

$$\begin{bmatrix} 1 & x_k^T \\ x_k & X_k \end{bmatrix} \succeq 0, \tag{3.6b}$$

$$\begin{bmatrix} X_k & * & * & * \\ A_j X_k + B_j Y_k & X_k & * & * \\ Q^{1/2} X_k & 0 & \gamma \mathbf{I} & * \\ R^{1/2} Y_k & 0 & 0 & \gamma \mathbf{I} \end{bmatrix} \succeq 0, \tag{3.6c}$$

$$\begin{bmatrix} u_{max}^2 & Y_k \\ Y_k^T & X_k \end{bmatrix} \succeq 0, \tag{3.6d}$$

$$\begin{bmatrix} y_{max}^2 \mathbf{I} & (A_j X_k + B_j Y_k)^T C^T \\ C(A_j X_k + B_j Y_k) & X_k \end{bmatrix} \succeq 0, \tag{3.6e}$$

where, the output matrix, C, is common for all vertices, and $j = 1, 2, ..., L$ (L is the number of vertices). Also, the symbol $*$ represents a symmetric structure in LMIs.

Under the above designed closed-loop feedback law, the solution for the optimization in (3.6a) can be obtained using the LMI technique, which results in stabilizing the LPV system (3.33) and the state variables are steered to zero. At each sampling time, an optimal

upper bound on the worst-case performance cost over the infinite horizon is obtained by forcing a quadratic function of the state to decrease by at least the amount of the worst case performance cost at each prediction time. Such on-line step-by-step optimization can lead to asymptotically stable evolution. But for the real-time application, especially for the shimmy suppression in the landing gear system, the computational efficiency of LMI is very critical to guarantee the practical implementation of RMPC. To improve the computational efficiency, one needs to sacrifice the optimal performance to the computational load at each step.

In order to alleviate the computational load in RMPC, an attempt is made to reduce the dimension of matrices in LMI Eqs. (3.6b) to (3.6e). Inequality (3.6c) involves the most computational load since it has to be satisfied by every vertex $[A_j, B_j]$. It is found that the matrices in rows 3 and 4 in inequality (3.6c) are directly related to the robust performance index, which appears at the right hand side of inequality (3.3). The modified RMPC will trade the optimal performance with the computational load [49].

In order to accomplish the above, a Lyapunov function $V(x) = x_{k|k}^T P_k x_{k|k}$ is defined where $P_k := \beta_k X_k^{-1}$ is a positive definite matrix, which will be obtained by solving the optimal problem at current time, k. In order to guarantee the asymptotic stability of the closed-loop system, which is the main objective of controller design, a Lyapunov function for the closed-loop system must be defined, which is strictly decreasing

$$V(x_{k+1|k}) - V(x_{k|k}) < 0, \tag{3.7}$$

and is equivalent to the following inequality

$$x_{k+1|k}^T P_k x_{k+1|k} - x_{k|k}^T P_k x_{k|k} < 0, \tag{3.8}$$

where the measured state variables at time $k+1$ are assumed to be equal to the predicted state variables $x_{k+1|k}$. In order to guarantee that inequality (3.8) holds, the following inequality must be ensured

$$x_{k+1|k+1}^T P_{k+1} x_{k+1|k+1} < x_{k+1|k+1}^T P_k x_{k+1|k+1}. \tag{3.9}$$

The inequality in terms of X_{k+1} and X_k using the definition of P_k can be written as

$$x_{k+1|k+1}^T \beta_{k+1} X_{k+1}^{-1} x_{k+1|k+1} < x_{k+1|k+1}^T \beta_k X_k^{-1} x_{k+1|k+1}. \tag{3.10}$$

The RMPC is designed to make the matrix X_{k+1}^{-1} at time $k+1$ to be smaller than X_k^{-1} at time k. This is equivalent to the case where the matrix X_k^{-1} is smaller than the matrix X_{k-1}^{-1}. It can be written in the following form by adding an upper bound $\beta\mathbf{I}$ to matrix X_k

$$X_{k-1} \prec X_k \prec \beta_k \mathbf{I}. \tag{3.11}$$

Furthermore, to guarantee that inequality (3.8) is satisfied, the right hand side of inequality (3.9) needs to satisfy the following inequality

$$x_{k+1|k+1}^T P_k x_{k+1|k+1} = x_{k+1|k}^T \beta_k X_k^{-1} x_{k+1|k} \leq x_{k|k} \beta_k X_k^{-1} x_{k|k}. \tag{3.12}$$

Substituting $x_{k+1|k} = A(\mathbf{p})x_{k|k} + B(\mathbf{p})u_k$ and feedback control $u_k = F_k x_{k|k}$ into the above inequality leads to

$$x_{k|k}^T (A(\mathbf{p}) + B(\mathbf{p})F_k)^T \beta_k X_k^{-1} (A(\mathbf{p}) + B(\mathbf{p})F_k) x_{k|k} - x_{k|k}^T \beta_k X_k^{-1} x_{k+1|k+1} \leq 0. \tag{3.13}$$

The following inequality will be obtained by using the definition of feedback control gain in (3.5) and polytopic model defined in (3.36),

$$\sum_{j=1}^{L} (w_j A_j + w_j B_j Y_k X_k^{-1})^T \beta_k X_k^{-1} \sum_{j=1}^{L} (w_j A_j + w_j B_j Y_k X_k^{-1}) - \beta_k X_k^{-1} \preceq 0, \tag{3.14}$$

which is equivalent to

$$\sum_{j=1}^{L} w_j (A_j + B_j Y_k X_k^{-1})^T \beta_k X_k^{-1} \sum_{j=1}^{L} w_j (A_j + B_j Y_k X_k^{-1}) - \beta_k X_k^{-1} \preceq 0, \tag{3.15}$$

using the convex hull definition in (2.5), the above inequality holds if the following L inequalities hold [16]

$$(A_j + B_j Y_k X_k^{-1})^T \beta_k X_k^{-1} (A_j + B_j Y_k X_k^{-1}) - \beta_k X_k^{-1} \preceq 0, \quad j = 1, 2, ..., L. \tag{3.16}$$

Then, the following LMIs will be obtained by applying Schur compliment as

$$\begin{bmatrix} X_k & (A_j + B_j Y_k)^T \\ A_j + B_j Y_k & X_k \end{bmatrix} \succeq 0, \quad j = 1, 2, ..., L. \tag{3.17}$$

In summary, the modification of the RMPC given in [49] for LPV system (3.33) subjected to the input and output constraints can be expressed as the following minimization problem.

$$\underset{\beta_k, X_k, Y_k}{\text{minimize}}\, \beta_k \tag{3.18a}$$

subject to

$$\begin{bmatrix} X_k & (A_j + B_j Y_k)^T \\ A_j + B_j Y_k & X_k \end{bmatrix} \succeq 0, \tag{3.18b}$$

$$X_{k-1} \prec X_k \prec \beta_k \mathbf{I}, \tag{3.18c}$$

$$\begin{bmatrix} u_{max}^2 & Y_k \\ Y_k^T & X_k \end{bmatrix} \succeq 0, \tag{3.18d}$$

$$\begin{bmatrix} y_{max}^2 X_k & (A_j + B_j Y_k)^T C^T \\ C(A_j + B_j Y_k) & y_{max}^2 \mathbf{I} \end{bmatrix} \succeq 0, \tag{3.18e}$$

The feedback control gain is obtained as $F_k = Y_k X_k^{-1}$ and the control signal is calculated as $u_{k+i|k} = F_k x_{k+i|k}$, $i \geq 0$.

The detailed step by step control algorithms are summarized as bellow.

Modified RMPC Algorithm:

Given an initial state x_0, the controller for LPV system is implemented as

1. Solve (3.6a) for $k = 0$ to obtain X_0 and Y_0, subjected to (3.6b) to (3.6e) with initial condition x_0, and constraints on input u_{max} and output y_{max}. Save the corresponding X_0, Y_0.

2. For $k = 1$ find Y_k and X_k by solving minimization problem (3.18a), subjected to (3.18b) to (3.18e), calculate the feedback gain as $F_k = Y_k X_k^{-1}$ and control input as $u_k = F_k x_k$

3. Apply the control input u_k to the LPV system, and $k = k + 1$ then go to 2.

As given above, the algorithm starts with solving the time-consuming minimization problem subjected to LMIs (3.6b) to (3.6e) before the iteration and then switches to the modified RMPC [51]. The optimization problem outside the loop is solved to obtain the initial X_0 which will serve as the lower bound of the next step X_1 in the loop. In each step, the matrix

X_k is solved and used for feedback gain F_k computation. The step-by-step optimization of the above problem can lead to asymptotically stable evolution.

The proposed algorithm is based on RMPC and the invariant ellipsoid concept [43]. In step 1, the upper bound (γ_0) of Lyapunov function for the initial state variables x_0 is minimized and the inequality $x_0^T P_0 x_0 \leq \gamma_0$ is held. The obtained subset of state space $x \in \mathbb{R}^n$, $\mathcal{X}_f = \{x | x^T X_0^{-1} x \leq 1\}$ becomes an invariant ellipsoidal terminal constraint set of state variables. And γ_0 also becomes the upper bound of Lyapunov function for system (3.33) by solving the proposed optimization problem in Step 2.

According to LMI optimization theory [16] the fastest Interior Point algorithms computational effort grows with $N \times J^3$; where N is the total row size of LMIs, and J is the total number of decision variables. Considering the minimization problem (3.18a), the total row size of LMIs (3.18b) to (3.18e) has been reduced compared with that of LMIs (3.6b) to (3.6e) in each iteration. Thus the computational load is significantly reduced.

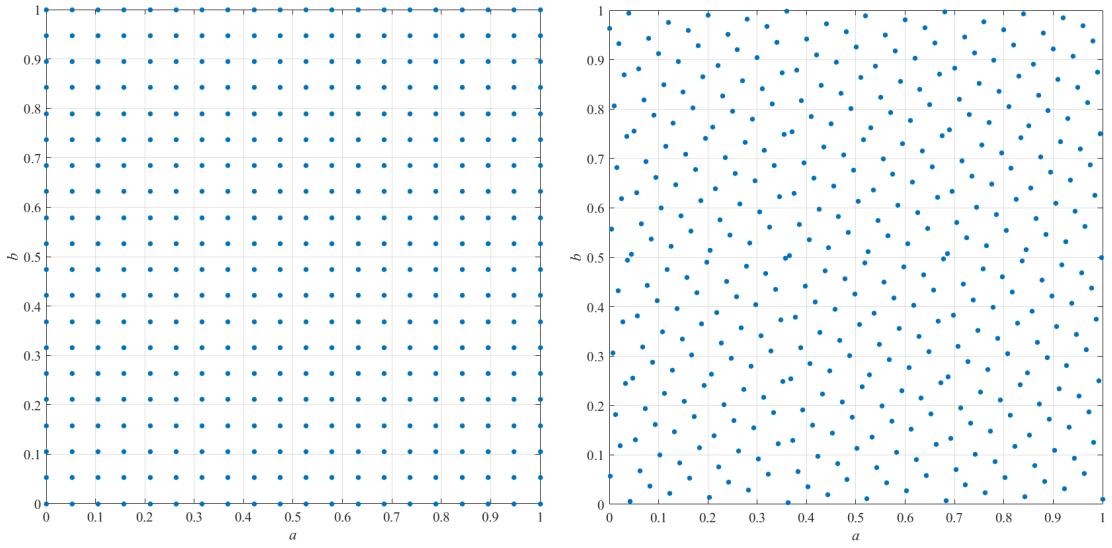
In summary, consider the LPV system (3.33) with $[A(\mathbf{p})|B(\mathbf{p})]$ varying in a polytope Ω (Convex Hull) of vertices $[A_1|B_1]$, $[A_2|B_2]$, ..., $[A_L|B_L]$. Assume that the system is subjected to input and output constraints (3.1). The state feedback matrix F_k in the control law $u_k = F_k x_k$ is given by $F_k = Y_k X_k^{-1}$, where $X_k \succ 0$ and Y_k are obtained from the solution to the linear objective minimization problem (3.18a) subjected to LMIs (3.18b) to (3.18e). Then the obtained control law robustly stabilizes the closed-loop system asymptotically.

3.3 TP Model Construction

The following is briefly presented to construct the TP model transformation of a given LPV model (the comprehensive details are given in [33]).

Step 1: To construct the TP model, the given LPV model must be discretized over the transformation space Ω which is a bounded hyper-rectangular space. The uncertain parameters of the system vary inside that space: $\mathbf{p} \in \Omega : [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_N, b_N]$, also, the weighting functions are defined over this interval. It means that the resulting TP model is only explicable in Ω.

Then, the hyper-rectangular space must be discretized by defining the grids in Ω to generate samples. In [33], an equidistance location of the grids is suggested. In this method, an N–dimensional hyper rectangular equidistant grid–by–grid net over the closed hypercube



(a) (b)

Figure 3.1: Sampling methods (a) equidistance generated samples, and (b) Hammersley Sequence Samples.

Ω is generated as follows

$$g_{n,m_n} = a_n + \frac{b_n - a_n}{M_n - 1}(m_n - 1), \qquad n = 1, \ldots, N, \tag{3.19}$$

where, N is the total number of the variable parameters in Ω. a_n, and b_n are the minimum and maximum of the closed hypercube elements on each dimension, respectively. Also, $a_n \leq g_{n,m_n} \leq b_n$, $m_n = 1, \ldots, M_n$, stands for the corresponding grid line locations, and M_n is the number of grids on n^{th} dimension [33].

In this thesis, the quasi Monte-Carlo sampling method, also called Hammersley Sequence Sampling (HSS) is used to generate samples. This method produce evenly distributed sample points in the sample space without having a high correlation between the points or not forming a regular grid. It has an advantage over purely deterministic methods, since, deterministic methods only give high accuracy when the number of samples increases [30]. Figure 3.1 illustrates both equidistance sampling methods and HSS for the same number of samples. It is evident that HSS method (Figure 3.1b) can cover the space more than equidistance method (Figure 3.1a). As a result, by using HSS method, a more accurate model can be attained with the lower number of samples compared with the equidistance method.

The Hammersley sequence is based on the inverse radix notation. The prime numbers are usually used as the radix ($\mathfrak{R}$). The radix notation of an integer $\mathfrak{p}$ is calculated as follows [30]

$$\mathfrak{p} = \mathfrak{p}_m \mathfrak{p}_{m-1} \dots \mathfrak{p}_1 \mathfrak{p}_0, \tag{3.20a}$$

$$\mathfrak{p} = \mathfrak{p}_0 \mathfrak{R}^0 + \mathfrak{p}_1 \mathfrak{R}^1 + \dots + \mathfrak{p}_m \mathfrak{R}^m, \tag{3.20b}$$

then by reversing the digits of $\mathfrak{p}$ about the decimal point, a unique fraction between 0 and 1 known as the inverse radix number can be obtained as

$$\Psi_R(\mathfrak{p}) = 0.\mathfrak{p}_0 \mathfrak{p}_1 \dots \mathfrak{p}_m, \tag{3.21a}$$

$$\Psi_R(\mathfrak{p}) = \mathfrak{p}_0 \mathfrak{R}^{-1} + \mathfrak{p}_1 \mathfrak{R}^{-2} + \dots + \mathfrak{p}_m \mathfrak{R}^{-m-1}. \tag{3.21b}$$

The Hammersley point of a k-dimensional hypercube are obtained as

$$\mathrm{x}_k(\mathfrak{p}) = \left[\frac{\mathfrak{p}}{N}, \Psi_{R_1}(\mathfrak{p}), \Psi_{R_2}(\mathfrak{p}), \dots \Psi_{R_{k-1}}(\mathfrak{p}) \right]. \tag{3.22}$$

This vector of Hammersley points gives a distributed numbers on the unit hypercube $[0, 1]^k$. Any given domain, $[a, b]$, can be now generated by mapping the obtained number between 0 and 1 using HSS.

A simple example is given to show the advantage of HSS over equidistance sampling method. The following function is considered

$$f(x) = (1 + x_1^{-3} + x_2^{-1.5})^2,$$

in which the transformation space defined as $\Omega = [1, 5] \times [1, 5]$. Both methods are used to generate the samples with different number of grids. Then, the TP transformation is used to find the function approximation, $\hat{f}(x)$, and the mean of error between the actual function and approximated one for different sample numbers are given in Table 3.1. It is obvious that the computational cost of TP modeling transformation can be reduced by virtue of HSS instead of equidistance method.

Table 3.1: The mean of the error between actual function and approximated one.

Methods	Grid Lines		
	5×5	20×20	50×50
Equidistance	$3.14e^{-2}$	$1.21e^{-6}$	$3.59e^{-9}$
HSS	$2.32e^{-2}$	$1.59e^{-7}$	$2.41e^{-11}$

After generating the sample points in Ω, a vector of grid points can be defined as follows

$$\mathbf{g}_{m_1,m_2,\ldots,m_N} = \begin{pmatrix} g_{1,m_1} \\ \vdots \\ g_{N,m_N} \end{pmatrix}. \tag{3.23}$$

The discretized system matrix $\mathbf{S}(\mathbf{p}) \in \mathbb{R}^{(n+p)\times(n+m)}$ is a tensor $\mathcal{S}^{\mathcal{D}}$ ($\mathcal{D}$ stands for "discretized") with the size $M_1 \times M_2 \times \ldots M_N \times (n+p) \times (n+m)$. The elements of $\mathcal{S}^{\mathcal{D}}$ are

$$\mathcal{S}_{m_1,m_2,\ldots,m_N} = \mathbf{S}(\mathbf{g}_{m_1,m_2,\ldots,m_N}). \tag{3.24}$$

Step 2: In this step, HOSVD is executed on the obtained tensor $\mathcal{S}^{\mathcal{D}}$. Using (2.14)

$$\mathcal{S}^{\mathcal{D}} = \mathcal{S} \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_n. \tag{3.25}$$

where the size of S is $I_1 \times I_2 \times \ldots I_N \times (n+p) \times (n+m)$ and I_n is the number of singular values and vectors obtained at each dimension. Now the compact HOSVD or the reduced HOSVD can be used to reduce the size of the original tensor.

Step 3: The final step of TP model transformation is to find the weighting functions. In this method the unitary matrix $\mathbf{U}_n$ gives the weighting functions. The i_n^{th} column vector,$\mathbf{u_{n,i_n}}$, of the matrix $\mathbf{U}_n$ determines the i^th_n column vector $w_{n,i_n}(p_n)$ of $\mathbf{w}_n^{\mathcal{D}}(p_n)$ evaluated at the discretized value of $p_n = g_{n,i_n}$.

The corresponding tensor $\mathcal{S}^{\mathcal{D}}$ for any parameter $\mathbf{p}$ becomes

$$\mathcal{S}^{\mathcal{D}} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \prod_{n=1}^{N} w_{n,i_n}(p_n) \mathbf{S}_{i_1,i_2,\ldots,i_N}. \tag{3.26}$$

In literatures, there are different types of weighting functions [18, 21, 52]. In this thesis, the weighting functions have the following properties

1. The sum of the weighting functions for all $p \in \Omega$ is 1, that is, $\sum_{i_n=1}^{I_n} w_{i_n}(p) = 1$. Therefore, the weighting function is *Sum Normalized* (SN).

2. The weighting functions values for all $p \in \Omega$ is *Non-Negative* (NN), $w_{i_n}(p) \geq 0$.

In this case, SN and NN types of weighting functions, the TP model including the LTI vertices which make a convex hull of the system matrix $\mathbf{S}(\mathbf{p})$.

In the following sections, the landing gear model will be presented, then the proposed method will be used to suppress the shimmy vibration of the landing gear.

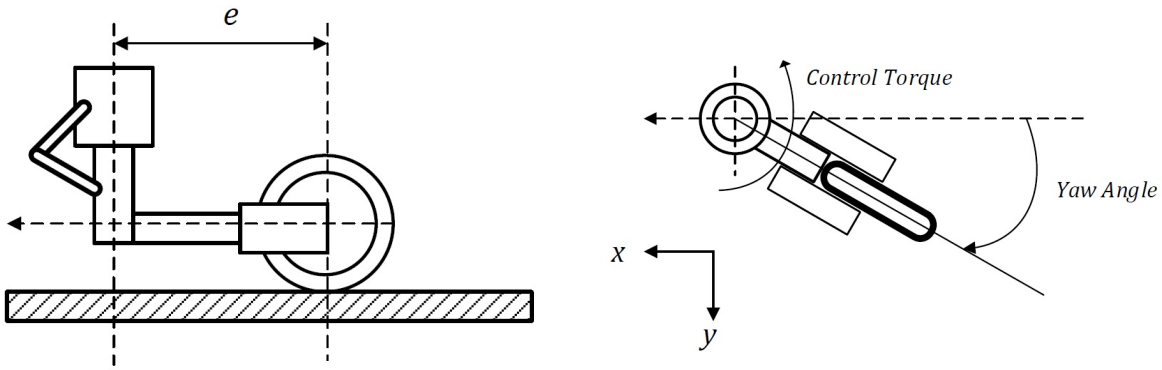
3.4 LPV Shimmy Modeling

Shimmy vibration can lead to serious problems such as excessive wear, shortened life cycle of gear parts, safety concerns, and discomfort for pilots and passengers. In order to suppress shimmy motion, shimmy damper is used in Boeing 737 and Airbus A-320 aircraft as a conventional preventive measure. However, as mentioned in [48], shimmy damping requirements often conflict with good high-speed directional control; furthermore, once the landing gear design is completed, the structural parameters for shimmy suppression cannot be changed. Hence, when external disturbances or uncertain parameters arise in the landing gear system, no further action can be taken. In some operation situations such as worn parts, severe climate, and rough runway, active control strategy can be effective for shimmy vibration control. With the advent of high-speed and highly reliable microprocessors used in controller implementation, the idea of actively controlled landing gear has gained new momentum [49]. Even though the concept of active landing gear is not new (started in the seventies), no production aircraft is as yet equipped with such a system, as reported in [50]. Furthermore, there is scant research on developing the control strategy that can deal with time varying parameters and the uncertainty of landing gear. The following, the dynamic model of the landing gear is presented.

An aircraft landing gear model described in [49, 50] is considered, and also shown in Figure 3.2. According to [49], the state-space equations of the linearized model can be written as

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{3.27a}$$

$$z(t) = Cx(t), \tag{3.27b}$$



(a) (b)

Figure 3.2: (a) Side view and (b) top view of Landing Gear model.

where,

$$x = \begin{pmatrix} \psi \\ \dot{\psi} \\ y \end{pmatrix}, \tag{3.28}$$

is the state vector of the system, including, yaw angle, $\psi\,(rad)$, yaw rate, $\dot{\psi}\,(rad/s)$, and lateral deflection, $y\,(m)$. z is the output of the system, yaw angle. Also, matrix $A \in \mathbb{R}^{3\times3}$, $A \in \mathbb{R}^{3\times1}$, and $C \in \mathbb{R}^{1\times3}$ are defined as follows

$$A = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{K}{I_z} & -\frac{c}{I_z} + \frac{\kappa}{VI_z} & \frac{F_z}{I_z\sigma}(C_{M\alpha} - eC_{F\alpha}) \\ V & e - a & -\frac{V}{\sigma} \end{pmatrix}, \tag{3.29a}$$

$$B = \begin{pmatrix} 0 \\ k_e \\ 0 \end{pmatrix}, \tag{3.29b}$$

$$C = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}. \tag{3.29c}$$

In this model, taxiing velocity, $V\,(m/s)$, wheel caster length, $e\,(m)$, and torsional spring constant, $K\,(N.m/rad)$, are considered as variable parameters. The values of fixed parameters are given in Table 3.2.

The landing gear is assumed to taxi along the runway with a varying taxiing velocity from $80\,m/s$ to $20\,m/s$ within 3 seconds. Taxiing velocity is critical in the shimmy analysis

Table 3.2: Parameters of the model.

Parameter	Value	Unit
Half contact length, a	0.1	m
Moment of Inertia, I_z	1	$kg.m^2$
Vertical force, F_z	9000	N
Torsional damping constant, c,	20	$\frac{N.m}{rad.s}$
Side force derivative, $C_{F\alpha}$	20	$\frac{1}{rad}$
Moment derivative, $C_{M\alpha}$	-2	$\frac{m}{rad}$
Tread width moment constant, κ	-270	$\frac{N.m^2}{rad}$
Relaxation length, $\sigma = 3 \times a$	0.3	m
Moment constant, k_e	10000	$\frac{N.m}{volt}$

[46]. It was reported in [49], lower taxiing speed leads to higher stability and the landing gear becomes stable at the lower speed of $10\,knots$ ($5.144\,m/s$). It is also reported in [48] that the shimmy vibration increases with raising the velocity. Before the aircraft touches down, forward velocity is supposed to be lower than $150\,knots$ ($77.2\,m/s$).

In addition, landing gear designers consider the caster length, e, as a time varying parameter [48], and it is now treated as another uncertain parameter, which belongs to the following set

$$e \in [0.1,\, 0.5]\,. \tag{3.30}$$

Also, the torsional spring constant, K, is considered as an uncertain parameter. For this parameter, Gaussian probability distribution with the mean value of $100000 N.m/rad$ and the standard deviation equals to 8500 is considered. The probability distribution function (PDF) of this uncertain parameter is shown in Figure 3.3.

The vector of uncertain parameters is defined as $\mathbf{p} = \{V,\, e,\, K\} \in \Omega$, which is the element of the closed hypercube $\Omega = [V_m, V_M] \times [e_m, e_M] \times [K_m, K_M] \subset \mathbb{R}^3$. The value of the parameters in Ω are considered as $V_m = 20$, $V_M = 80$, $e_m = 0.1$, $e_M = 0.5$, $K_m = 75000$, and $K_M = 125000$.

In order to investigate the uncertainties effect on stability, 100 samples are generated using HSS method from Ω. It should be noted that the Gaussian probability distribution shown in Figure 3.3 is used to generate the probabilistic uncertain parameter K samples. The open-loop poles of 100 samples are shown in Figure 3.4. The obtained results show that

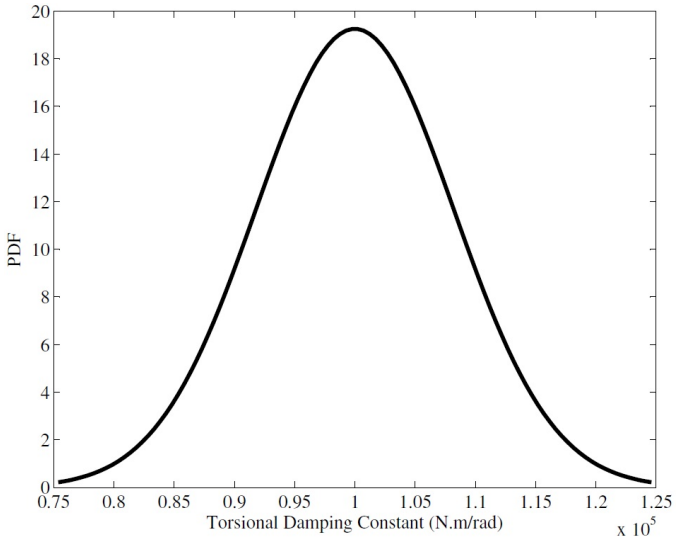


Figure 3.3: PDF of the torsional spring constant (K).

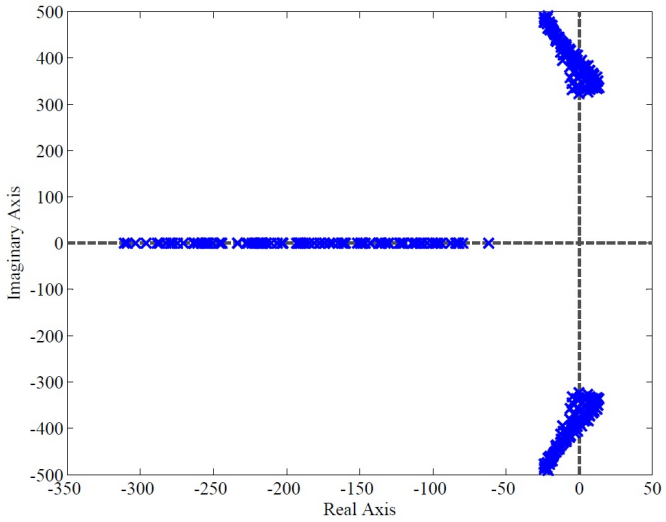


Figure 3.4: Position of the uncertain shimmy open-loop poles.

Table 3.3: The mean of the error between actual function and approximated one.

Weighting functions No.	Mean of Error
$3 \times 2 \times 2$	0.011
$2 \times 2 \times 2$	0.013
$1 \times 2 \times 2$	0.014

36 out of 100 systems have poles on the right half plane, it means that the probability of instability for this system is 36% with respect to considered uncertainties. Also, most of the dominant poles are located close to the imaginary axis that lead to oscillatory behavior with large overshoot. Therefore, the designed controller should provide stable and robust performance.

The discrete time model instead of continuous time dynamic model given in 3.27 is used to design the MPC. The discrete time landing gear state-space model can be written as

$$x(t+1) = A_d x(t) + B_d u(t), \tag{3.31}$$

where A_d, and B_d are discrete time system and input matrices defined as follows

$$A_d = e^{hA}, \qquad B_d = A^{-1}\left(e^{hA} - I\right)B, \tag{3.32}$$

and h is the sample interval equal to 0.01 in this study.

In order to generate the hyper-rectangular N-dimensional space grid and using the TP model transformation, 100 samples are generated on each dimension using HSS methods for discretization.

Using (3.32) to discretize the system, makes both system and input matrices time varying. The system matrix of the obtained LPV model can be written as

$$\mathbf{S}(\mathbf{p}) = \begin{pmatrix} A_d(\mathbf{p}) & B_d(\mathbf{p}) \end{pmatrix}. \tag{3.33}$$

As mentioned before, the density of the sampling grids is considered as $100 \times 100 \times 100$. After using HOSVD method on each 3-dimension of the tensor $\mathcal{S} \in \mathbb{R}^{100\times100\times100\times3\times4}$, the

nonzero singular values in each dimension are obtained as follows

$$\begin{array}{lll} \sigma_1^{(1)} = 269.4 & \sigma_1^{(2)} = 268.1 & \sigma_1^{(3)} = 269.3 \\ \sigma_2^{(1)} = 0.061 & \sigma_2^{(2)} = 26.2 & \sigma_2^{(3)} = 7.7 \\ \sigma_3^{(1)} = 0.009 & & \end{array}$$

A trade-off between complexity and accuracy should be made to reduce the computational load of the control design. Three different models can be obtained based on the obtained singular values. If all of the nonzero singular values are kept, the TP model has $3 \times 2 \times 2 = 12$ weighting functions which is same number as vertices. Also, the models with $2 \times 2 \times 2 = 8$ or $1 \times 2 \times 2 = 4$ vertices can be extracted by discarding $\sigma_3^{(1)}$ or both $\sigma_2^{(1)}$ and $\sigma_3^{(1)}$, respectively. The mean of error between actual LPV model and TP models which are tested for 100 samples are given in Table 3.3. As it can be seen, the error between compact HOSVD model $(3 \times 2 \times 2)$ and the reduced one $(1 \times 2 \times 2)$ is small; therefore, trading accuracy with complexity and choosing the small number of vertices can result in designing a real-time MPC. The error between the compact tensor $\mathcal{S}$ and the reduced one, $\hat{\mathcal{S}}$, by discarding singular values $\sigma_2^{(1)}$, and $\sigma_3^{(1)}$ can be approximated by

$$\|\mathcal{S} - \hat{\mathcal{S}}\|^2 \leq (\sigma_2^{(1)})^2 + (\sigma_3^{(1)})^2 \approx 0.003. \tag{3.34}$$

Therefore, the results show that the aircraft landing gear model can be approximately given in the HOSVD-based polytopic model form with minimum $1 \times 2 \times 2 = 4$ linear time invariant (LTI) vertex models. The sum normalization (SN) and non-negativeness (NN) type weighting functions are used [21] to obtain the convex TP model which can satisfy LMI control design conditions. The weighting functions for all three models are illustrated in Figure 3.5. The LTI system matrices of the polytopic TP model are

$$\begin{aligned} A_{d_{1,1,1}} &= \begin{pmatrix} 0.94 & 0.001 & -0.183 \\ -121.61 & 0.83 & -345.69 \\ 0.018 & 0.0004 & 0.76 \end{pmatrix}, \\ B_{d_{1,1,1}} &= \begin{pmatrix} 0.048 \\ 93.75 \\ 0.022 \end{pmatrix} \end{aligned} \tag{3.35a}$$

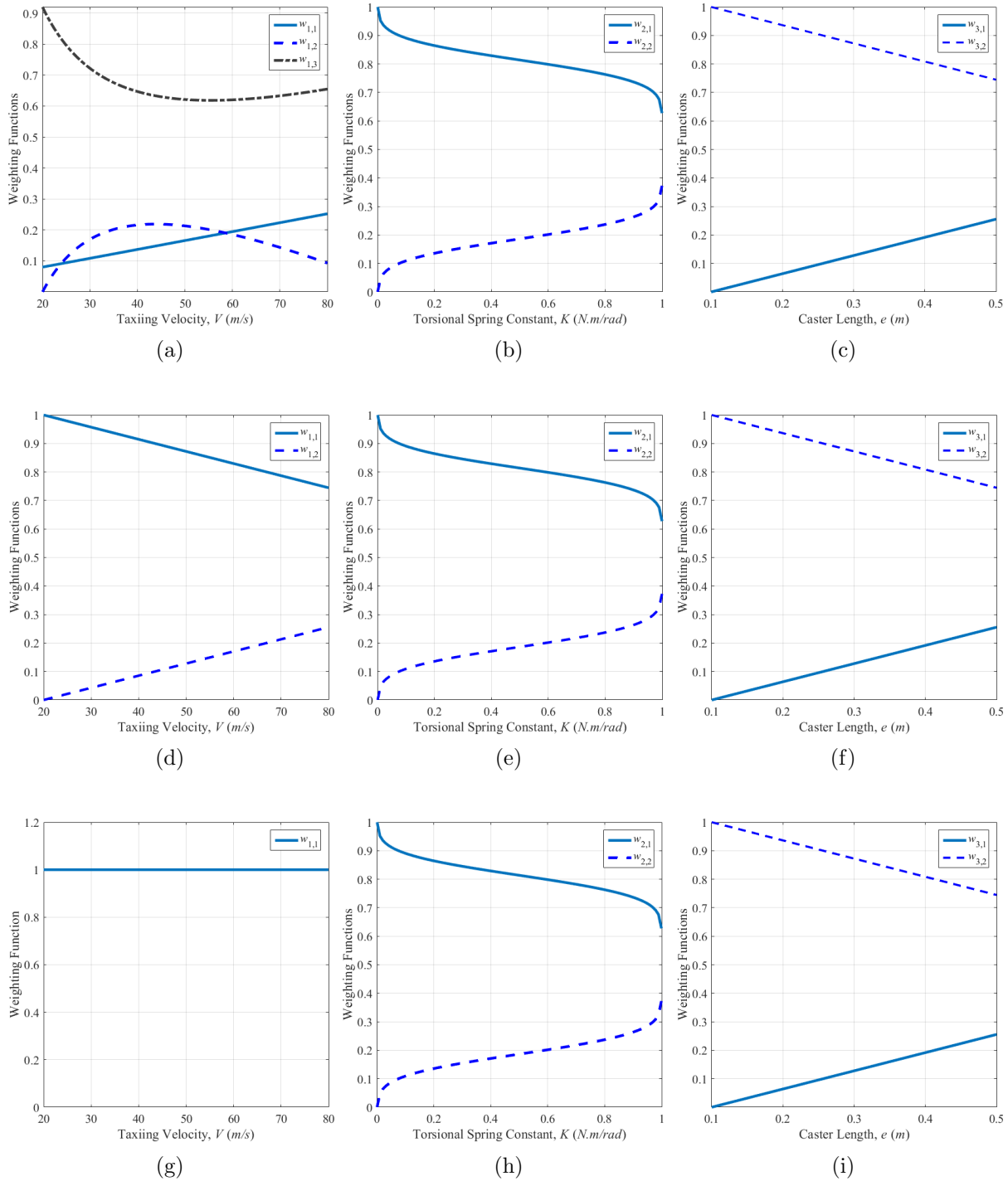


Figure 3.5: SNNN type weighting functions of the TP model for, (a-c) $3 \times 2 \times 2$, (d-f) $2 \times 2 \times 2$, and (h-i) $1 \times 2 \times 2$ vertices models.

$$A_{d_{1,1,2}} = \begin{pmatrix} 0.96 & 0.001 & -0.184 \\ -85.01 & 0.85 & -348.13 \\ 0.027 & 0.0004 & 0.76 \end{pmatrix}, \quad B_{d_{1,1,2}} = \begin{pmatrix} 0.048 \\ 94.38 \\ 0.022 \end{pmatrix} \tag{3.35b}$$

$$A_{d_{1,2,1}} = \begin{pmatrix} 0.94 & 0.001 & -0.057 \\ -118.75 & 0.91 & -109.29 \\ 0.045 & 0 & 0.84 \end{pmatrix}, \quad B_{d_{1,2,1}} = \begin{pmatrix} 0.049 \\ 96.71 \\ 0.001 \end{pmatrix} \tag{3.35c}$$

$$A_{d_{1,2,2}} = \begin{pmatrix} 0.96 & 0.001 & -0.057 \\ -80.97 & 0.93 & -110.05 \\ 0.045 & 0 & 0.84 \end{pmatrix}, \quad B_{d_{1,2,2}} = \begin{pmatrix} 0.049 \\ 97.36 \\ 0.001 \end{pmatrix} \tag{3.35d}$$

The discrete time TP model of the aircraft landing gear system can be written as follows

$$x(t+1) = \mathbf{S}(\mathbf{p}) \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} = \sum_{i=1}^{2} \sum_{j=1}^{2} w_{2,i}(e) w_{3,j}(K)(A_{d_{1,i,j}} \mathbf{x}(t) + B_{d_{1,i,j}} u(t)), \tag{3.36}$$

where $w_{2,i}$, $i = 1, 2$, and $w_{3,i}$, $j = 1, 2$, are weighting functions, also, w_1 is the weighting function for V which is equal to 1 for all values of V (shown in Figure 3.5g).

The system $[A_d(\mathbf{p})|B_d(\mathbf{p})]$ varies in a polytope Ω (Convex Hull) of vertices $[A_{d_{1,1,1}}|B_{d_{1,1,1}}]$, $[A_{d_{1,1,2}}|B_{d_{1,1,2}}]$, $[A_{d_{1,2,1}}|B_{d_{1,2,1}}]$, and $[A_{d_{1,2,2}}|B_{d_{1,2,2}}]$, which satisfies convexity conditions in 2.5.

The control objective is to design a RMPC for the LPV system based on the LTI vertex models.

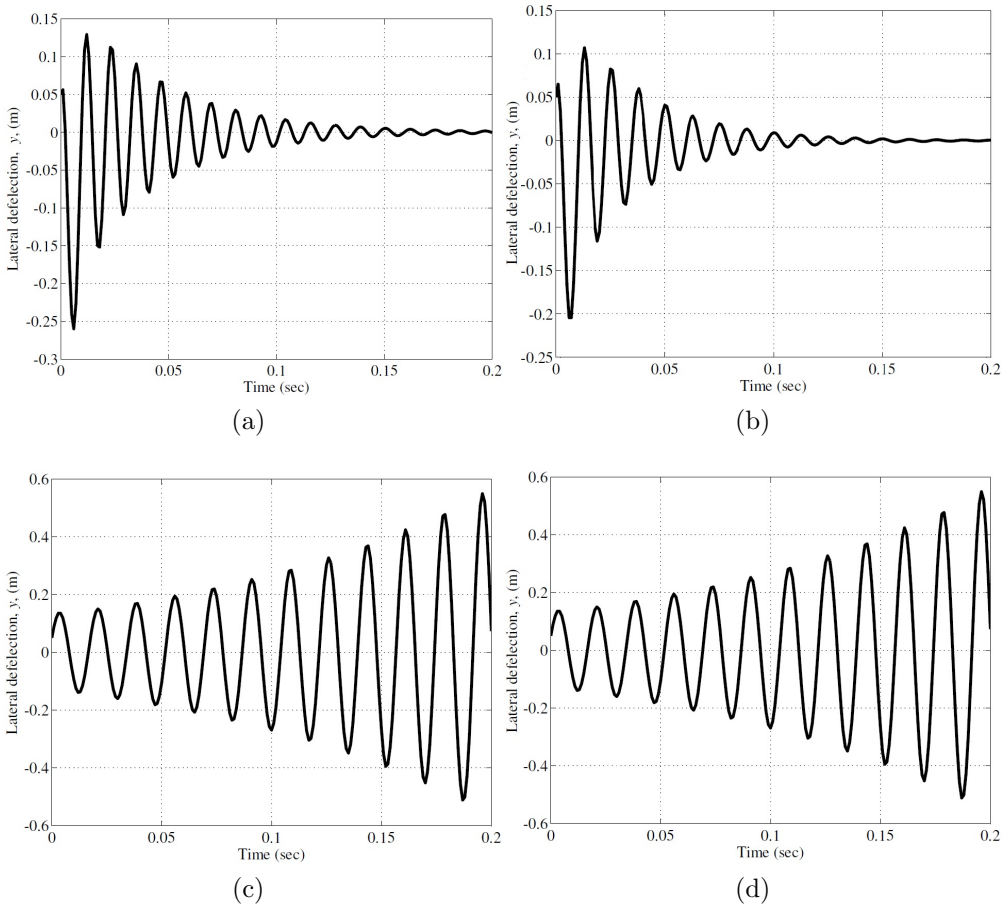


Figure 3.6: Lateral deflection (m) in taxiing for (a) vertex 1, (b) vertex 2, (c) vertex 3, and (d) vertex 4 without controller.

3.5 RMPC Simulation Results

The modified RMPC algorithm presented in Section 3.2 is applied on the linear landing gear model. The objective of shimmy control is to asymptotically suppress yaw vibration with low overshoot and short settling time during the landing process and to robustly stabilize the landing gear system.

Figure 3.6 shows the shimmy vibration of the landing gear in lateral deflection without the controller for different vertices in (3.35). It can be seen that two vertices are stable (vertex 1 and 2) with the high frequency oscillation, and two vertices are unstable (vertex 3 and 4). Therefore, the designed state-feedback controller should stabilize the unstable landing gear system with good high-speed directional control.

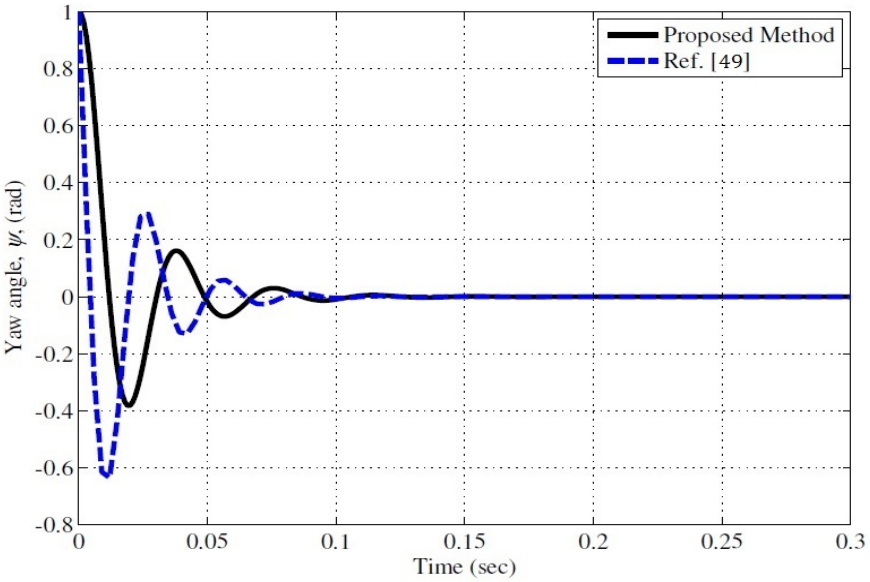


Figure 3.7: Yaw angle (rad) in taxiing.

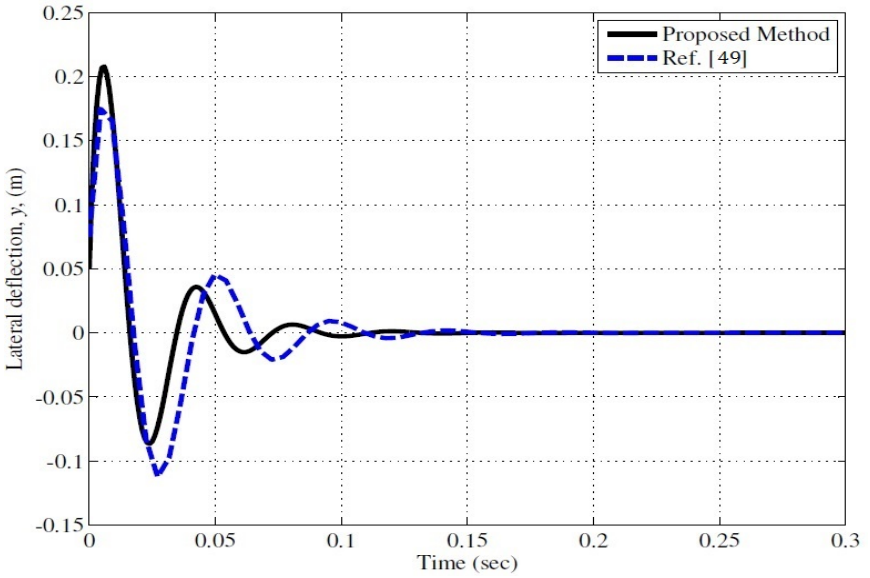


Figure 3.8: Lateral deflection (m) in taxiing.

In order to design the controller, the control input and output constraints are considered as $2.5\,volt$ and $1\,rad$, respectively. The initial conditions for the state is $[\psi,\, \dot{\psi},\, y]^T = [1,\, 0,\, 0.05]^T$. Also, weight matrices for the robust MPC are chosen as $Q = \mathbf{I}_{3\times 3}$ and $R = 1$. In shimmy control design, all involved LMIs are effectively solved by Matlab + YALMIP

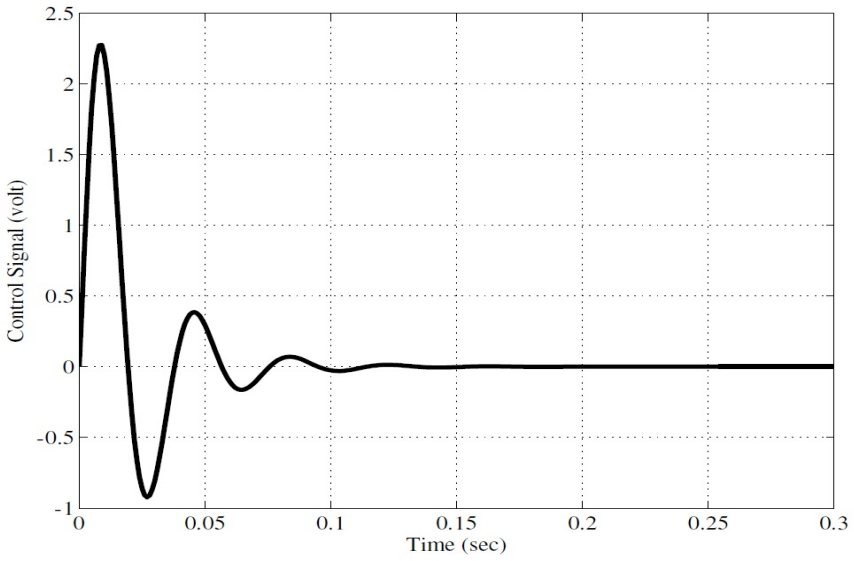


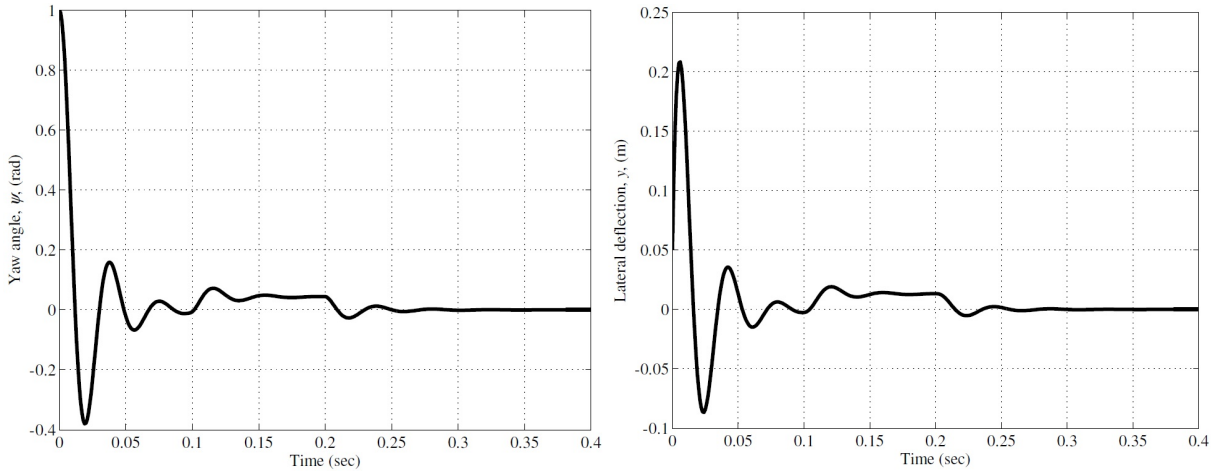
Figure 3.9: Control input (volt).

Toolbox [24].

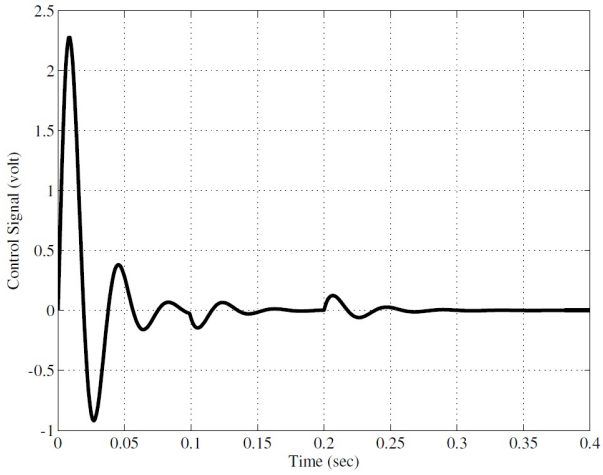
The simulation results are shown in Figures 3.7, 3.8, and 3.9. It can be seen that both yaw angle and lateral deflection approach to zero less than 0.15 *sec*. The landing gear body should not have large shimmy, and the shimmy oscillations should dissipate as quickly as possible; therefore, the designed controller demonstrates promising performance and robust stability in the presence of uncertainties. Also, the results from [49] are shown in Figures 3.7 and 3.8. It can be seen that the state convergence is almost the same as that in [49] with smaller overshoot when dealing with taxiing velocity. However, it should be noted that in [49] only taxiing velocity was considered as the varying-parameter, and in this thesis three varying-parameters are considered. The proposed RMPC in [49] cannot handle the case when the system is subjected to varying caster length and torsion spring.

In order to investigate the disturbance rejection ability of the proposed method, the system is subjected to external disturbance. In this study, 0.2 *volt* step disturbance on the input is considered at time 0.1 *sec*, which lasts for 10 time steps. The simulation results are shown in Figures 3.10a, 3.10b, and 3.10c. It is evident that the controller can deal with disturbance and stabilize the system with the small overshoot and settling time.

According to the simulation results, the proposed method can be effectively used to design the robust MPC for the LPV system, which shows good performance and robust stability.



(a) (b)



(c)

Figure 3.10: (a)Yaw angle, (b) Lateral deflection, and (c) Control input with step disturbance at 0.1 *sec*.

3.6 Conclusion

In this chapter, an active control strategy is proposed for landing gear shimmy control using RMPC. Also, the TP model transformation and HOSVD technique are used to reduce the computational load of the control design. A modified RMPC algorithm using the LMI method is proposed to improve the computational efficiency in the optimization problem. Introducing RMPC state feedback, the control law is calculated by step-by-step optimization and the LMI solutions can be found to stabilize the LPV system with disturbance rejection ability.

The simulation results demonstrate that the modified RMPC can effectively suppress shimmy vibration for the nominal operation range of an aircraft during landing when the taxiing velocity changes from $80\,m/s$ to $20\,m/s$, and wheel caster length varies between $0.1\,m$ and $0.5\,m$. Also, Gaussian probability distribution is considered for the torsional spring constant.

Chapter 4

Real-Time MPC for Image-Based Visual Servoing

A well recognized imperfection of MPC is that it cannot be implemented in the systems with fast dynamics, where the sampling interval is small. However, there are a few techniques for applying fast MPC which are mainly based on the computing the control law offline. Then, the on-line controller can be implemented as a lookup table [53]. This method works well when there are few constraints, small state and input dimensions, and short time horizons. Moreover, this method is very sensitive to the disturbances and uncertainties. In other words, it cannot work robustly to deal with th disturbance or uncertainty in real time; because the lookup table is designed offline. While the system is working, any disturbance or uncertainty may deteriorate the controller performance.

The major goal of this chapter is to propose a real-time MPC method based on the method proposed in Chapter 3 and on-line optimization technique for a fast dynamic system which can handle uncertainty and be executable on an experimental set-up. For this purpose, a 6 degree-of-freedom (DOF) manipulator and Image-Based Visual Servoing Control are used to apply real-time MPC.

4.1 Introduction

Visual servoing has been used extensively in robotics as a solution to make the machines faster and more dexterous [54]. It is, also, referred to as the use of computer vision data to control the motion of a robot in many applications such as robotic assembly [55], unmanned

aerial vehicle [56], robotized endoscope tracking [57], ultrasound probe guidance [58], etc. Typical visual servoing controls can be classified into three categories, position-based visual servoing (PBVS), image-based visual servoing (IBVS) and hybrid visual servoing [54, 59]. The main idea of a visual servoing system is illustrated in Figure 4.1. The system consists of two separate controllers which are Visual Servoing and Robot Controller. The visual servoing block uses a controlling command to generate a velocity screw as the control input for the robotic systems which leads to the desired joint velocity. The robot controller takes the signal produced by the visual servoing block as its desired path and the robot controller drives the robot to follow that path [60, 61].

The classical IBVS uses a set of geometric features such as points, segments or straight lines in image plane as image features [62]. The controller is designed using the inverse (or pseudo-inverse) of the image Jacobian matrix to obtain the control errors in Cartesian space, and normally the proportional control law is applied to achieve a local convergence to the desired visual features. This proportional controller is very easy to implement; however, its drawbacks are its possible unsatisfactory behavior due to the difficulty of constraint handling. Also, the stability of the system is only guaranteed in the region around the desired position, and there may exist image singularities and image local minima, leading to IBVS performance degradation. Moreover, if the errors between the initial position and the desired one are large, with the visibility constraint, the camera motion may be affected by loss of features, or conflict with the robot physical limitations, or even lead to servoing failure. Hence numerous published literatures focus on improving the control performance and overcoming the visibility problem [63, 64].

To solve the problem of image singularities, finding the suitable visual features such as Polar [65], cylindrical [66], spherical [67] coordinate systems and moment features [68], for visual servoing is a good solution. In [69], authors used Takagi-Sugeno fuzzy framework to model IBVS, and they could handle singularity. However, these methods still have not addressed the constraints explicitly which are crucial for real systems control designing [70]. Also some advanced control techniques have been applied in visual servoing controller design [71–74]. A switching controller is proposed in [71] to realize a large displacement grasping task. In [72], a robust fuzzy gain scheduled visual servoing with sampling time uncertainties has been reported. An MPC method based on the discrete time visual servoing model is introduced to obtain the convergence of robot motion by nonlinear constraint optimization in [73]. Another predictive control for constrained IBVS is proposed in [74]. The authors solve

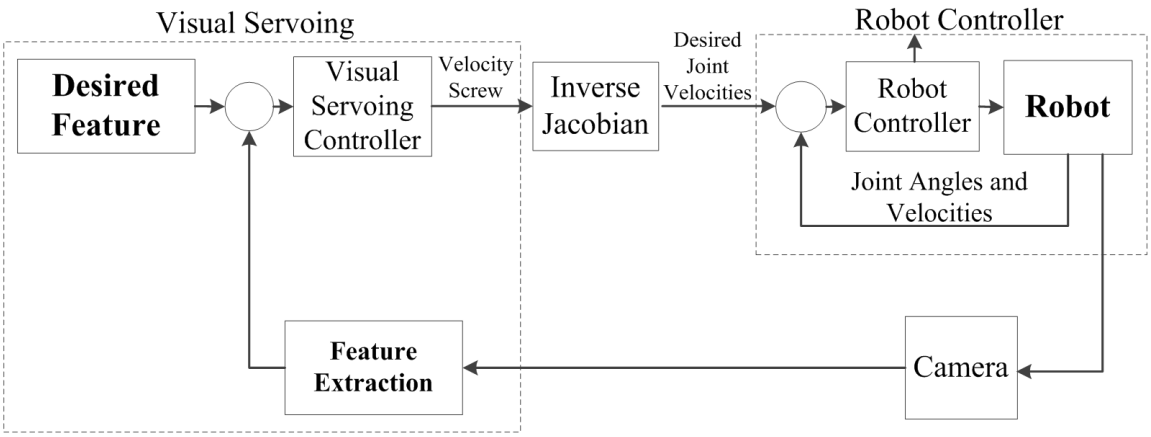


Figure 4.1: General Visual Servoing Approach.

a finite horizon open-loop constrained optimization problem and demonstrate the results in simulation. A robust MPC based on the polytopic model of IBVS has been proposed in [75] with the fixed depth parameter. In [75], the optimization time for calculating the control signal exceeds the real system sampling time. Hence, the proposed controller is not implemented *on-line* and cannot be applied for the real system. To our best knowledge, very few experimental results have been obtained on this topic.

As mentioned, the major drawback of MPC is long computational time required to solve the optimization problem which often exceeds sampling interval in real time situation [13]. In order to make MPC implementable in practice for the fast dynamic systems, the optimization problem must be solved within the time dictated by the sampling period of the system. The major contribution of this chapter is to design an *on-line* RMPC which allows explicit incorporation of plant uncertainties and constraints when system parameters vary over a given range. In this chapter, based on the chosen image features, image points and the discretized model of image Jacobian matrix, a RMPC law is formulated for IBVS. Using the discretized relationship between the time derivative of image features and camera velocity, a discrete time model of the visual servoing system is obtained. In the whole working space, the image Jacobian matrix varies with the bounded variable parameters of image point coordinates and the object depth. Therefore, it is considered as LPV model. A *polytopic* model of a discrete time model for the visual servoing system is obtained using TP model transformation described in Chapter 2, from LPV model. Hence, the robust control signal can be calculated at every sampling time, by performing convex optimization involving LMIs in MPC [24].

Using RMPC law, robot workspace limitations, the visibility constraints, parametric uncertainties and actuator limitations can be easily posed as inequality constraints associated with the output and the input of the visual servoing model. Since the proposed IBVS controller avoids the direct inverse of the image Jacobian matrix, some intractable problems in the classical IBVS controller, such as large displacement from the initial pose to the desired pose of the camera, can be solved by this method. At the same time, the visual features are kept in the image plane even when both the initial and the desired image features are close to the field of view (FOV) boundary. The real time experimental results demonstrate the effectiveness of this method.

The chapter is organized as follows; in Section 4.2, the image-based visual servoing model is established to predict the future behavior of the system. Then a real-time MPC algorithm for IBVS is given in Section 4.3. In Section 4.4, the experiments on an eye-in-hand 6 DOF robot illustrate the effectiveness of the proposed approach. Finally, conclusions are given in Section 4.5.

4.2 LPV Visual Servoing Model

In this work, MPC is used to control the image-based visual servoing system for a robotic system consists of a 6 DOFs manipulator with a camera installed on its end-effector. The target object is assumed to be stationary with respect to robot's reference frame. The constrained finite-horizon optimal controller design is based on the optimization technique in which the current control action is obtained by minimizing the cost function *on-line* [76]. The cost function includes the current measurement, the prediction of the image future states and the current and future control signals based on a discrete time model of the system [77]. The purpose of measuring the states and considering them at each time step is to compensate for unmeasured disturbances and model uncertainty [8].

To control the system using MPC, it needs to find a model whereby the future behavior of the image feature vector can be predicted. The relationship between the time variation of the image feature vector of the predictor, $\dot{\boldsymbol{s}}_m$, and the camera velocity screw, $\mathbf{V}_c$ can be written as [54]

$$\dot{\boldsymbol{s}}_m = \mathbf{L_s}\mathbf{V}_c, \tag{4.1}$$

in which $\mathbf{V}_c$ is the control signal i.e. the camera velocity screw written as follows

$$\mathbf{V}_c = [v_{cx},\, v_{cy},\, v_{cz},\, \omega_{cx},\, \omega_{cy},\, \omega_{cz}]^{\mathrm{T}}, \tag{4.2}$$

also, $\mathbf{L_s} \in \mathbb{R}^{\kappa \times 6}$ is named as the image Jacobian or the interaction matrix.

Suppose the coordinate of a point in 3D space represented in the camera frame is defined as $\mathbf{\Gamma} = (X, Y, Z)$, and projected coordinate of γ on the camera frame is defined as $\gamma = (x, y)$ [78]. Moreover, assuming that the camera projects the 3D geometry with a perspective projection model, $\mathbf{\Gamma}$ is projected to γ by

$$\gamma = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}. \tag{4.3}$$

The image space coordinate can be calculated using the pixel coordinate of the image using the following equation

$$\gamma = \mathbf{B}^{-1}\mathbf{m}, \tag{4.4}$$

where $\mathbf{m} = (u, v)$ is the pixel coordinate of the point γ, and $\mathbf{B}$ is the intrinsic parameter matrix

$$\mathbf{B} = \begin{bmatrix} \alpha_x & \beta & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.5}$$

where α_x and α_y are the scale factors in x and y directions, respectively. u_0 and v_0 are the principal points of the image and β is the skew coefficient between the camera's x and y axis and it is often zero. These parameters are achieved through a camera calibration process [79].

Taking the time derivative of equation (4.3)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} \\ \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} \end{bmatrix}, \tag{4.6}$$

where $\dot{\mathbf{\Gamma}} = (\dot{X}, \dot{Y}, \dot{Z})$ is the velocity of the 3D point in space with respect to camera frame, and $\dot{\gamma} = (\dot{x}, \dot{y})$ is the velocity of the image of $\mathbf{\Gamma}$ in the image space. The following equation can be used to find the relationship between the camera motion and the features

$$\dot{\mathbf{\Gamma}} = -\mathbf{v}_c - \boldsymbol{\omega}_c \times \mathbf{\Gamma}, \tag{4.7}$$

Now by substituting (4.3) and (4.6), the interaction matrix is obtained as follows [54]

$$\mathbf{L}_s = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix}, \tag{4.8}$$

where x and y are the projected coordinate of the feature position on the camera frame, and Z is the depth of the feature with respect to camera frame. Usually the depth parameter Z in the image Jacobian matrix is assumed to be known [80]. However, in the monocular eye-in-hand configuration, it is difficult to measure the depth on-line. Thus, this parameter can be considered as an uncertain variable which varies over a given range. Therefore, all the parameters in the image Jacobian matrix are time varying variables. Thus, $\mathbf{L}_s$ is the function of the vector of time-varying parameters defined as $\mathbf{p}(t) = \{x, y, Z\}$. Here, $\mathbf{p}(t) \in \Omega$ is the element of the closed hypercube $\Omega = [x_m, x_M] \times [y_m, y_M] \times [Z_m, Z_M] \subset \mathbb{R}^3$, in which x_m, x_M, y_m, and y_M, are the minimum and maximum ranges of the image point coordinates, and Z_m, and Z_M are the minimum and maximum depths between the object and the camera, respectively.

In order to apply the controller in real–time, the accuracy must be traded with the computational load at each sampling time. Therefore, instead of using the given LPV model, its TP model is used for control design [33]. For the considered time–varying parameter vector, $\mathbf{p}(t)$, a convex combination for the polytopic vertex form of the image Jacobian matrix can be obtained for the LMI–based RMPC controller design.

As explained in Chapter 2, the first step of obtaining TP model is to discretize a given LPV model over the transformation space Ω which means that the resulting TP model is only explicable in Ω [33]. To apply TP transformation on LPV model, an N-dimensional closed hypercube Ω is generated using HSS method. Each generated sample is confined to be in the following constraint

$$a_n \leq g_{n,m_n} \leq b_n, \qquad n = 1, \ldots, N,\ m_n = 1, \ldots, M_n \tag{4.9}$$

where, N is the total number of the time varying parameters in the image Jacobian matrix or the dimension of Ω, which is equal to 3. M_n is the number of samples on n^{th} dimension. Also, a_n, and b_n are the minimum and maximum of the closed hypercube elements on each

dimension, respectively, and are given as follows

$$a_1 = x_m, \quad a_2 = y_m, \quad a_3 = Z_m,$$
$$b_1 = x_M, \quad b_2 = y_M, \quad b_3 = Z_M.$$

Then the discretization of LPV model, $\mathbf{B}_d(\mathbf{q}(t))$, is obtained by sampling over the grid points in Ω as follows

$$\mathcal{L}_{m_1,m_2,m_3} = \mathbf{L}_s(\mathbf{g}_{m_1,m_2,m_3}), \tag{4.10}$$

where $\mathcal{L}_{m_1,m_2,m_3}$ is the element of the tensor $\mathcal{L}^{\mathcal{D}}$ (superscript "$\mathcal{D}$" denotes "discretized") with the size equal $M_1 \times M_2 \times M_3 \times 2 \times 6$, that is, $M_1 \times M_2 \times M_3$ different image Jacobian matrices are obtained within the domain of Ω, and each of which represents an image Jacobian matrix at a specific time. Because at each sampling time, the system has different values of x, y, and Z, but they belong to transformation space Ω.

By applying HOSVD, the corresponding image Jacobian matrix becomes

$$\mathcal{L}^{\mathcal{D}} = \sum_{m_1=1}^{M_1} \sum_{m_2=2}^{M_2} \sum_{m_3=1}^{M_3} \prod_{n=1}^{3} w_{n,m_n}(q_n) \mathcal{L}_{m_1,m_2,m_3}, \tag{4.11}$$

where, $w_{n,m_n}(q_n)$ is the weighting function value evaluated at the discretized values of $p_n = g_{n,m_n}$ over the n-dimension interval $[a_n, b_n]$. Based on (4.11), the $(N+2)$-dimensional coefficient tensor $\mathcal{L}^{\mathcal{D}} \in \Re^{M_1 \times M_2 \times M_3 \times 2 \times 6}$ is constructed from linear time invariant (LTI) vertex systems $\mathcal{L}_{m_1,m_2,m_3} \in \Re^{2 \times 6}$.

In order to have convex TP model, the weighting function for all $\mathbf{p} \in \Omega$ should satisfy the following conditions

$$\forall n,\, m,\, q_n \; w_{n,m}(q_n) \geq 0, \tag{4.12}$$

$$\forall n,\, q_n \; \sum_{m_n=1}^{M_n} w_{n,m}(q_n) = 1. \tag{4.13}$$

The conditions (4.12) and (4.13) imply that the TP model type is Non-Negativeness (NN) and Sum-Normalization (SN), respectively.

By discarding some nonzero singular values and using Reduced HOSVD technique, the computational load of the control design will be reduced. The error between the exact tensor, $\mathcal{L}^{\mathcal{D}}$, and the reduced one, $\hat{\mathcal{L}}^{\mathcal{D}}$, can be approximated by (2.18). Then the extracted reduced TP model can be used to design the MPC.

4.3 Controller Design

To design the MPC, the discrete time model is used instead of the continuous time dynamic model given in (4.1). The discrete time state-space model of each feature point can be expressed as

$$\boldsymbol{s}_m(t+1) = \mathbf{I}\boldsymbol{s}_m(t) + \mathbf{L}\mathbf{V}_c(t), \tag{4.14}$$

where, $\boldsymbol{s}_m = [x_m, y_m]^T$ is the projected position of each feature point on the camera frame. Also, $\mathbf{L} = h\mathbf{L}_s(t)$ is the discrete time image Jacobian matrix, and h is the sampling time. In (4.14), $\mathbf{I}$ is 2×2 identity matrix which is fixed, and only matrix $\mathbf{L}$ is the function of time-varying parameters.

It is well known that a unique camera pose can theoretically be obtained by using at least four image points; hence, $m = 1, \ldots, 4$.

To design MPC, the image features error at sample time k is defined as $\boldsymbol{e}(k) = \boldsymbol{s}(k) - \boldsymbol{s}_d$, in which $\boldsymbol{s} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_4]^T$. Also, $\boldsymbol{s}_d$ is the desired feature vector acquired from the image of the stationary object taken in the robot target pose.

The underlying goal of designing RMPC is to find a control law for the system input, $\mathbf{V}_c$, so that each image features error, $\boldsymbol{e}(k)$ defined at sampling time k can be steered to zero within a desirable time period.

The control signal is defined as linear state feedback $\mathbf{V}_{c,k} = F_k\boldsymbol{e}_k$ that minimize an upper bound of the worst-case finite horizon quadratic cost at sampling time k

$$\underset{u}{\text{minimize}} \max_{Z \in \mathscr{Z}} \sum_{i=0}^{N-1} \boldsymbol{e}_{k+i|k}^T Q \boldsymbol{e}_{k+i|k} + \mathbf{V}_{c,k+i|k}^T R \mathbf{V}_{c,k+i|k}, \tag{4.15}$$

where, $Q \succ 0$ and $R \succ 0$ are weighting matrices which let the designer make a trade-off between small control signal (big value of R) and fast response (big value of Q). The Lyapunov function $V(\boldsymbol{e}) = \boldsymbol{e}_k^T P_k \boldsymbol{e}_k$ with $P_k \succ 0$ defined at sampling time k is an upper bound on the worst-case cost if it holds for all vertices that satisfy the following inequality [11]

$$\boldsymbol{e}_{k+1|k}^T P_k \boldsymbol{e}_{k+1|k} - \boldsymbol{e}_{k|k}^T P_k \boldsymbol{e}_{k|k} \leq -\boldsymbol{e}_{k|k}^T Q \boldsymbol{e}_{k|k} - \mathbf{V}_{c,k|k}^T \mathcal{R} \mathbf{V}_{c,k|k}. \tag{4.16}$$

It can be seen that by summing up the left–hand and right–hand side of the above inequality from 0 to ∞, and inserting a linear feedback $u_k = F_k\boldsymbol{e}_k$, a matrix inequality can

be obtained as follows

$$(\mathbf{I} + \mathbf{L}_k F)^T P_k (\mathbf{I} + \mathbf{L}_k F) - P_k \preceq -Q - F_k^T R F_k, \tag{4.17}$$

where, $\mathbf{L}_k$ is the discrete time Jacobian matrix at sampling time k. According to Boyd et al. [16], by applying a congruence transformation $P_k = X_k^{-1}$, defining $Y_k = F_k X_k$, and an LMI in terms of X_k and Y_k can be found using a Schur complement as follows

$$\begin{pmatrix} X_k & * & * & * \\ X_k + \mathbf{L}_k Y_k & X_k & * & * \\ X_k & \mathbf{0} & Q^{-1} & * \\ Y_k & \mathbf{0} & \mathbf{0} & R^{-1} \end{pmatrix} \succeq 0, \tag{4.18}$$

where, the symbol $*$ represents a symmetric structure in LMIs. This LMI is only valid for one model, but for this system which is a TP model, LMI should hold for all possible model or vertices.

In order to solve the LMI in *real-time* and consider all of the possible vertices, the control signal at time k is defined as $u_{k+i|k} = F_k \boldsymbol{e}_{k+i|k}$ used for the future as well. The control signal is obtained by minimizing the upper bound on the worst–case value of the quadratic objective function considered as $\gamma_k = \boldsymbol{e}_{k|k}^T P_k \boldsymbol{e}_{k|k}$, where $X_k = \gamma_k P_k \succ 0$ and Y_k are obtained from the solution of the following semi-definite program [51]

$$\underset{\gamma_k, X_k, Y_k}{\text{minimize}}\, \gamma_k \tag{4.19}$$

subject to

$$\begin{pmatrix} X_k & * & * & * \\ X_k + \mathbf{L}_d^j Y_k & X_k & * & * \\ X_k & \mathbf{0} & \gamma_k \mathcal{Q}^{-1} & * \\ Y_k & \mathbf{0} & \mathbf{0} & \gamma_k \mathcal{R}^{-1} \end{pmatrix} \succeq 0, \tag{4.20}$$

$$\begin{pmatrix} \mathbf{1} & x_{k|k}^T \\ x_{k|k} & X_k \end{pmatrix} \succeq 0, \tag{4.21}$$

where, $j = 1, 2, \ldots, L$ (L is the number vertices). Also, constraints on input and output,

to ensure that the constraints are satisfied, can be defined as follows [81]

$$\begin{pmatrix} v_{max}^2\mathbf{I} & Y_k \\ Y_k^T & X_k \end{pmatrix} \succeq 0, \tag{4.22}$$

$$\begin{pmatrix} y_{max}^2\mathbf{I} & X_k + \mathbf{L}_d^j Y_k \\ (X_k + \mathbf{L}_d^j Y_k)^T & X_k \end{pmatrix} \succeq 0, \tag{4.23}$$

where, v_{max}, and y_{max} are the upper bound for input and output, respectively.

Under the above designed closed-loop feedback law, the solution for the optimization in (4.19) can be obtained using the LMI technique, which results in stabilizing the LPV system and steering the state variables to zero. At each sampling time, an optimal upper bound on the worst-case performance cost over the infinite horizon is obtained by forcing a quadratic function of the state to decrease by at least the amount of the worst case performance cost at each prediction time. Such on-line step–by-step optimization can lead to asymptotically stable evolution.

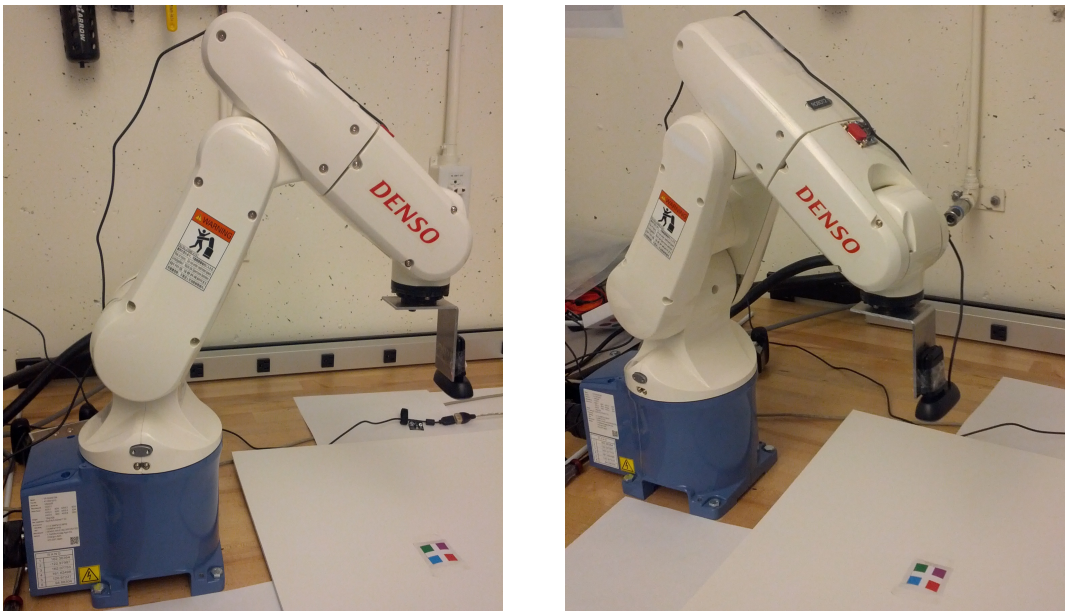
Here, YALMIP toolbox, which is used for modeling and solving the convex and nonconvex optimization problems [24], is utilized to implement model predictive controller. This toolbox helps to accomplish the on-line optimization and obtain the control signal at each sampling time.

4.4 Experimental Results

In this section, the proposed controller is tested on a 6 DOF Denso robot [82]. The experimental setup consists of a VS-6556G Denso robot and a camera mounted on the end-effectors (Figure 4.2).

The robot communicates with its controller with a frequency of $1\,kHz$. A CCD camera is used as the vision system, and is mounted on the robot's end-effector. The camera characteristics is given in Table 4.1. The program has been run using MATLAB 2014b/Simulink Real-Time Workshop on PC Intel (R) Core (TM) i7-4770 CPU 3.4 GHz in Microsoft Windows 7 operating system.

The camera capturing rate is 30 frames per second. The object is stationary in the working space. The visual servoing task is completed when the image features match the desired features. In this work, four different tests with different strategies have been performed to



(a) (b)

Figure 4.2: Experimental Setup.

Table 4.1: Camera Parameters

Parameters	Values
Focal length	0.004 (m)
X axis scaling factor	110000 (pixel/m)
Y axis scaling factor	110000 (pixel/m)
Image plane offset of X axis	120 (pixel)
Image plane offset of Y axis	187 (pixel)

validate the algorithms.

4.4.1 TP Model Transformation

The hyper-rectangular N-dimensional space grid is generated, then the TP model transformation is used to find the discretization model. 100 samples are considered on each dimension for discretization. Therefore, a $100 \times 100 \times 100 \times 2 \times 6$ tensor of the system is obtained. The value of the parameters in Ω are considered as $x_m = -0.4\,m$, $x_M = 0.4\,m$, $y_m = -0.4\,m$, $y_M = 0.4\,m$, $Z_m = 0.2\,m$, and $Z_M = 0.6\,m$. After applying HOSVD on each 3–dimension of the system tensor, the nonzero singular values in each dimension are obtained

as follows

$$\begin{array}{lll} \sigma_1^{(1)} = 51.15 & \sigma_1^{(2)} = 51.15 & \sigma_1^{(3)} = 53.02 \\ \sigma_2^{(1)} = 7.23 & \sigma_2^{(2)} = 7.23 & \sigma_2^{(3)} = 4.00 \\ \sigma_3^{(1)} = 0.13 & \sigma_3^{(2)} = 0.13 & \end{array}$$

In this case the small singular values which are possible to discard them are $\sigma_3^{(1)}$, and $\sigma_3^{(2)}$. So, two first singular values of the first and second dimensions and all the nonzero singular values of the third dimension are kept, and the error between the full rank of tensor $\mathcal{L}^{\mathcal{D}}$ with 18 LTI models and the reduced one, $\hat{\mathcal{L}}^{\mathcal{D}}$ with 8 LTI models can be approximated by

$$\|\mathcal{L}^{\mathcal{D}} - \hat{\mathcal{L}}^{\mathcal{D}}\|^2 \leq (\sigma_3^{(1)})^2 + (\sigma_3^{(2)})^2 \approx 0.034. \tag{4.24}$$

Therefore, the results show that the system in (4.14) can be approximately given in the HOSVD-based canonical polytopic model form with minimum $2 \times 2 \times 2 = 8$ linear time invariant (LTI) vertex models. In order to have the convex TP model which can satisfy LMI control design conditions, the sum normalization (SN) and non-negativeness (NN) type weighting functions are used [33]. The weighting functions for exact tensor and reduced one are illustrated in Figure 4.3.

The LTI system matrices of the polytopic TP model are

$$\mathbf{L}_{1,1,1} = \begin{pmatrix} 0.08 & 0 & 0.03 & 0.01 & 0.59 & -0.01 \\ 0 & 0.08 & 0.03 & -0.59 & -0.01 & 0.01 \end{pmatrix},$$

$$\mathbf{L}_{1,1,2} = \begin{pmatrix} -0.13 & 0 & -0.05 & 0.01 & 0.59 & -0.01 \\ 0 & -0.13 & -0.05 & -0.59 & -0.01 & 0.01 \end{pmatrix},$$

$$\mathbf{L}_{1,2,1} = \begin{pmatrix} 0.08 & 0 & 0.03 & -0.04 & 0.59 & 0.09 \\ 0 & 0.08 & -0.20 & 0.40 & 0.04 & 0.01 \end{pmatrix},$$

$$\mathbf{L}_{1,2,2} = \begin{pmatrix} -0.13 & 0 & -0.05 & -0.04 & 0.59 & 0.09 \\ 0 & -0.13 & 0.33 & 0.40 & 0.04 & 0.01 \end{pmatrix},$$

$$\mathbf{L}_{2,1,1} = \begin{pmatrix} 0.08 & 0 & -0.20 & -0.04 & -0.40 & -0.01 \\ 0 & 0.08 & 0.03 & -0.59 & 0.04 & -0.09 \end{pmatrix},$$

$$\mathbf{L}_{2,1,2} = \begin{pmatrix} -0.13 & 0 & 0.33 & -0.04 & -0.40 & -0.01 \\ 0 & -0.13 & -0.05 & -0.59 & 0.04 & -0.09 \end{pmatrix},$$

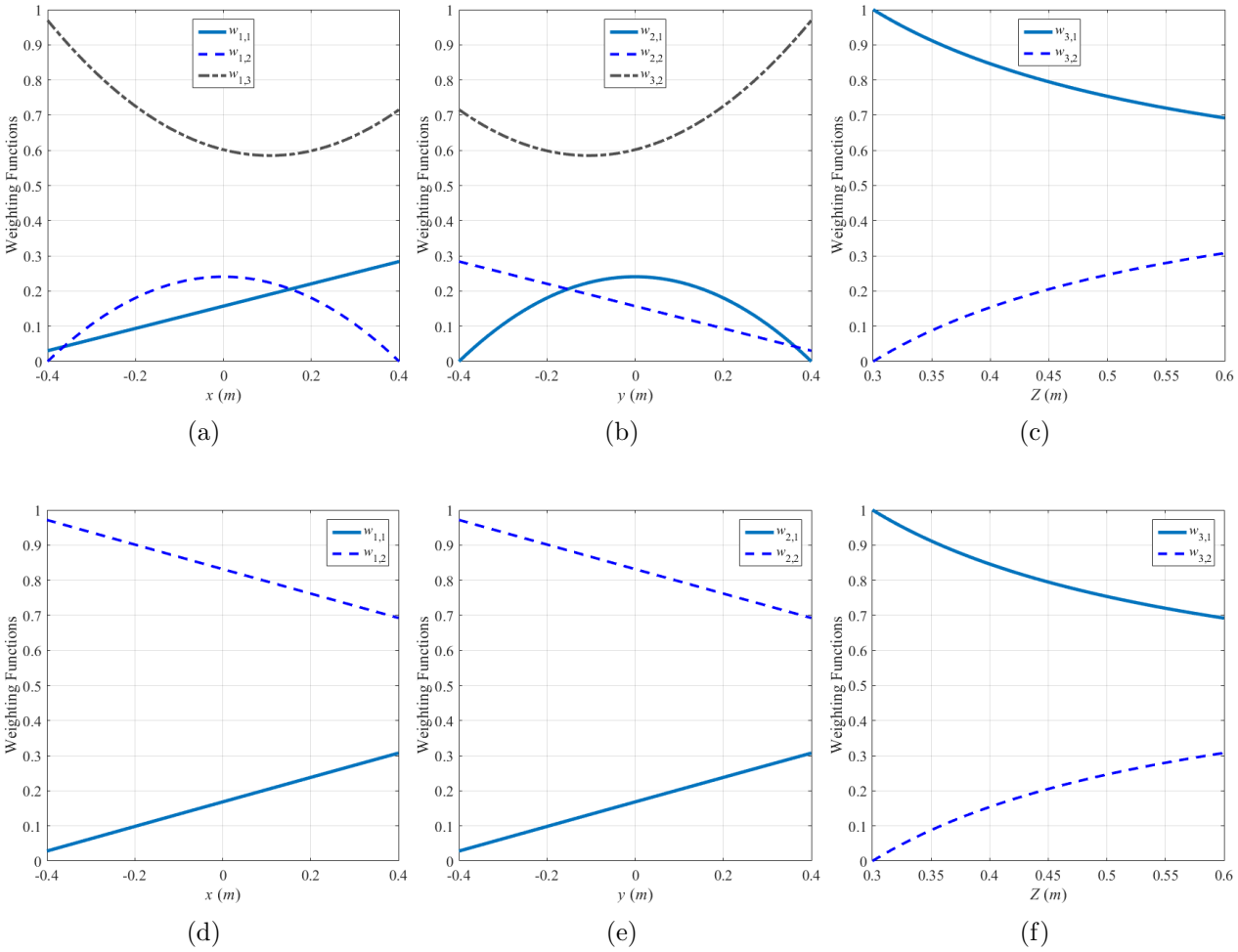


Figure 4.3: SNNN type weighting functions of (a-c) the exact TP model of 18 LTI systems, and (d-f) the reduced TP model of 8 LTI systems.

$$\mathbf{L}_{2,2,1} = \begin{pmatrix} 0.08 & 0 & -0.20 & 0.24 & -0.40 & 0.09 \\ 0 & 0.08 & -0.20 & 0.40 & -0.24 & -0.09 \end{pmatrix},$$

$$\mathbf{L}_{2,2,2} = \begin{pmatrix} -0.13 & 0 & 0.33 & 0.24 & -0.40 & 0.09 \\ 0 & -0.13 & 0.33 & 0.40 & -0.24 & -0.09 \end{pmatrix}.$$

The image Jacobian matrix or interaction matrix, $\mathbf{L}$, at each sampling time can be written as follows

$$\mathbf{L}(t) = \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} w_{1,i}(x(t)) w_{2,j}(y(t)) w_{3,k}(Z(t)) \mathbf{L}_{d_{i,j,k}}, \tag{4.25}$$

where $w_{1,i}$, $i = 1, 2$, $w_{2,j}$, $j = 1, 2$ and $w_{3,k}$, $k = 1, 2$ are weighting functions which are shown in Figures 4.3d–4.3f.

The interaction matrix (4.25) varies in a polytope Ω (Convex Hull) of vertices which satisfies convexity conditions given in Equations (4.12), and (4.13).

4.4.2 Results and Analysis

The maximum control input of camera velocity screw in Equation (4.22), v_{max}, is limited to $0.25\, m/s$ for the translational speed and $0.25\, rad/s$ for the rotational speed.

Using the proposed method and YALMIP toolbox, the experiments are performed in real-time and the computational time of the optimization problem is less than the sampling time ($0.04\, sec$). There is scant research on developing real-time MPC technique for IBVS control. For example, both [75] and [83] only have presented the simulation results. However, in this work, all experiments have carried out in real-time.

The bigger elements of the matrix $\mathcal{Q}$ are chosen in comparison with the ones of the matrix $\mathcal{R}$ to have the fast convergence response

$$Q = 10 \times \mathbf{I}_{8\times 8}, \quad R = \mathbf{I}_{6\times 6}, \tag{4.26}$$

where $\mathcal{I}_{n\times n}$ is the identity matrix. The experimental results of the four different cases are given in the following.

Test 1:

In the first test the initial and desired features are chosen in a way that a 90 degrees rotation about the camera's center is required to complete the task. The initial and desired locations

Table 4.2: Initial (I) and Desired (D) locations of feature points in pixel

		Point1		Point2		Point3		Point4	
		(x	y)	(x	y)	(x	y)	(x	y)
Test 1	I	121	100	161	101	160	138	120	137
	D	152	159	152	119	188	121	187	160
Test 2	I	279	153	308	179	282	207	254	179
	D	37	63	77	52	87	88	47	99
Test 3	I	36	176	71	176	69	210	32	209
	D	260	207	278	179	307	197	288	226
Test 4	I	152	36	230	84	177	162	105	111
	D	119	110	162	110	161	150	117	150

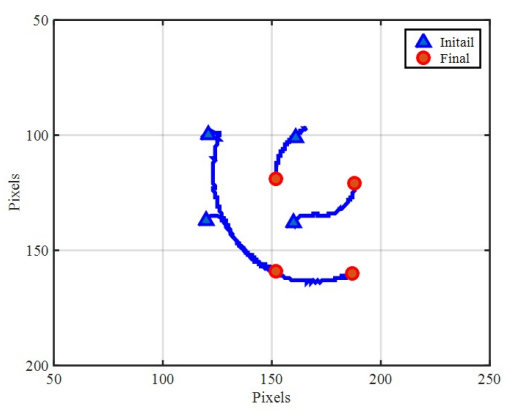
of the features are given in Table 4.2. The results of this test are given in Figure 4.4.

Figure 4.4a shows the trajectory of the features in image plane. The trajectory starts from the initial position indicated as the triangle sign and ends at the positions indicated as the circle sign. This figure shows how the controller takes the features to their desired value without any unnecessary motions. A similar test was performed in [60]. The comparison between two results shows that considering the constraints in the controller could improve the trajectory of the features in image plane. The joint angles' changes during the visual servoing task are shown in Figure 4.4c. Finally, the 3D trajectory of the robot end-effector in space is shown in Figure 4.4b.

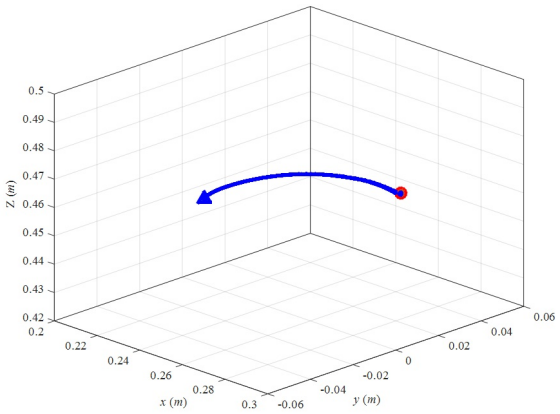
Test 2:

In the second test, a long distance visual servoing task is performed. The initial and the desired locations of the features are located in relatively far distance from each other as shown in Table 4.2. The results of this test are given in Figure 4.5.

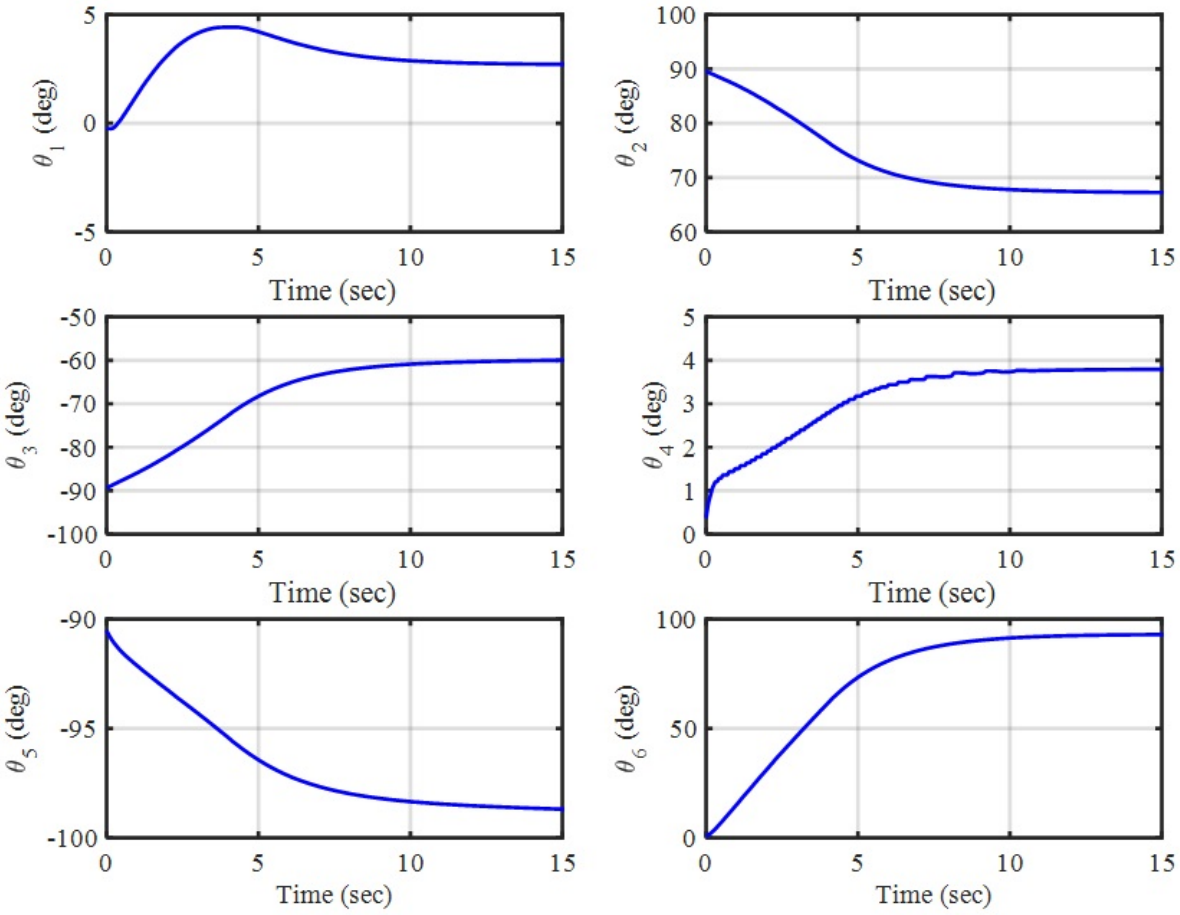
One of the drawbacks of the IBVS controller is that it can not keep the end effector inside the workspace when a long distance task is required. The MPC controller could provide better results for such tasks because of its prediction algorithm. Thus, the MPC controller prevents reaching the limits of the space during the operation. The results of Test 2 show how the MPC controller succeeds in completing the task. The sequence of the result figures is the same as the one in Test 1.



(a) Feature trajectory in image plane

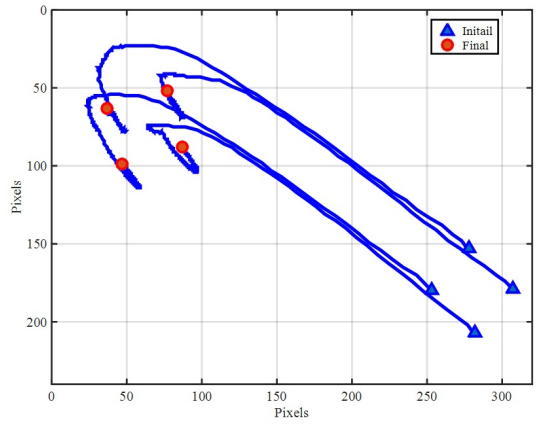


(b) Camera 3D trajectory

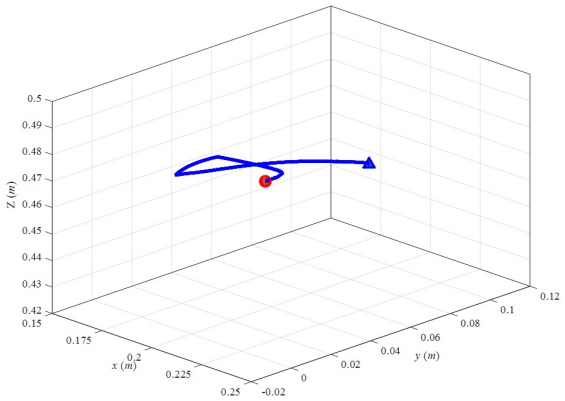


(c) Robot joint angles

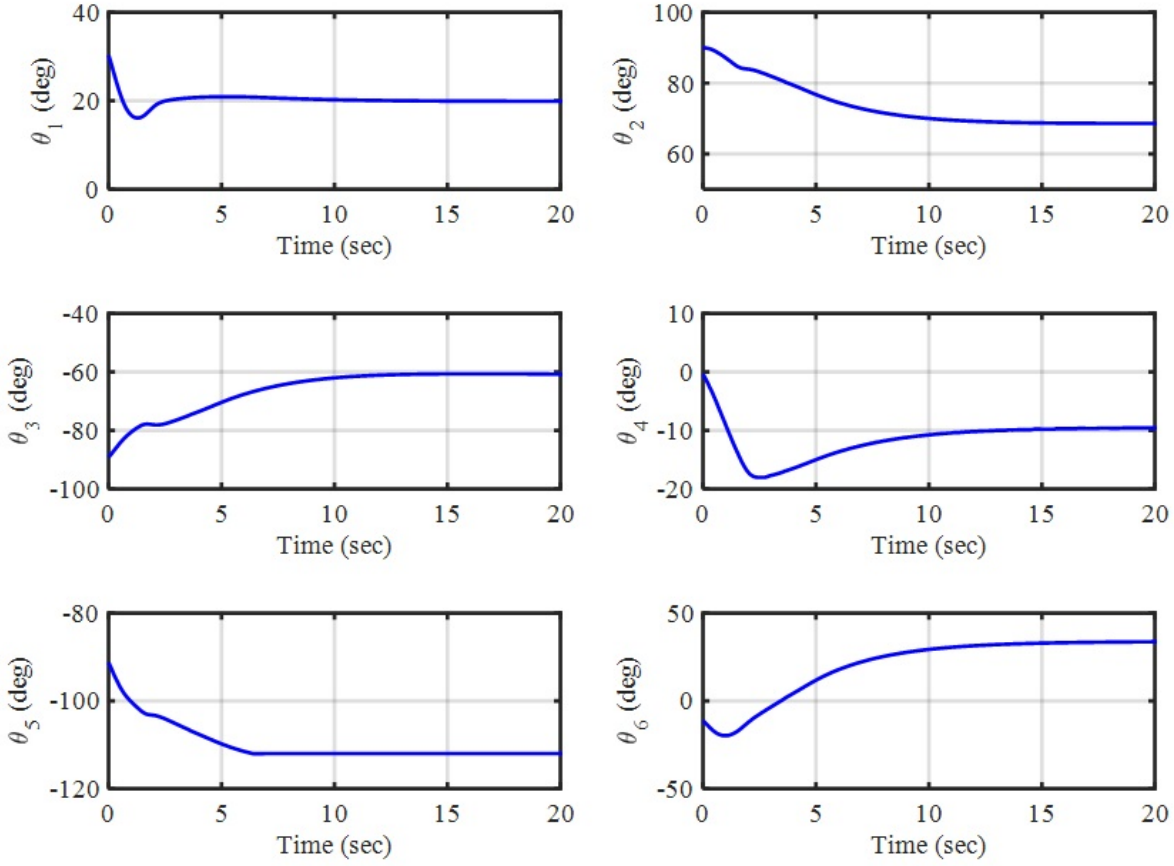
Figure 4.4: Results for Test 1.



(a) Feature trajectory in image plane



(b) Camera 3D trajectory



(c) Robot joint angles

Figure 4.5: Results for Test 2.

Test 3:

In Test 3, another long distance task is tested where the features move close to the FOV limit. The initial and desired locations of the features are given in Table 4.2. Performing the same test using an IBVS controller causes the features to leave the field of view [60]. The IBVS controller rotates the features while taking the features to the desired position. However, the proposed MPC controller avoids rotating the features in order to respect the system constraints. The rotation of the features happens when the features are close to the desired features. The results of this test are given in Figure 4.6.

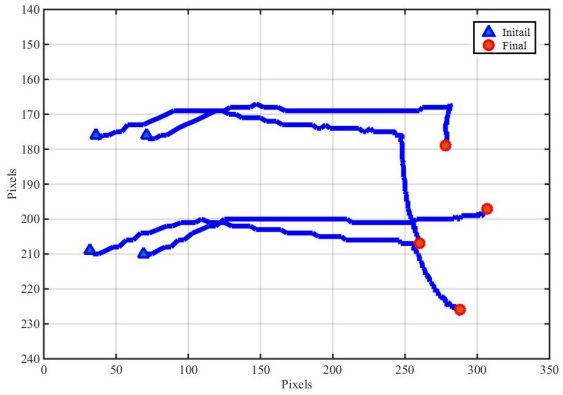
Test 4:

In Test 4, a visual servoing task is prepared which requires a complicated motion in 3D space and involves all the 6 DOFs motions in space. The initial and desired locations of the features are given in Table 4.2. The results for this test are given in Figure 4.7. The results show how the proposed MPC controller manages to reach the desired target while keeping the image features and the robot within limits.

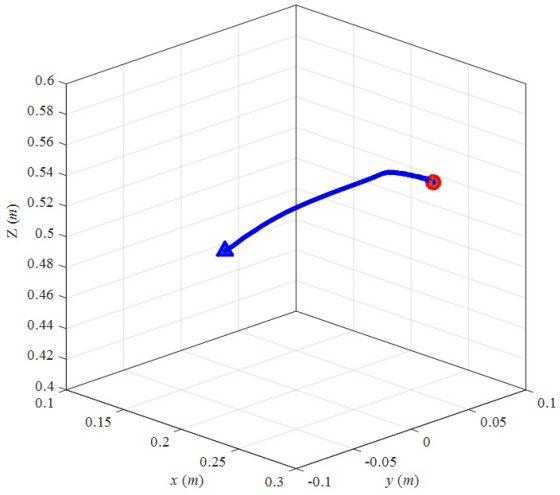
According to the obtained results, it is obvious that for different tests, the camera has different translational movement in Z direction from initial pose to the desired or final pose. Figure 4.7b shows that camera moves from $Z = 0.25\,m$ to $Z = 0.4\,m$ which is a long vertical translation. In most of the researches such as [60], for simplification purpose, a constant depth value is considered as the depth of the object with respect to the camera. This assumption can affect the performance of the controller unless the controller is designed with the robustness against the uncertainty. Therefore, by using robust optimization method in which Z is considered as an uncertain bounded variable, the robust MPC can be designed. The robust MPC can deal with the time varying depth of object. The results of Test 4 demonstrate that it is far better to consider the variable depth instead of the fixed one.

4.5 Conclusion

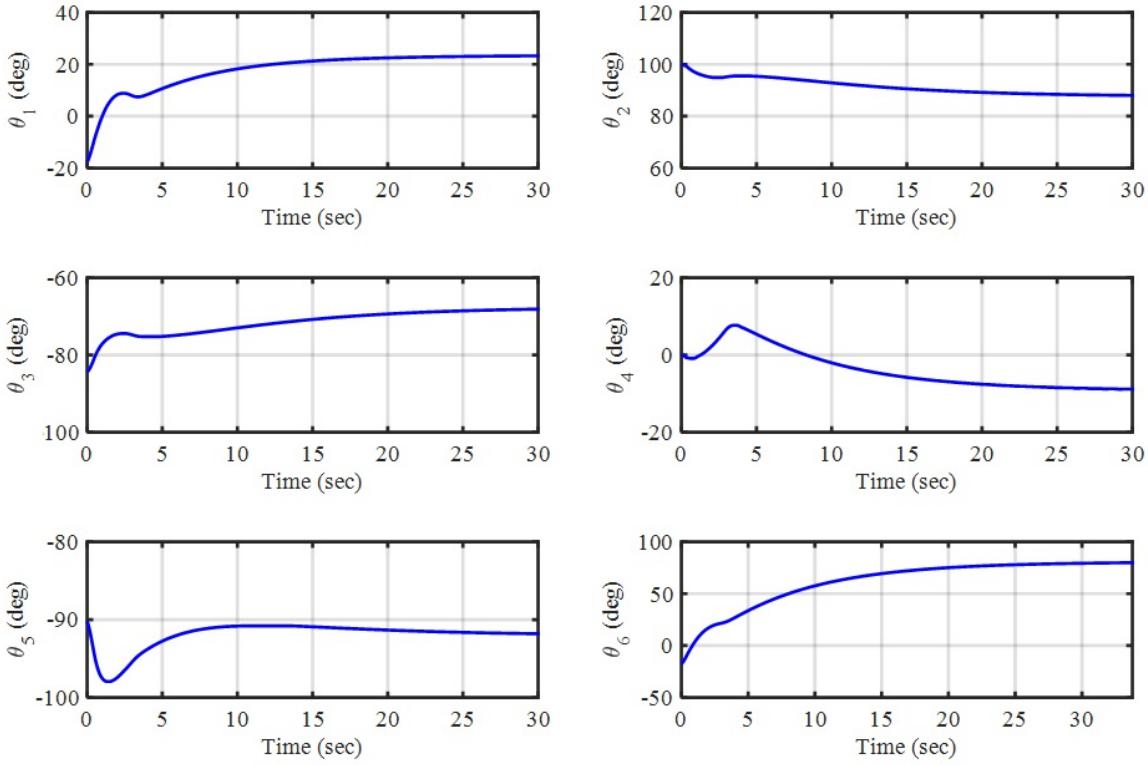
In this chapter a real-time MPC-based IBVS controller is developed based on the discretized model of image Jacobian matrix. The control signal is obtained by minimizing the cost function based on the error in image plane and provides the stability and convergence of the robot motion. The constraints due to actuator limitations and visibility constraints can



(a) Feature trajectory in image plane

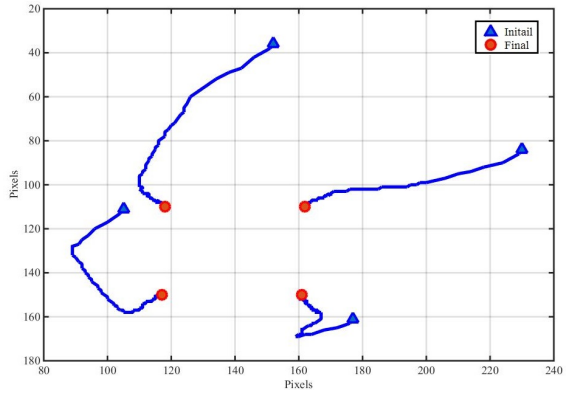


(b) Camera 3D trajectory

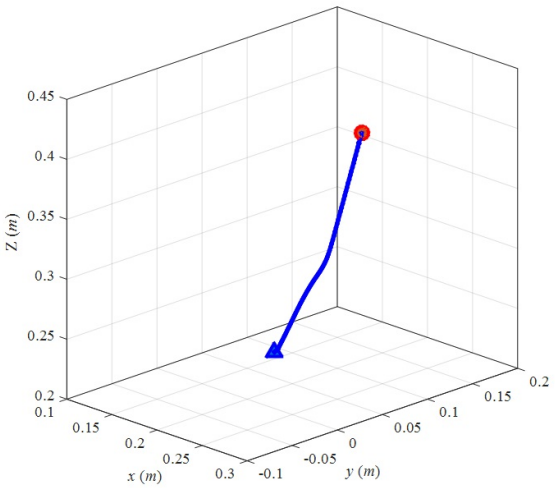


(c) Robot joint angles

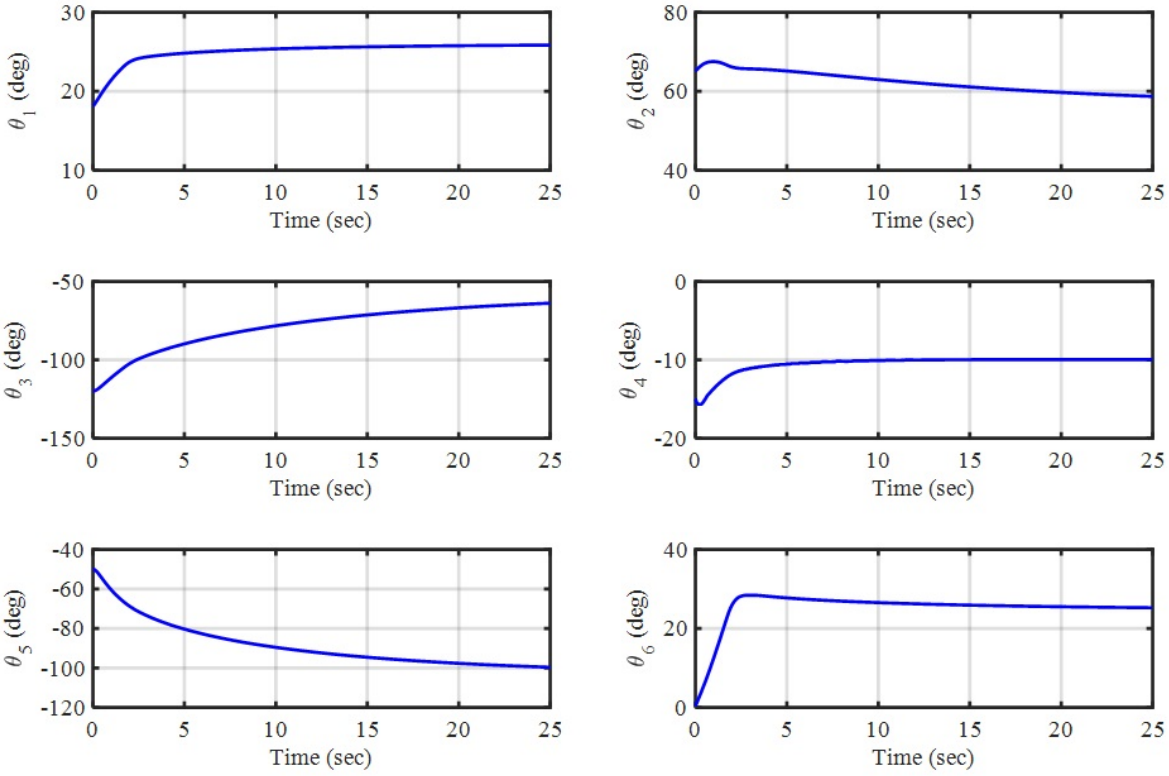
Figure 4.6: Results for Test 3.



(a) Feature trajectory in image plane



(b) Camera 3D trajectory



(c) Robot joint angles

Figure 4.7: Results for Test 4.

be taken into account using model predictive control strategy. The experimental results on a 6-DOF eye-in-hand visual servoing system have demonstrated the effectiveness of the proposed method. The experiments have been carried out in a true real–time fashion. The ability of MPC to keep the system within the desired limits increase the success chance of visual servoing tasks compared to basic visual servoing controllers.

Chapter 5

Multi-Objective Model Predictive Control

In the standard MPC formulations, there is a single objective function which normally is the summation of weighted quadratic functions. Although several control specifications which are often irreconcilable can be considered in the single objective function, choosing the appropriate weighting functions are another challenge faced by control designers. This chapter proposes a novel MPC scheme based on multi-objective optimization. At each sampling time, the MPC control action is chosen among a set of optimal solutions based on the Nash bargaining solution. The main contribution is to formulate the standard MPC optimization problem as a multi-objective optimization problem. Furthermore, as the second contribution, it is presented that the optimal performance of the proposed multi-objective MPC scheme is close to the global optimal solution. The stability and controller design are projected as LMIs. It is shown through the examples that the proposed method can execute approvingly compared to other methods in the literature of the control systems.

5.1 Introduction

Multi-objective optimization design recently has attracted great attention of the researchers in solving engineering problems that have conflicting objectives [84, 85]. For example, there always exist more than one objective functions in practical control problem which should be optimized simultaneously (e.g., rise time, overshoot, control efforts, ...) [84]. The control design objectives are normally in conflict with each other. Since the conflicts exist

among the objective functions, and in order to find the trade-offs between the objectives, the controller design can be formulated as a multi-objective optimization problem (MOP) [25, 84, 86]. Hence, a control system design which is associated with multiple and often conflicting performance criteria can be defined as multi-objective control problems [87, 88].

In most of the optimal control system design, such as MPC design, a cost function consists of the weighted quadratic sum of those non-commensurable objectives. Choosing the appropriate weighting factors is inherently difficult and could be regarded as a subjective design parameter [84, 89]. Furthermore, the possible existing trade-off between the objective functions cannot be explored; therefore, it is not possible to find an optimum design point which can reflect the compromise of the designer's choice. Consequently, it is far better to formulate the multi-objective control problems as MOP.

The multi-objective controller design problem can be solved by LMI formulation efficiently, especially, when the objective being traded off are in the 2-norm forms (convex) [25]. Still, there is scant research on developing MPC scheme based on multi-objective optimization [90–92]. The problem of this approach is that in MPC design at each sampling time only one optimal solution is needed. However, in multi-objective optimization, a set of non-dominated solutions are obtained which is called *Pareto Frontier* [93]. Furthermore, the computational time required to solve the optimization problem for MPC with single objective is usually large, let alone multiple objectives. For example, Garcia *et al.* [92] used genetic algorithm to solve the optimization problem at each sampling time. But genetic algorithm needs a large computational time. In order to confine the optimization time within the sampling time, they reduced the initial population size and the number of generations. Therefore, finding the global minimum solution is not possible, and the proposed method is a kind of suboptimal strategy. Then, a fuzzy inference system was used to select intelligently a trade-off point among Pareto fronts. Indeed, this is an additional design objective, because different fuzzy system should be designed for different problems in advance.

In [90], authors used convex optimization techniques to design multi-objective MPC for linear systems. They used the convex combination of the objective functions to generate Pareto front points. They used a pre-defined target weight vector to find the convex combination of the objective functions. In order to choose the element of this vector which are the function of system states, the control designer must be familiar with the system physical limitation, also, the working properties of the system. Otherwise, it is really hard to choose an appropriate function for the target weight vector. Therefore, appropriate selection

method which is a general and an automatic one, regardless of the system physical property, has not been provided in literatures yet.

In this chapter, a novel MPC scheme based on the multi-objective optimization is proposed in which at each sampling time, the MPC control action is chosen automatically among the set of Pareto optimal solutions based on the *Nash Bargaining Solution* from Game Theory [26]. This method is independent of the system type. It is applied on the nonlinear systems along with TP transformation to design multi-objective MPC. As a result, LMIs and convex optimization techniques can be utilized to provide an on-line solution for the multi-objective MPC design. Finally, the proposed method is executed on a complex nonlinear system and the obtained results are compared with the proposed method presented in [21].

The rest of this chapter is organized as follows. Section 5.2 is dedicated to preliminary notions of multi-objective optimization. The problem statement is presented in 5.3. In Section 5.4, the proposed method and control design procedure are introduced. Numerical examples and simulation results are presented in Sections 5.5 and 5.6, followed by the concluding remark in Section 5.7.6

5.2 Multi-Objective Optimization

Multi-objective optimization also called multi-criteria optimization or vector optimization is defined as finding of the decision variables vector satisfying constraints to give optimal values to all objective functions [84]. Without loss of generality, assume all objective functions must be minimized, then a MOP can be defined as follows

$$\underset{\nu}{\text{minimize}} \quad \mathbf{J}(\nu) = [J_1(\nu),\, J_2(\nu), \ldots,\, J_l(\nu)] \tag{5.1a}$$

$$\text{subject to}$$

$$\mathcal{G}_i(\nu) \leq 0, \quad i = 1,\, \ldots,\, m, \tag{5.1b}$$

$$\mathcal{H}_j(\nu) = 0, \quad i = 1,\, \ldots,\, p, \tag{5.1c}$$

where, $\nu \in \mathbb{R}^n$ is the decision vector or design variables, $\mathbf{J}(\nu) \in \mathbb{R}^l$ is the objective functions vector. Also, the solution of the optimization in (5.1a), ν^*, must satisfy both inequality constraints in (5.1b) and equality constraints in (5.1c).

The followings are some definition of multi-objective minimization based on the Pareto approach [94].

Definition 5.2.1 (Pareto Dominance)
A vector $\bar{\mathbf{J}} \in \mathbb{R}^l$ dominates a vector $\mathbf{J} \in \mathbb{R}^l$ if and only if the set of inequalities

$$\bar{J}_i \leq J_i, \quad i = 1, \ldots, l,$$

and at least one of the inequalities is strict, that is, $\exists j \in 1, 2, \ldots, l : \bar{J}_j < J_j$.

Definition 5.2.2 (Pareto Optimality)
A vector ν^* is said to be a Pareto Optimal (minimal) if and only if $J_i(\nu^*) < J_i(\nu), \forall i$. In other words, no other solution can be found to dominate ν^* based on the Pareto dominance definition.

Definition 5.2.3 (Pareto Set)
Pareto set $\mathcal{P}$ is a set of decision variables including all of the possible Pareto optimal vectors. It means that any point in Pareto set dominates all of the points out of this set.

Definition 5.2.4 (Pareto Front)
Pareto front $\mathcal{P}_f$ is a set of objective functions vector obtained by the design variables vectors in the Pareto set $\mathcal{P}$.

In a Pareto front, typically there are always more than one solutions. Any solution in Pareto front is called non-dominated solution, because it is never dominated by another solution in Pareto front.

As in many practical optimization problems (MPC design), only one solution (one control action at each sampling time) among the Pareto set should be selected as the final solution, it is crucial to find a practical and generalized method to choose a trade-off point among Pareto front. As mentioned before, the proposed methods in [90] and [92] are not the general methods, because the designers must choose different criteria for different systems. Nevertheless, in this thesis, a method will be proposed for any kind of system without any limitation.

To begin with, some properties of the Pareto front will be given as follows. First of all, Pareto front is a convex set, that is, suppose $\alpha_i \in (0,\, 1)$, with $\sum_{i=1}^{l} \alpha_i = 1$,

$$\hat{\nu} \in \arg\min_{\nu \in \mathcal{P}} \sum_{i=1}^{l} \alpha_i J_i(\nu),$$

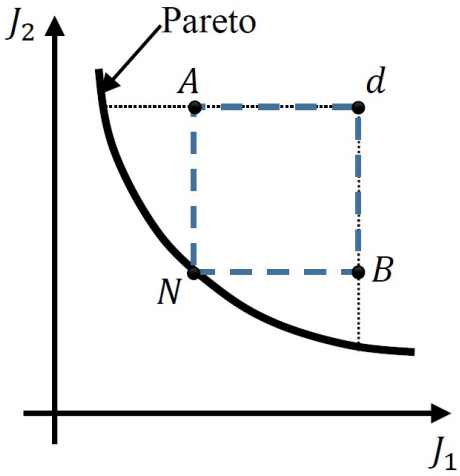


Figure 5.1: The Nash bargaining solution.

then $\hat{\nu}$ is Pareto optimally [26].

In addition, if $\mathcal{P}_f$ is convex and J_i, $\forall i$ is convex, for all Pareto optimal $\nu^* \in \mathcal{P}$ there exist $\alpha \in \mathcal{A}$, such that

$$\nu^* \in \arg\min_{\nu \in \mathcal{P}} \sum_{i=1}^{l} \alpha_i J_i(\nu),$$

where $\mathcal{A} := \{\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_l) | \alpha_i \geq 0 \text{ and } \sum_{i=1}^{l} \alpha_i = 1\}$. It simply means that a vector α can be found which gives a convex combination of the Pareto front and the result is still a point of the Pareto front.

Therefore, it is necessary to find an appropriate vector α which yields in selecting a trade-off point. This led us to the most influential concept in game theory, the Nash equilibrium. One of the useful methods to find the trade-off point is based on the Bargaining concept which has its origin in two papers by Nash [26].

The Nash Equilibrium point, or Nash Bargaining Solution is the point of the Pareto front set, $\mathcal{P}_f$, at which the product of utility gains from Threat point is maximal [95]. Threat or disagreement point is a point in which the objectives can expect to receive other better results than the one which becomes effective when they do not cooperate or negotiations break down. Consider an optimization problem with two objective functions, $\mathbf{J} = [J_1, J_2]$. If ν_1 is the solution of the single objective optimization, without J_2, likewise, ν_2 is the solution when J_1 is not considered; then, $d = (d_1, d_2)$ is a disagreement point in which $d_1 = J_1(\nu_2)$ and $d_2 = J_2(\nu_1)$.

According to the Nash bargaining solution, the trade-off point is

$$N(\mathcal{P}_f, d) = \arg \max_{J \in \mathcal{P}_f} \prod_{i=1}^{l} (J_i - d_i), \text{ for } J \in \mathcal{P}_f \text{ with } J \ll d, \tag{5.2}$$

where J_i is the objective function, $J \ll d$ means J dominates d, and l is the objective functions number. Based on the assumption that all cost functions J_i are convex, every optimal solution which results in Pareto front can be obtained by minimizing a linear combination of these cost functions [26]. Therefore, if the set of all cooperative Pareto solutions is given by

$$\left[(J_1(\nu^*(\alpha)), \ldots, J_l(\nu^*(\alpha)) \right], \tag{5.3}$$

the corresponding Pareto optimal solution is obtained as [26]

$$\nu^*(\alpha) = \arg \min_{\nu \in \bar{\nu}} \sum_{i=1}^{l} \alpha_i J_i. \tag{5.4}$$

Figure 5.1 illustrates the concept of Nash bargaining solution geometrically. The Nash bargaining solution is the point on the edge of $\mathcal{P}_f$ and a part of the Pareto frontier which yields the largest rectangle (N, A, B, d) [26]. This point can be obtained by a few Pareto points [96]; therefore, this method can be beneficial to MPC design. The methodology of the controller design using Nash Bargaining Solution is given in the following sections.

5.3 Problem Statement

Consider the following discrete-time LPV system of the form

$$x_{k+1} = A(\mathbf{p}_k)x_k + B(\mathbf{p}_k)u_k, \tag{5.5}$$

$$y_k = C(\mathbf{p}_k)x_k + D(\mathbf{p}_k)u_k, \tag{5.6}$$

where $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^m$ is the control input, and $y_k \in \Re^p$ is the measured output. Also, the input and output are subjected to the following constraints

$$-u_{max} \leq u_k \leq u_{max} \tag{5.7}$$

$$-y_{max} \leq y_k \leq y_{max} \tag{5.8}$$

The time-varying system matrix is defined as follows

$$\mathbf{S}(\mathbf{p}_k) = \begin{pmatrix} A(\mathbf{p}_k) & B(\mathbf{p}_k) \\ C(\mathbf{p}_k) & D(\mathbf{p}_k) \end{pmatrix} \in \mathbb{R}^{(n+p)\times(n+m)}, \tag{5.9}$$

as described in Chapter 2, this matrix belong to the convex hall given in (2.5), including the convex combination of (L) LTI models.

Using the LPV model, the control signal is derived by minimizing an upper bound of the worst-case infinite horizon quadratic cost at sampling time k

$$J^k = \max_{\mathbf{p}\in\Omega} \sum_{i=0}^{\infty} x_{k+i|k}^T Q x_{k+i|k} + u_{k+i|k}^T R u_{k+i|k}, \tag{5.10}$$

where $\mathcal{Q} \succ 0$ and $\mathcal{R} \succ 0$ are weighting matrices which should be designed by the designer to make a trade-off between the response performance, and control input cost. Therefore, the performance of the control system depends on these matrices. In the following sections, a method will be proposed in which the designer does not need to specify these matrices.

5.4 Multi-Objective MPC Design

As mentioned before, the goal of this chapter is to find a method which helps the designer to solve the multi-objective MPC problems without choosing the weighting matrices. In order to obtain the control signal at each sampling time, using the method described in Section 4.3, the following optimization problem should be solved at each sampling time

$$\underset{\gamma_k, X_k, Y_k}{\text{minimize}}\, \gamma_k \tag{5.11a}$$

subject to

$$\begin{bmatrix} X_k & * & * & * \\ A_j X_k + B_j Y_k & X_k & * & * \\ X_k & \mathbf{0} & \gamma_k Q^{-1} & * \\ Y_k & \mathbf{0} & \mathbf{0} & \gamma_k R^{-1} \end{bmatrix} \succeq 0, \tag{5.11b}$$

$$\begin{bmatrix} \mathbf{1} & x_{k|k}^T \\ x_{k|k} & X_k \end{bmatrix} \succeq 0, \tag{5.11c}$$

$$\begin{bmatrix} u_{max}^2 \mathbf{I} & Y_k \\ Y_k^T & X_k \end{bmatrix} \succeq 0, \tag{5.11d}$$

$$\begin{bmatrix} y_{max}^2 \mathbf{I} & C_k(A_j X_k + B_j Y_k) \\ (A_j X_k + B_j Y_k)^T C_k^T & X_k \end{bmatrix} \succeq 0, \tag{5.11e}$$

where, $j = 1, 2, \ldots, L$ (L is the number vertices). And, u_{max}, and y_{max} are the upper bound for input and output, respectively.

The LMI control design method in (5.11) can be extended to the multi-objective optimization problem. Since, the performance of the control system depends on the weighting matrices, the following method is proposed to tune these matrices based on the Nash bargaining solution at each sampling time.

According to the Pareto front properties, since the objective functions in (5.10) are convex, the cost function J can be defined as a linear combination of those convex objective functions by specifying the weighting matrices, Q and R as follows

$$Q = \mathbf{diag}(\hat{\alpha}), \quad \hat{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n], \tag{5.12}$$

$$R = \mathbf{diag}(\bar{\alpha}), \quad \bar{\alpha} = [\alpha_{n+1}, \alpha_{n+2}, \ldots, \alpha_{n+m}] \tag{5.13}$$

where **diag** means the diagonal matrix and $\alpha = [\hat{\alpha}, \bar{\alpha}] \in \mathcal{A}$ is the tuning parameter vector obtained by Nash bargaining solution at each sampling time. Therefore, the designer does not need to design the weighting matrices, since they are automatically tuned at each sampling time.

In order to find the Nash equilibrium point, first of all, the threat or disagreement point must be found. The threat point can be obtained by solving the optimization problem at $k = 0$ for $\alpha_s = 1$, and $\alpha_i = 0, i \neq s$, that is, the optimization problem is solved for each objective function separately. Then n different points will be obtained. Each of which has the best value for the corresponding objective function and they may have the worse value for other objective functions. Finally, one single point among them is obtained, which is dominated by other points, and it is considered as a threat or disagreement point. Now, the multi-objective MPC can be designed using Nash equilibrium point.

The design steps of the multi-objective MPC are given as follows;

Step 1: **at** $k = 0$

1. **for** $i = 1$ **to** l
2. $\alpha \leftarrow [0, 0, \ldots, 0]$
3. $\alpha_i = 1;$
4. **find** $\nu_i = \arg\min_{\nu} J_i$
5. **end**

Step 2: **find threat point** d

6. **for** $i = 1$ **to** l
7. $d_i \leftarrow \max(J_i(\nu_1), J_i(\nu_2), \ldots, J_i(\nu_l))$
8. **end**

Step 3: **for** $k = 1$ **to** End **do**

9. **set:** $\alpha^0 \leftarrow [\frac{1}{l}, \ldots, \frac{1}{l}]$
10. **compute:** $\nu^* = \arg\min_{\nu \in \tilde{\nu}} \sum_{i=1}^{l} \alpha_i^0 J_i(\nu)$
11. **for** $i = 1$ **to** l **verify**
12. **if** $J_i(\nu^* \leq d_i)$
13. **calculate:** $\tilde{\alpha}_j^N = \frac{\prod_{i \neq j} (d_i - J_i(\nu^*(\alpha^0)))}{\sum_{i=1}^{l} \prod_{h \neq i} (d_h - J_h(\nu^*(\alpha^0)))}$, $j = 1, \ldots, l$
14. **else find** i_0 **which** $J_{i_0}(\nu^*) > d_{i_0}$
15. **update:** $\alpha_{i_0}^0 := \alpha_{i_0}^0 + 0.01$, $\alpha_i^0 := \alpha_i^0 - \frac{0.01}{l-1}, i \neq i_0$
16. **return to 10**
17. **end**
18. **if** $|\tilde{\alpha}_j^N - \alpha_i^0| < 0.01, i = 1, \ldots, l$

19. **terminate**

20. **set:** $\alpha_j^N = \tilde{\alpha}_j^N$

21. $k \leftarrow k + 1$

22. **else** $\alpha_i^0 := 0.8\alpha_i^0 + 0.2\tilde{\alpha}_i^N$

23. **return to 10**

24. **end**

It should be noted, the numbers 0.01 and 0.8 in the above-mentioned design procedure are chosen arbitrary [26].

5.5 Mass and Spring System

To illustrate the effectiveness of the proposed MPC method, the system consisting of a two-mass-spring model with a time-varying nonlinear spring coefficient, as in [43], [44] and [97], is considered and shown in Figure 5.2.

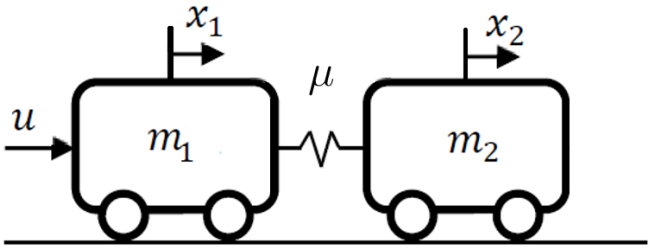


Figure 5.2: Two-mass-spring system.

5.5.1 Problem Formulation

The discrete-time state-space model of the two-mass-spring system (obtained from the continuous time model using a first-order Euler approximation with sampling time $h = 0.1\,s$) is

given as follows [44]

$$x_{k+1} = \begin{pmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ -0.1\frac{\mu}{m_1} & 0.1\frac{\mu}{m_1} & 1 & 0 \\ 0.1\frac{\mu}{m_2} & -0.1\frac{\mu}{m_2} & 0 & 1 \end{pmatrix} x_k + \begin{pmatrix} 0 \\ 0 \\ \frac{0.1}{m_1} \\ 0 \end{pmatrix} u_k, \tag{5.14}$$

where m_1 and m_2 are two masses and μ is the spring constant [97]. The state vector at each sampling time, x_k, includes the position of the masses $x_{1,k}$, $x_{2,k}$ and their velocities, $x_{3,k}$, $x_{4,k}$. In this example the masses are constant $m_1 = 1$ and $m_2 = 1$, while spring constant varies with time according to the following equation

$$\mu_k = 5.25 - 4.75\sin(0.5k). \tag{5.15}$$

It can be seen that $\mu_k \in [0.5, 10]$. According to [97], the weighting functions w_k can be defined as $w_{1,k} = 1 - \frac{\mu_k - 0.5}{9.5}$ and $w_{2,k} = 1 - w_{1,k}$ which satisfy convex hull condition in (2.5). For this system, two vertices based on the maximum and minimum values of the spring constant can be obtained as follows

$$A_1 = \begin{pmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ -0.05 & 0.05 & 1 & 0 \\ 0.05 & -0.05 & 0 & 1 \end{pmatrix}, \tag{5.16}$$

$$A_2 = \begin{pmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ -1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}, \tag{5.17}$$

$$B_1 = B_2 = \begin{pmatrix} 0 \\ 0 \\ 0.1 \\ 0 \end{pmatrix}. \tag{5.18}$$

The objective of the control design is to steer the two masses from the initial condition $x_0 = [1, 1, 0, 0]^T$ to the origin. The control system must satisfy the input and output

(positions of the masses) constraints, $\|u_k\| \leq 0.05$ and $\|y_k\| \leq 1$, respectively. Since, the constraints are considered only for the positions, the matrix $C = [1\,0\,0\,0; 0\,1\,0\,0]$ is independent of w_k and is kept the same for both vertices.

5.5.2 Simulation Results

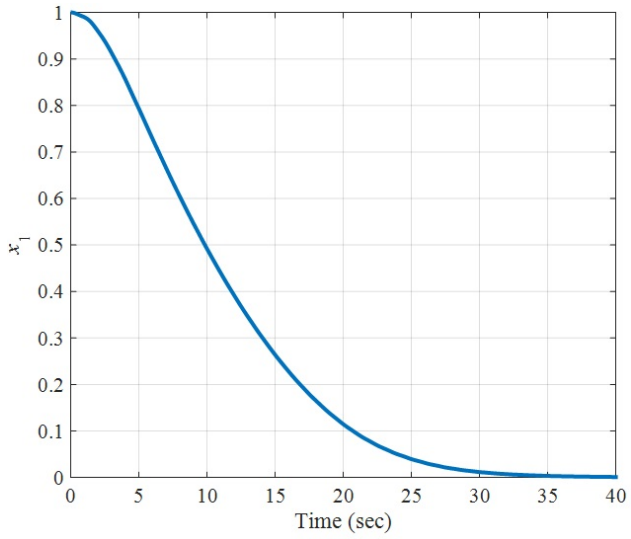
Figure 5.3 illustrates the obtained simulation results. It is obvious that the system is asymptotically stable, and the states are steered to the origin efficiently. Figure 5.4 shows the results from [43], [44], and [44]. It is noticed that the proposed MPC method steers the masses to the origin significantly faster compared with the other methods. According to the obtained results, the settling time is about 25 *sec*, while it is about 45 *sec* in [97].

The control signal of the proposed MPC method is shown in Figure 5.5a. Also, Figure 5.5b illustrates the control signal behavior of the methods in [43, 44, 97]. It can be seen that both input and output responses satisfied the considered constraints. Although the results show that the proposed MPC method has the greater control signal magnitude than the other methods do, it does not violate the constraint; therefore, the results are acceptable. It shows the advantage of the proposed method. At the beginning of the response, the state error is large. Therefore, using the Nash bargaining solution, the bigger states weights are chosen while the weight of the control input is small. This automatic tuning procedure is carried out within the simulation at each sampling interval to find the variable weighting matrices as $Q = \alpha_1 \mathbf{I}_{4\times4}$, and $R = \alpha_2$, $([\alpha_1, \alpha_2] \in \mathcal{A})$. On the contrary, the other methods use the fixed weighting matrices as $Q = \mathbf{I}_{4\times4}$, and $R = 1$.

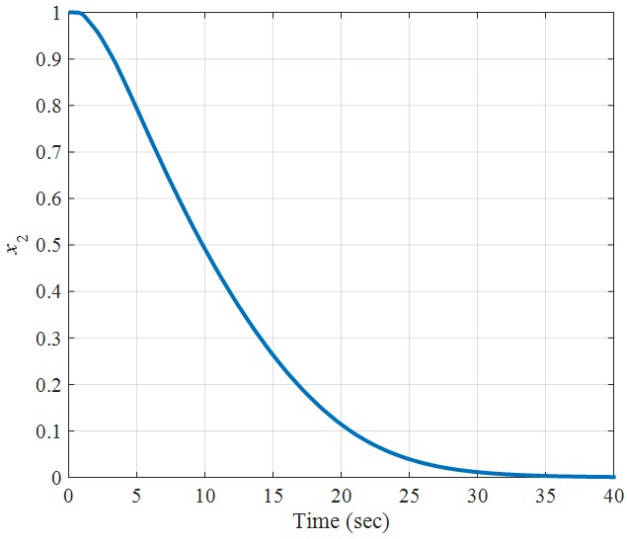
Figure 5.6 shows the minimized upper bound on the worst-case cost function in (5.11a). It clearly shows that the proposed method obtains incomparably smaller upper bound at each sampling time (Figure 5.6a) in comparison with the other methods (Figure 5.6b). As a result, the proposed controller is closer to the optimal solution that may be obtained for the unconstrained optimal control method (global minimal solution).

It is evident that applying the proposed MPC method [98] on the considered LPV model achieve significant performance improvements compared with the methods presented in [43], [44], [97].

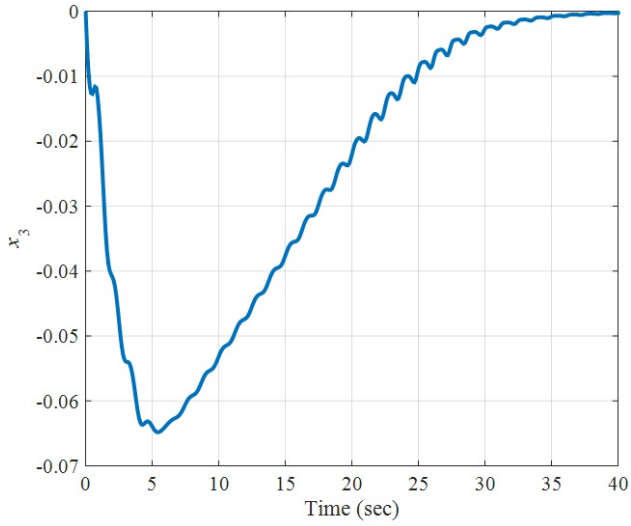
To generalize the application of the proposed method, in the following section a highly nonlinear system is considered and the proposed method is applied to that. Finally, the obtained results are compared with proposed method in [33].



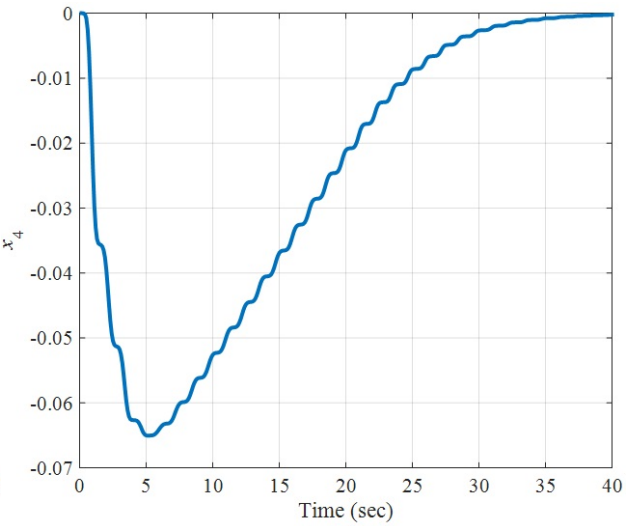
(a) Mass 1 Position (m)



(b) Mass 2 Position (m)

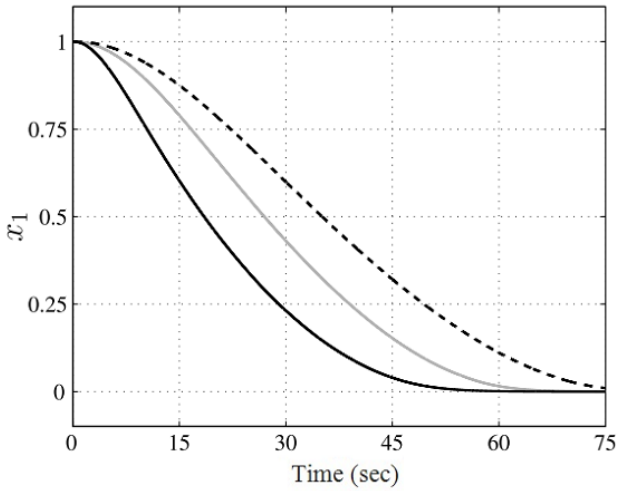


(c) Mass 1 Velocity (m/s)

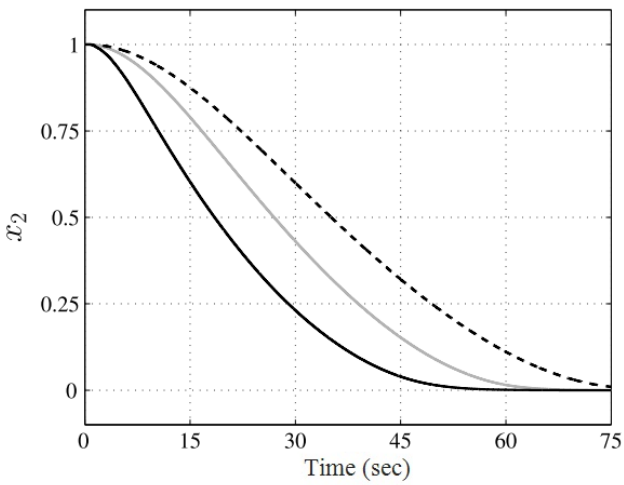


(d) Mass 2 Velocity (m/s)

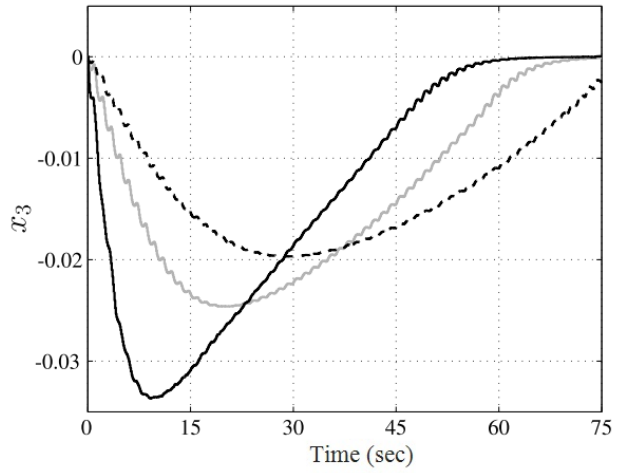
Figure 5.3: Time response of the state variables.



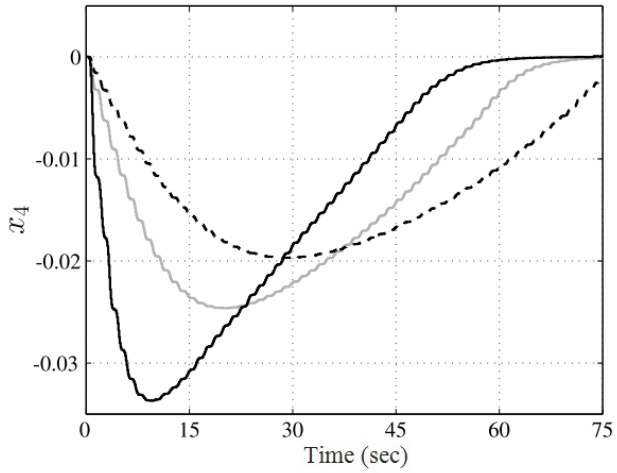
(a) Mass 1 Position (m)



(b) Mass 2 Position (m)

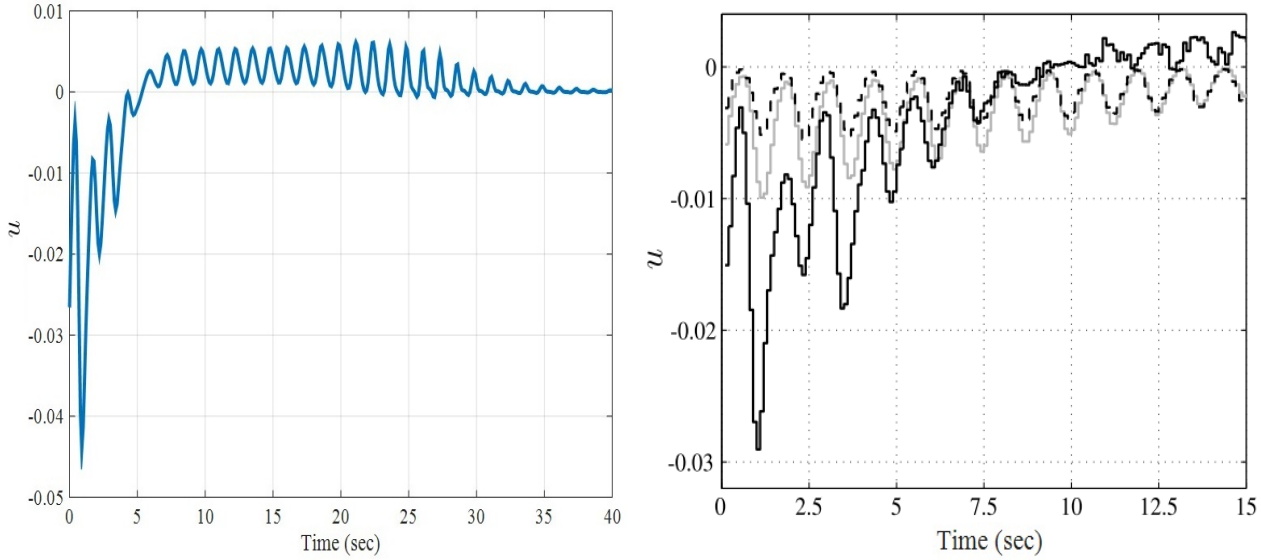


(c) Mass 1 Velocity (m/s)



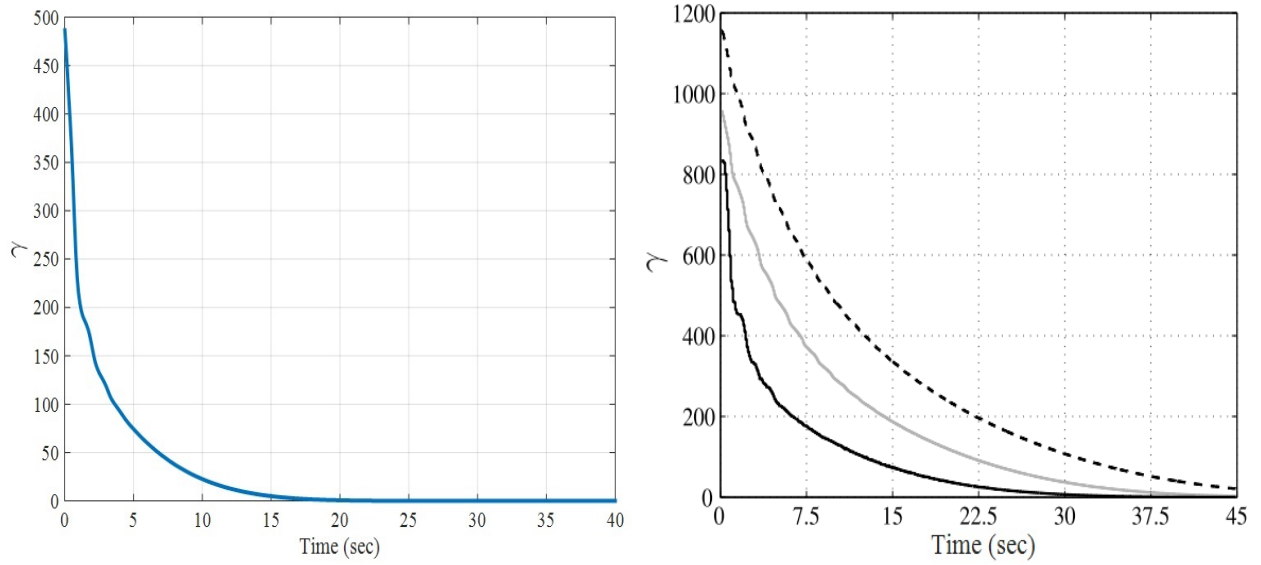
(d) Mass 2 Velocity (m/s)

Figure 5.4: The state responses of the results from [43] (dashed line), [44] (gray line), and [97] (solid black line)



(a) (b)

Figure 5.5: Control signal of (a) the proposed method, (b) [43] (dashed line), [44] (gray line), and [97] (solid black line).



(a) (b)

Figure 5.6: Upper bound on the considered cost functions, γ (a) the proposed method, (b) [43] (dashed line), [44] (gray line), and [97] (solid black line).

5.6 TORA System

In this section, the novel multi-objective model predictive control is executed on a nonlinear system. Also, the results will be compared with the results of the proposed method by Baranyi *et al.* in [33]. The disturbance rejection is also investigated as a robustness criteria for the proposed method.

5.6.1 Problem Formulation

Figure 5.7 illustrates a Translational Oscillator with an eccentric Rotational proof-mass Actuator (TORA) system which has been used in many research projects as a challenging benchmark for control designers [99–102]. The oscillator consists of a cart of mass M connected to a fixed wall by a linear spring of stiffness k. It can be seen that the cart is confined to have one-dimensional motion in the horizontal plane. As the motion is on the horizontal plane, the gravitational forces have no effect on motion. The proof-mass actuator is attached to the cart which has mass m and moment of inertia I about its center of mass. The mass is located a distance e from the rotational point. The control torque denoted by N is applied to the proof-mass [103, 104].

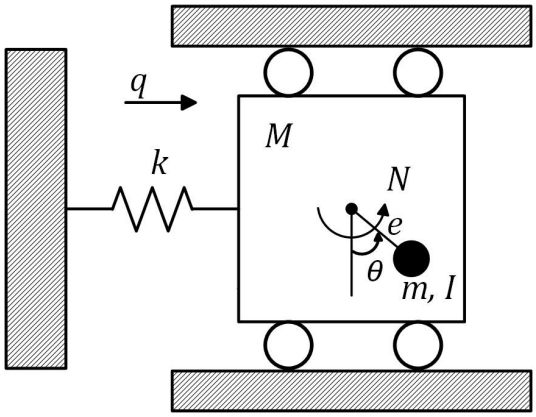


Figure 5.7: TORA system.

In order to derive the equations of motion, the followings are considered; the cases $\theta = 0^\circ$ where the mass motion is perpendicular to the cart motion, and $\theta = 90^\circ$ where the mass motion is in the direction of cart motion which is illustrated by q in Figure 5.7. The equations

of motion are adopted from [104] which are given as follows

$$(M+m)\ddot{q} + kq = -me\left(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta\right), \tag{5.19}$$

$$\left(I + me^2\right)\ddot{\theta} = -me\ddot{q}\cos\theta + N, \tag{5.20}$$

with the normalization

$$\xi \simeq \sqrt{\frac{M+m}{I+me^2}}q, \tag{5.21}$$

$$\tau \simeq \sqrt{\frac{k}{M+m}}t, \tag{5.22}$$

$$u \simeq \frac{M+m}{k\left(I+me^2\right)}N. \tag{5.23}$$

The equations of motion become

$$\ddot{\xi} + \xi = \varepsilon\left(\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta\right), \tag{5.24}$$

$$\ddot{\theta} = -\varepsilon\ddot{\xi}\cos\theta + u, \tag{5.25}$$

where ξ is the normalized cart position, and u is the non-dimension control torque, τ is the normalized time on which the differentiation is based. The coupling between the rotational and the translational motions, ε, is defined by

$$\varepsilon = \frac{me}{\sqrt{\left(I+me^2\right)\left(M+m\right)}}. \tag{5.26}$$

Let $\mathbf{x}(t) = [x_1,\, x_2,\, x_3,\, x_4]^T = \left[\xi,\, \dot{\xi},\, \theta,\, \dot{\theta}\right]^T$. The state-space model of the non-dimensional equations of motion can be written as

$$\dot{x}(\tau) = f(x(\tau)) + g(x(\tau))u(t), \tag{5.27a}$$

$$y(\tau) = h(x(\tau)), \tag{5.27b}$$

where

$$f(x) = \begin{pmatrix} x_2 \\ \frac{-x_1+\varepsilon x_4^2 \sin x_3}{1-\varepsilon^2 \cos^2 x_3} \\ x_4 \\ \frac{\varepsilon \cos x_3 \left(x_1-\varepsilon x_4^2 \sin x_3\right)}{1-\varepsilon^2 \cos^2 x_3} \end{pmatrix}, \tag{5.28a}$$

$$g(x) = \begin{pmatrix} 0 \\ \frac{-\varepsilon \cos x_3}{1-\varepsilon^2 \cos^2 x_3} \\ 0 \\ \frac{1}{1-\varepsilon^2 \cos^2 x_3} \end{pmatrix}, \tag{5.28b}$$

$$h(x) = \begin{pmatrix} x_1 & 0 & 0 & 0 \\ 0 & 0 & x_3 & 0 \end{pmatrix}. \tag{5.28c}$$

Now, the LPV state-pace model of the nonlinear system can be written as

$$\dot{\mathbf{x}}(t) = \mathbf{S}(\mathbf{p}(t)) \begin{pmatrix} \mathbf{x}(t) \\ u(t) \end{pmatrix}, \quad y(t) = \mathbf{C}\mathbf{x}(t), \tag{5.29}$$

where system matrix $\mathbf{S}(\mathbf{p}(t))$ includes

$$\mathbf{S}(\mathbf{p}(t)) = \begin{pmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B}(\mathbf{p}(t)) \end{pmatrix}, \tag{5.30}$$

and the vector of varying parameters is $\mathbf{p}(t) = [x_3(t) \quad x_4(t)] \in \Omega$; therefore, one has

$$\begin{gathered} \mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{-1}{1-\varepsilon^2 \cos^2 x_3} & 0 & 0 & \frac{\varepsilon x_4 \sin x_3}{1-\varepsilon^2 \cos^2 x_3} \\ 0 & 0 & 0 & 1 \\ \frac{\varepsilon \cos x_3}{1-\varepsilon^2 \cos^2 x_3} & 0 & 0 & \frac{-\varepsilon^2 x_4 \cos x_3 \sin x_3}{1-\varepsilon^2 \cos^2 x_3} \end{pmatrix} \\ \mathbf{B} = g(\mathbf{x}(t)), \quad \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \end{gathered} \tag{5.31}$$

The parameters of the laboratory-scale model of this benchmark problem described in [104] are given in Table 5.1. The constraints of the system are considered as

$$|q| \leq 0.025m, \tag{5.32a}$$

Table 5.1: Parameters of the TORA system

Description	Parameter	Value	Units
Cart mass	M	1.3608	kg
Arm mass	m	0.096	kg
Arm eccentricity	e	0.0592	m
Arm inertias	I	0.0002175	kg.m^2
Spring stiffness	k	186.3	N/m
Coupling parameter	ε	0.2	–

$$|N| \leq 0.1Nm. \tag{5.32b}$$

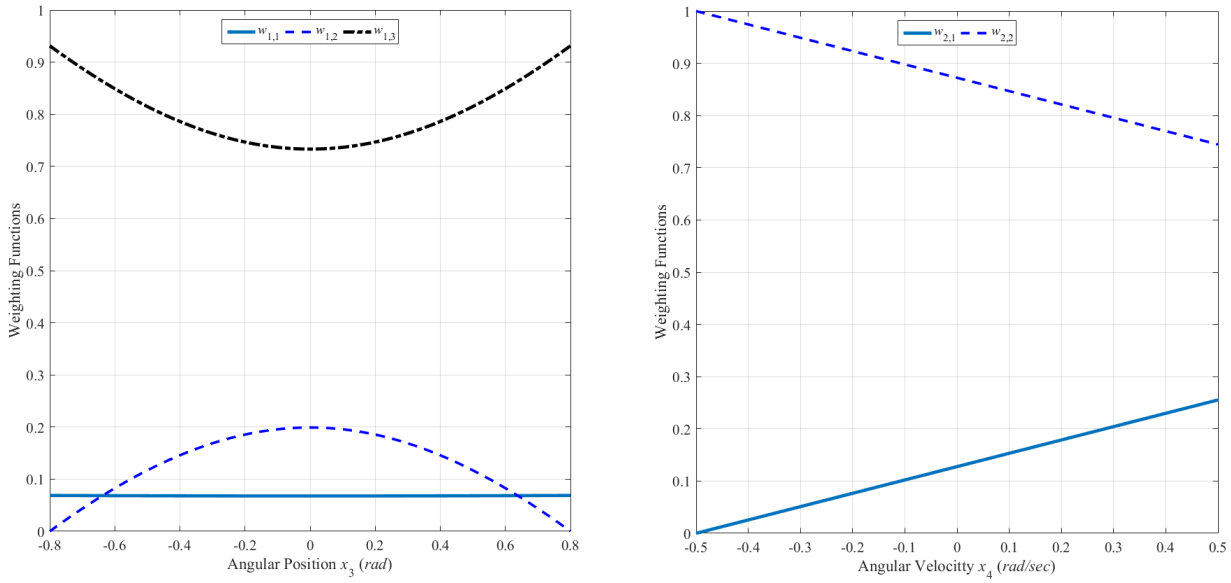
In order to find the TP model transformation of the nonlinear model, the constraints defined in (5.32) must be considered. Also, θ is considered to be smaller than $0.85\,rad$ and $\dot{\theta}$ is not larger than $0.5\,rad/sec$ [104]. Therefore, the varying parameters space can be defined as $x_3(t) \in [-0.85,\, 0.85]$ and $x_4(t) \in [-0.5,\, 0.5]$.

Using the method described in Chapter 2, to find the discretization model, the hyper-rectangular N-dimensional space grid is generated and the TP model transformation is used. 100 samples are considered on each dimension for discretization. Therefore, a $100 \times 100 \times 100 \times 2 \times 5$ tensor of the system is obtained. The nonzero singular values are obtained as

$$\begin{aligned} \sigma_1^{(1)} &= 205.056, \quad \sigma_1^{(2)} = 205.076, \\ \sigma_2^{(1)} &= 2.846, \qquad \sigma_2^{(2)} = 2.65, \\ \sigma_3^{(1)} &= 2.653, \\ \sigma_4^{(1)} &= 0.042, \\ \sigma_5^{(1)} &= 0.028. \end{aligned}$$

Hence, the TORA system can be exactly given in the HOSVD-based polytopic model form with minimum $5 \times 2 = 10$ LTI vertex models. To reduce the computational load of the control design, a trade-off between complexity and accuracy is made. In this case the small singular values which are possible to discard are $\sigma_4^{(1)}$, and $\sigma_5^{(1)}$. Hence, the reduced HOSVD-based polytopic model is formed with $3 \times 2 = 6$ LTI vertex model, which can be represented as follows

$$\dot{x}(t) = \sum_{i=1}^{3} \sum_{j=1}^{2} w_{1,i}(x_3(t)) w_{2,j}(x_4(t)) (A_{i,j}x(t) + B_{i,j}u(t)), \tag{5.33}$$



(a) (b)

Figure 5.8: SNNN type weighting functions of the reduced TP model form by 6 LTI vertices.

where $w_{i,j}(.)\, i = 1, 2\, 3,\, j = 1, 2$ are the weighting functions depicted in Figure 5.8. Also, the LTI system matrices of the polytopic model are

$$A_{1,1} = \begin{pmatrix} 0 & 1.000 & 0 & 0 \\ -1.035 & 0 & 0 & -0.185 \\ 0 & 0 & 0 & 1.000 \\ 0.247 & 0 & 0 & 0.005 \end{pmatrix}, \qquad B_{1,1} = \begin{pmatrix} 0 \\ -0.247 \\ 0 \\ 1.035 \end{pmatrix},$$

$$A_{1,2} = \begin{pmatrix} 0 & 1.000 & 0 & 0 \\ -1.035 & 0 & 0 & 0.185 \\ 0 & 0 & 0 & 1.000 \\ 0.247 & 0 & 0 & -0.005 \end{pmatrix}, \qquad B_{1,2} = \begin{pmatrix} 0 \\ -0.247 \\ 0 \\ 1.035 \end{pmatrix},$$

$$A_{2,1} = \begin{pmatrix} 0 & 1.000 & 0 & 0 \\ -1.061 & 0 & 0 & 0.108 \\ 0 & 0 & 0 & 1.000 \\ 0.118 & 0 & 0 & -0.019 \end{pmatrix}, \qquad B_{2,1} = \begin{pmatrix} 0 \\ -0.118 \\ 0 \\ 1.061 \end{pmatrix},$$

$$A_{2,2} = \begin{pmatrix} 0 & 1.000 & 0 & 0 \\ -1.061 & 0 & 0 & -0.108 \\ 0 & 0 & 0 & 1.000 \\ 0.118 & 0 & 0 & 0.019 \end{pmatrix}, \qquad B_{2,2} = \begin{pmatrix} 0 \\ -0.118 \\ 0 \\ 1.061 \end{pmatrix},$$

$$A_{3,1} = \begin{pmatrix} 0 & 1.000 & 0 & 0 \\ -1.017 & 0 & 0 & -0.009 \\ 0 & 0 & 0 & 1.000 \\ 0.164 & 0 & 0 & 0.003 \end{pmatrix}, \qquad B_{3,1} = \begin{pmatrix} 0 \\ -0.164 \\ 0 \\ 1.017 \end{pmatrix},$$

$$A_{3,2} = \begin{pmatrix} 0 & 1.000 & 0 & 0 \\ -1.017 & 0 & 0 & 0.009 \\ 0 & 0 & 0 & 1.000 \\ 0.164 & 0 & 0 & -0.003 \end{pmatrix}, \qquad B_{3,2} = \begin{pmatrix} 0 \\ -0.164 \\ 0 \\ 1.017 \end{pmatrix}.$$

As mentioned before, in order to investigate the advantage of the proposed method over other methods, the simulation results are compared with those of the proposed method in [33, 105] (call it Baranyi Method). In [105], the authors used Parallel Distributed Compensation (PDC) framework to design the state-feedback controller for LPV systems. The PDC framework was proposed by Wang and Tanaka in 1995, [106]. First, it was developed for Takagi-Sugeno (TS) fuzzy system, and recently a number of LMI design theorems have developed for the PDC design framework [29]. In this method, the goal is to find a local feedback controller for each LTI system such that the state feedback law stabilizes the local system. Then the control signal is obtained as a convex combination of each local controller using weighting functions.

The simulation and comparison results are presented in the next section.

5.6.2 Simulation Results

In order to design the multi-objective MPC, the weighting matrices are chosen as follows

$$Q = \begin{pmatrix} \alpha_1 & 0 & 0 & 0 \\ 0 & \alpha_2 & 0 & 0 \\ 0 & 0 & \alpha_3 & 0 \\ 0 & 0 & 0 & \alpha_4 \end{pmatrix}, \tag{5.34a}$$

$$R = \alpha_5, \tag{5.34b}$$

where, $\alpha_i > 0$, $i = 1, \ldots, 5$, and $\sum_{i=1}^{5} \alpha_i = 1$, which are tuned automatically at each sampling time.

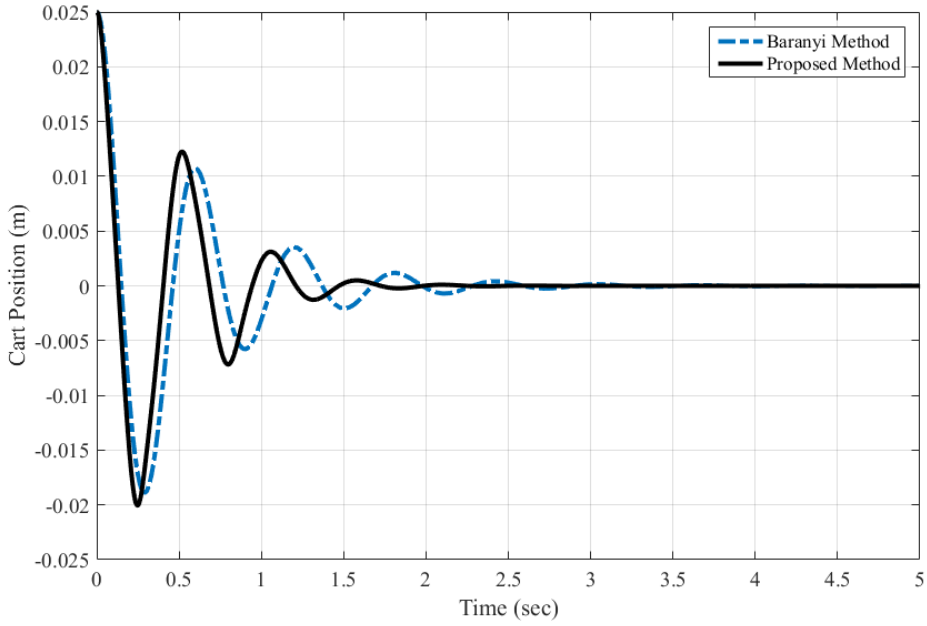
The control design objective is to steer the cart from the initial condition $x_0 = [0.025\,m, 0, 0, 0]^T$ to the origin using proof-mass actuator.

The state responses are illustrated in Figures 5.9 and 5.10. It can be seen in Figure 5.9a the proposed controller acts faster than the Baranyi method. Based on the simulation results, the system settling time with the proposed controller is $1.5\,sec$, while with the Baranyi controller it is $2.5\,sec$.

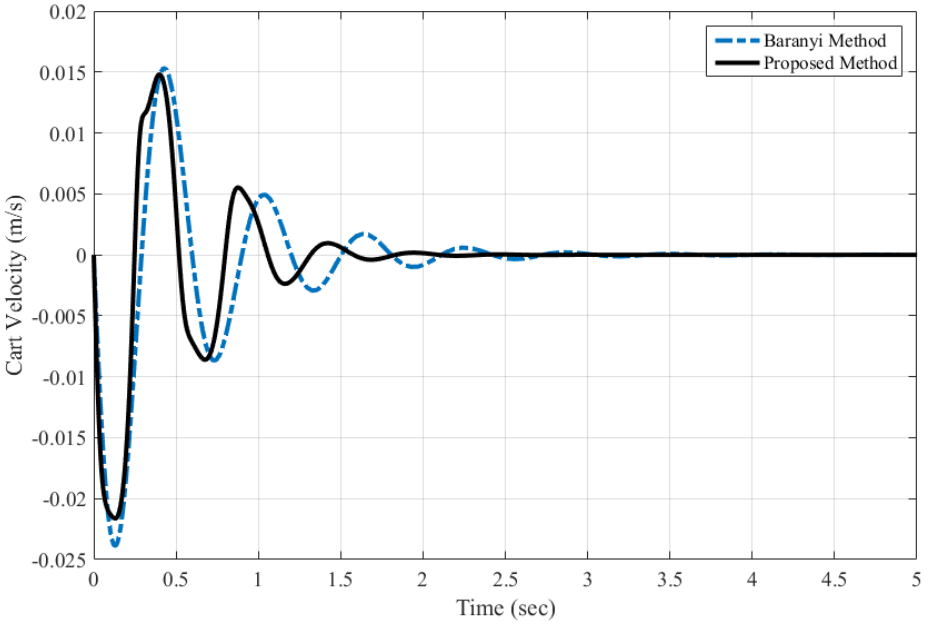
Also, with the proposed MPC, less oscillation is obtained compared with the Baranyi method. It can be seen in Figure 5.10a that the maximum proof-mass actuator angle for the system with the proposed controller is $100\,degree$, but for the system with the Baranyi controller is about $140\,degree$.

Consequently, the obtained control signal using the proposed method is smaller than that is the Baranyi methods, which shows the efficiency of the proposed method. The faster response with the smaller control signal is attained in comparison with the Baranyi method, that is, the obtained MPC method is closer to the optimal solution that would be obtained if the optimization problem could be solved analytically.

Figure 5.12 graphically illustrates the comparison results between the proposed method and the Baranyi method. It clearly shows the supremacy of the proposed method over the Baranyi method. Although the Baranyi method has the peak of the position with 10% less than the proposed method, in terms of other criteria the proposed method shows 26%, 28%, and 18% improvement in Angle Peak, Settling Time, and Power, respectively. It should be mentioned that the power is measure by $P = \tau \times \omega$.

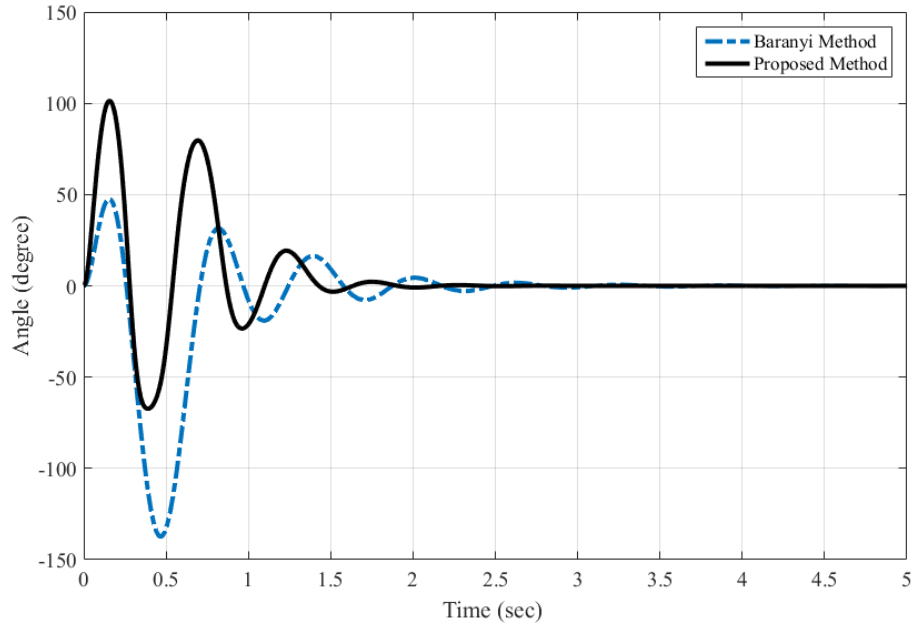


(a)

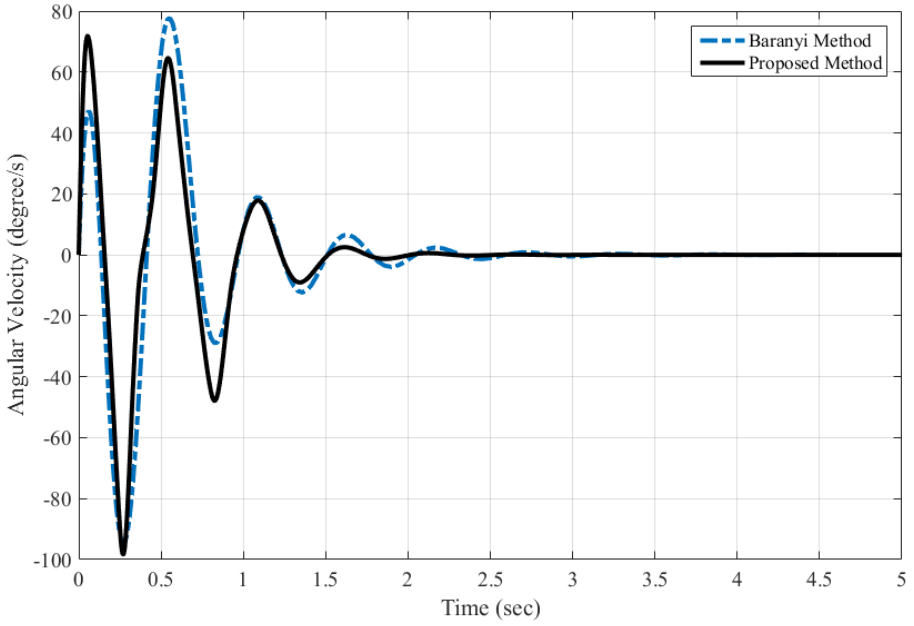


(b)

Figure 5.9: The cart (a) position, and (b) velocity responses.



(a)



(b)

Figure 5.10: The proof-mass actuator (a) angle, and (b) angular velocity responses.

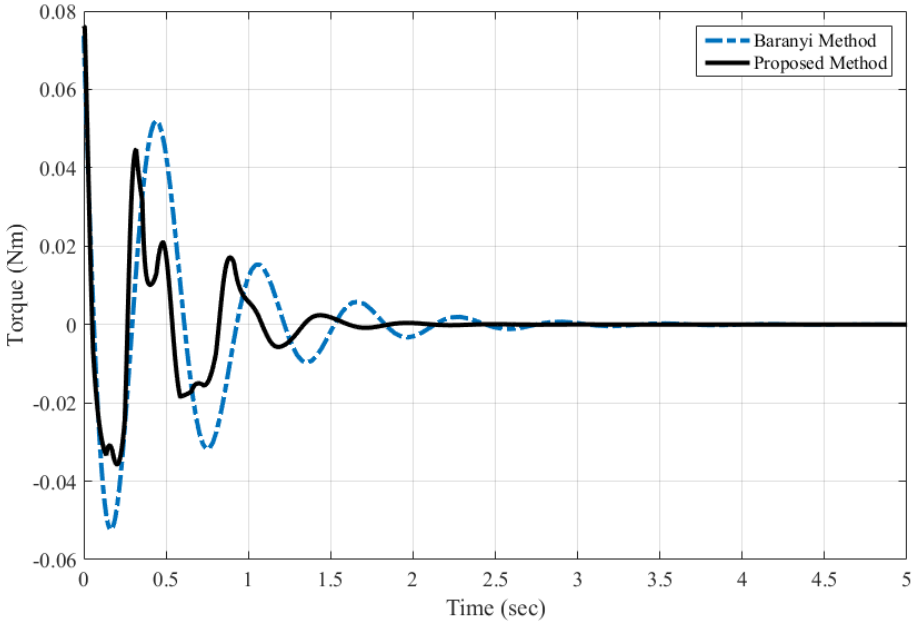


Figure 5.11: The applied torque on the system.

The disturbance rejection performance is considered as a measure to analyze the robustness of the proposed method, and to compare it with Baranyi method.

Assuming the applied disturbance as an external forces on the cart, the extended state-space model can be written as [33]

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) + \mathcal{D}(x(t))\mathcal{W}(t), \tag{5.35a}$$

$$y(t) = h(x(t)), \tag{5.35b}$$

where

$$f(x) = \begin{pmatrix} x_2 \\ \frac{-x_1+\varepsilon x_4^2 \sin x_3}{1-\varepsilon^2 \cos^2 x_3} \\ x_4 \\ \frac{\varepsilon \cos x_3 \left(x_1-\varepsilon x_4^2 \sin x_3\right)}{1-\varepsilon^2 \cos^2 x_3} \end{pmatrix}, \tag{5.36a}$$

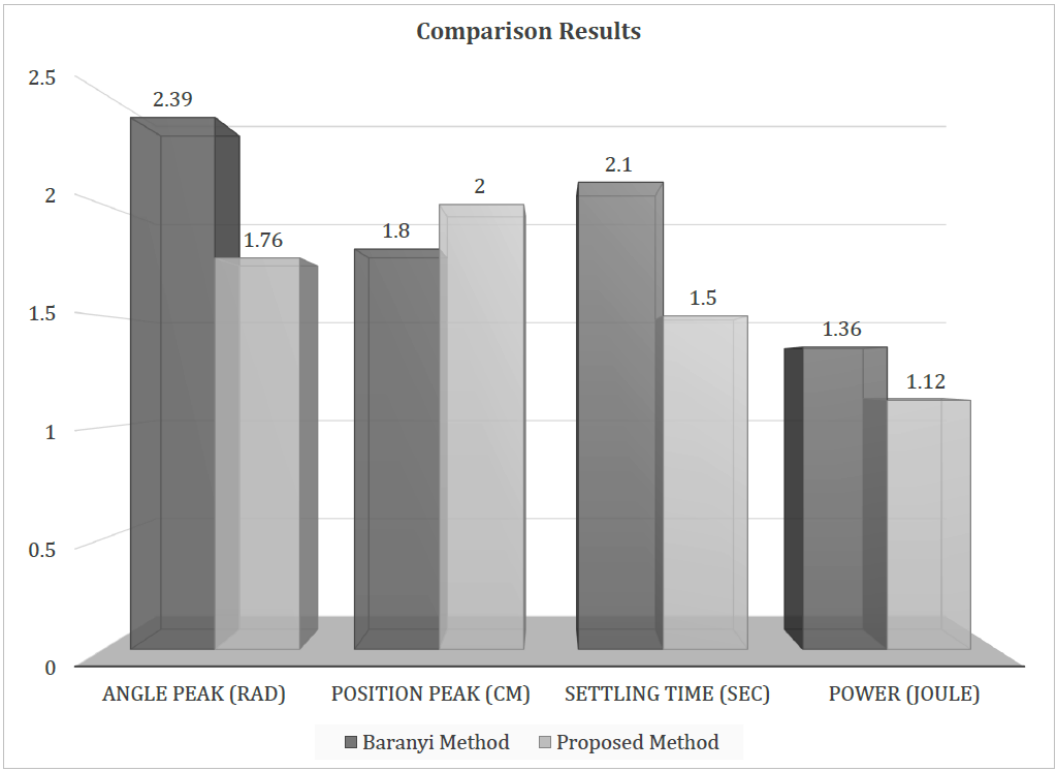


Figure 5.12: The comparison results between the proposed method and the Baranyi method.

$$g(x) = \begin{pmatrix} 0 \\ \frac{-\varepsilon \cos x_3}{1-\varepsilon^2 \cos^2 x_3} \\ 0 \\ \frac{1}{1-\varepsilon^2 \cos^2 x_3} \end{pmatrix}, \tag{5.36b}$$

$$\mathcal{D}(x) = \begin{pmatrix} 0 \\ \frac{1}{1-\varepsilon^2 \cos^2 x_3} \\ 0 \\ \\ \frac{-\varepsilon \cos x_3}{1-\varepsilon^2 \cos^2 x_3} \end{pmatrix}, \tag{5.36c}$$

$$h(x) = \begin{pmatrix} x_1 & 0 & 0 & 0 \\ 0 & 0 & x_3 & 0 \end{pmatrix}, \tag{5.36d}$$

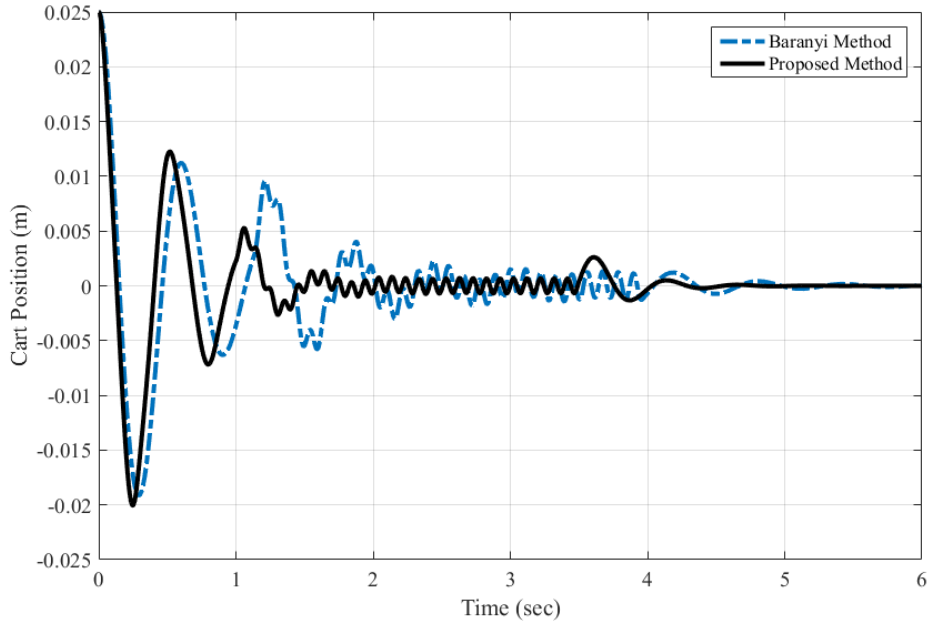
and $\mathcal{W}(t)$ is the disturbance.

In this simulation, $\mathcal{W}(t) = 0.5\sin(5t)$ is added to the system for $1\,sec < t < 3.5\,sec$. The

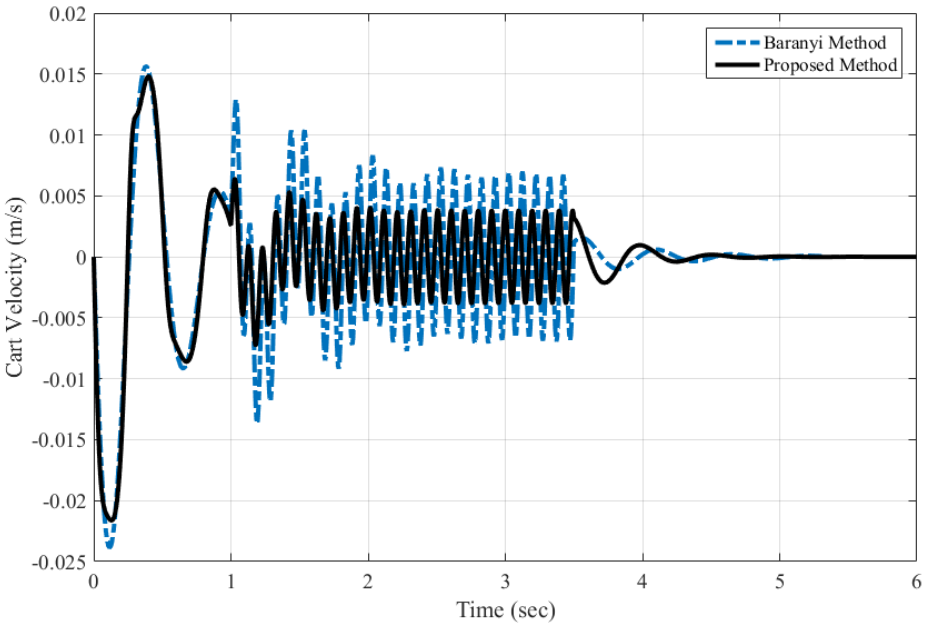
initial conditions are considered as $x_0 = [0.025\,m, 0, 0, 0]^T$. The states responses are shown in Figures 5.13 and 5.14. It can be observed that the proposed MPC method can efficiently reject the effect of the disturbance on the system. Also, by comparing the results illustrated in 5.13 and 5.14, it can be concluded that the proposed method rejects the disturbance with less oscillation than Baranyi method responses.

Furthermore, the behavior of the control signal in Figure 5.15 shows that the proposed controller is able to react against the disturbance more efficiently compared with the Baranyi method. It is evident that within the period of applying the disturbance, the control signal obtained by Baranyi method is almost 10 times more than the proposed method. As the control torque is applied on the proof-mass actuator, the bigger torque results in more oscillation in the response, which is evident in Figure 5.14b.

To sum up, the application of the proposed MPC method on the highly nonlinear system shows the satisfactory performance and robustness. Also, the improvement in terms of performance and robustness are obvious compare with the controller approach in [33].

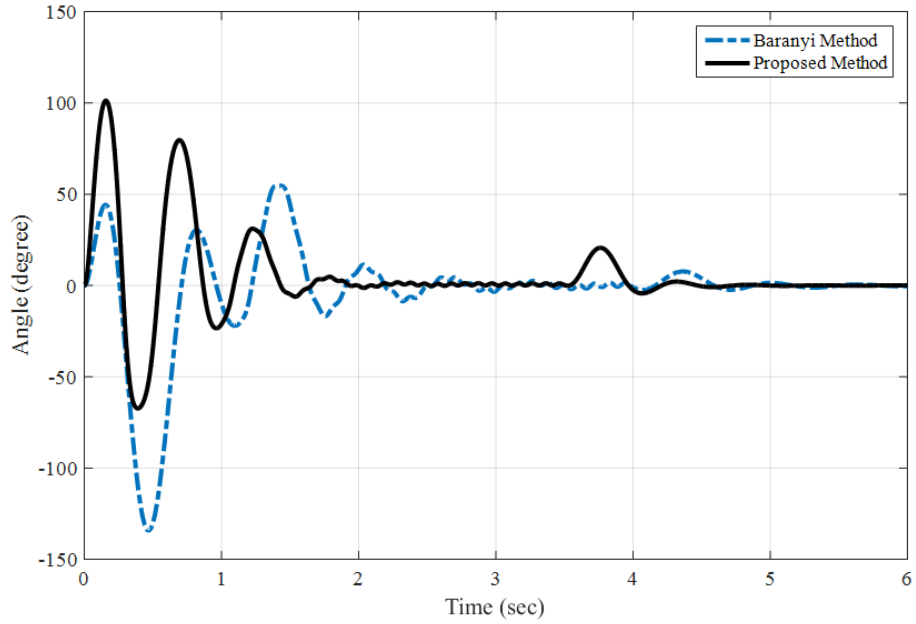


(a)

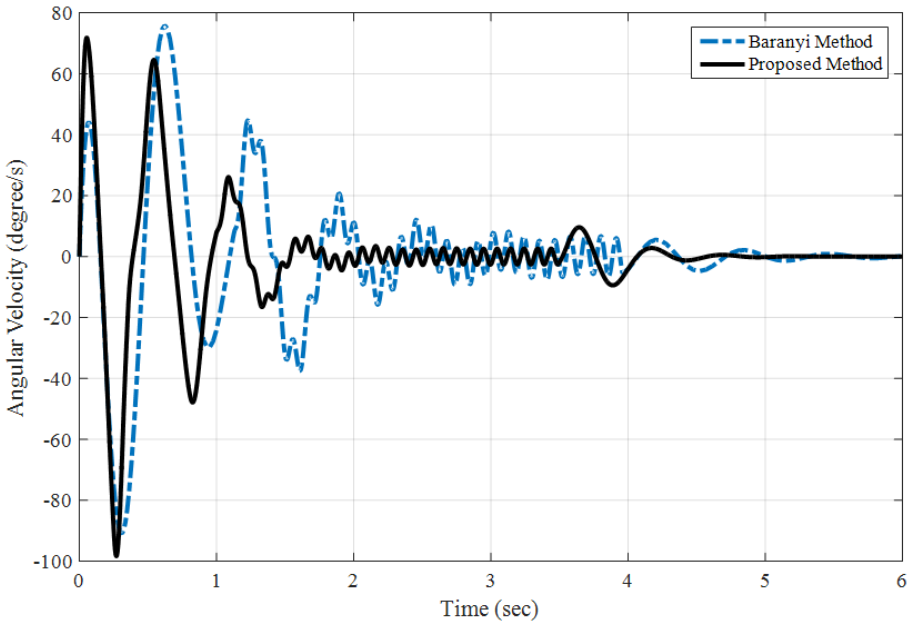


(b)

Figure 5.13: The cart (a) position, and (b) velocity responses with the external disturbance.



(a)



(b)

Figure 5.14: The proof-mass actuator (a) angle, and (b) angular velocity responses with the external disturbance.

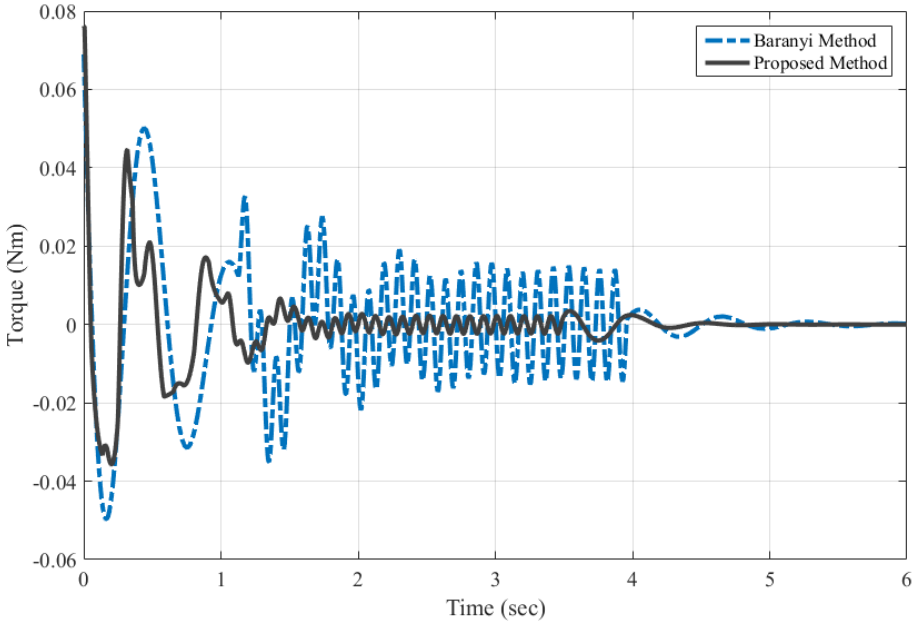


Figure 5.15: The applied torque on the system in the presence of the disturbance

5.7 Conclusion

This chapter presents a novel methodology to solve the problem of multi-objective model predictive control design. This method is proposed for linear parameter-varying systems. Multi-objective functions instead of single objective function are considered at each sampling time. This method leads to finding the trade-off between the objective functions. In order to solve the multi-objective optimization problem at each sampling time, the game theory and Nash bargaining solution are used to find the trade-off point. The Nash bargaining solution can find the trade-off point in game theories and can tune the weighting factors properly at each sampling time. The multi-objective optimization results are the solutions to a convex optimization problem based on linear matrix inequalities that are solved repeatedly at each sampling instant.

The simulation results show the effectiveness of the proposed method that can be generally used for the control system design with more than one objective functions.

Chapter 6

Conclusion and Future Work

6.1 Concluding Remark

A novel, computationally efficient MPC approach for linear parameter-varying systems has been developed in this thesis. Also, a diverse range of systems has been studied in this thesis. These systems include an LPV system with probabilistic uncertainty (Chapter 3), the 6 DOF robotics visual servoing system (Chapter 4), and the complex nonlinear TORA system (Chapter 5).

The MPC design for nonlinear or LPV systems normally leads to non-convex optimization problems. The TP model transformation method has been used to model the LPV systems based on the convex combination of the LTI vertices (models). As the convexity of the TP model, the optimization problems have been formulated in terms of LMIs, which can be solved efficiently using available optimization software.

The accuracy and computational time of the TP model transformation have been improved using HSS. With the aid of this method, the more coverage of the parameter varying space is obtained with the fewer sample numbers compared with the equidistance sampling method. Also, the robust MPC approach has been proposed to handle both time-varying and probabilistic uncertainties. Instead of considering norm-bounded model uncertainty, probabilistic uncertainty has been considered to reduce the conservatism of the norm-bounded uncertain model. Propagating the probabilistic uncertainties through the system model parameters yields the most likely models of the uncertain systems in designing the controller.

The major contributions in this thesis lie in improving the performance and robustness of the model predictive control. And they are summarized below

- The results attained from Shimmy vibration control clearly show that the proposed robust MPC method offers considerable benefits in reducing the optimization time.
- The proposed robust MPC method can readily handle the probabilistic uncertainty considered for the landing gear parameter, and the simulation results demonstrate that it can effectively reduce the shimmy vibration for the nominal operation range of an aircraft during landing.
- The real-time MPC-based IBVS controller shows the significant improvement in optimization time, which results in applying MPC on a fast dynamic system in real time.
- Different experimental tests on the 6 DOF robot set-up perfectly demonstrate the effectiveness of the proposed robust MPC. Moreover, the controller shows the good performance in the presence of the uncertain parameter, i.e. the feature depth, which is usually considered as a known parameter in the previous researches.
- The novel multi-objective MPC illustrate the significant improvement in terms of performance and robustness on a benchmark TORA system.
- The comparison results show clearly the supremacy of the proposed method over the other practical methods.
- Multi-objective optimization and Nash bargaining solution can obtain the optimal solution very close to the unconstrained optimal control solution.
- The proposed multi-objective MPC also shows the outstanding performance and disturbance rejection ability for complex nonlinear systems.

6.2 Future Work

The methodologies developed in this thesis are a first study on the robust multi-objective MPC which deal with the systems uncertainties. A number of extensions and possible future works are listed below

- In this thesis, it has been considered that the exact states are available at each sampling time. But, in many practical and industrial systems not all of the states are available. Therefore, the state estimation in conjunction the multi-objective MPC design can be considered as a potential future work.

- A parallel computing method can be used to apply multi-objective MPC in real-time. The proposed multi-objective MPC method needs to find a few Pareto front points at each sampling time. Then, using Nash bargaining solution, a trade-off point is obtained and applied to the system. The advantage of the using parallel computing method is to find Pareto front points at the same time, but it depends on the processor numbers. Then it would be possible to apply multi-objective MPC on-line in the future work.

- The polytopic LPV model has been considered in this thesis, so it is possible to study other LPV models such as norm-bound and diagonal norm-bound LPVs as a future work.

Bibliography

[1] H. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

[2] E. F. Camacho and C. Bordons, *Model Predictive Control*. Springer, 1998.

[3] T. Besselmann, J. Lofberg, and M. Morari, "Explicit mpc for lpv systems: Stability and optimality," *Automatic Control, IEEE Transactions on*, vol. 57, no. 9, pp. 2322–2332, 2012.

[4] M. Cannon, "Efficient nonlinear model predictive control algorithms," *Annual Reviews in Control*, vol. 28, no. 2, pp. 229 – 237, 2004.

[5] S. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733 – 764, 2003.

[6] M. Lawrynczuk, "A family of model predictive control algorithms with artificial neural networks," *Int. J. Appl. Math. Comput. Sci.*, vol. 17, no. 2, pp. 217–232, Jun. 2007.

[7] A. Vivas and P. Poignet, "Predictive functional control of a parallel robot," *Control Engineering Practice*, vol. 13, no. 7, pp. 863 – 874, 2005.

[8] L. Grune and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*. Springer, 2011.

[9] A. Topalov, O. Kaynak, and N. Shakev, "Neural network identification, predictive modeling and control with a sliding mode learning mechanism: an application to the robotic manipulators," in *Intelligent Systems, 2002. Proceedings. 2002 First International IEEE Symposium*, vol. 1, 2002, pp. 102 – 107 vol.1.

[10] J. Rawlings, "Tutorial overview of model predictive control," *Control Systems, IEEE*, vol. 20, no. 3, pp. 38 –52, jun 2000.

[11] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789 – 814, 2000.

[12] R. Findeisen and F. Allgwer, "An introduction to nonlinear model predictive," in *Control, 21st Benelux Meeting on Systems and Control, Veidhoven*, 2002, pp. 1–23.

[13] R. Milman and E. Davison, "Evaluation of a new algorithm for model predictive control based on non-feasible search directions using premature termination," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3, dec. 2003, pp. 2216 – 2221 Vol.3.

[14] V. A. Akpan and G. D. Hassapis, "Nonlinear model identification and adaptive model predictive control using neural networks," *ISA Transactions*, vol. 50, no. 2, pp. 177 – 194, 2011.

[15] A. Casavola, D. Famularo, and G. Franz, "Predictive control of constrained nonlinear systems via lpv linear embeddings," *International Journal of Robust and Nonlinear Control*, vol. 13, no. 3-4, pp. 281–294, 2003.

[16] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Siam, 1994, vol. 15.

[17] P. Baranyi, P. Korondi, R. J. Patton, and H. Hashimoto, "Trade-off between approximation accuracy and complexity for ts fuzzy models," *Asian Journal of control*, vol. 6, no. 1, pp. 21–33, 2004.

[18] P. Baranyi, Y. Yam, D. Tikk, and R. J. Patton, "Trade-off between approximation accuracy and complexity: Ts controller design via hosvd based complexity minimization," in *Interpretability Issues in Fuzzy Modeling*. Springer, 2003, pp. 249–277.

[19] C. Lin, *LMI approach to analysis and control of Takagi-Sugeno fuzzy systems with time delay*. Springer, 2007, vol. 351.

[20] P. Baranyi, "Tp model transformation as a way to lmi-based controller design," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 2, pp. 387–400, April 2004.

[21] P. Baranyi, "Tp model transformation as a way to lmi-based controller design," *Industrial Electronics, IEEE Transactions on*, vol. 51, no. 2, pp. 387–400, 2004.

[22] L. El Ghaoui and S.-I. Niculescu, *Advances in linear matrix inequality methods in control*. Siam, 2000, vol. 2.

[23] J. Löfberg, "Automatic robust convex programming," *Optimization Methods and Software*, vol. 27, no. 1, pp. 115–129, 2012.

[24] J. Lofberg, "Yalmip : a toolbox for modeling and optimization in matlab," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, Sept 2004, pp. 284–289.

[25] H. A. Hindi, B. Hassibi, and S. P. Boyd, "Multiobjective h_2/h_∞ optimal control via finite dimensional q-parametrization and linear matrix inequalities," in *American Control Conference, 1998. Proceedings of the 1998*, vol. 5. IEEE, 1998, pp. 3244–3249.

[26] J. Engwerda, *LQ Dynamic Optimization and Differential Games*. Wiley, 2005.

[27] K. Tanaka and H. O. Wang, *Fuzzy control systems design and analysis: a linear matrix inequality approach*. Wiley. com, 2004.

[28] C. Lin, *LMI approach to analysis and control of Takagi-Sugeno fuzzy systems with time delay*. Springer, 2007, vol. 351.

[29] C. Lin, G. Wang, T. H. Lee, and Y. He, *LMI approach to analysis and control of Takagi-Sugeno fuzzy systems with time delay*. Springer, 2007, vol. 351.

[30] B. A. Smith, S. P. Kenny, and L. G. Crespo, "Probabilistic parameter uncertainty analysis of single input single output control systems," *NASA report, TM-2005-213280*, 2005.

[31] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[32] A. Bertram and R. Glüge, "Introduction to tensor calculus," in *Solid Mechanics*. Springer, 2015, pp. 43–120.

[33] Y. Y. Baranyi, P. and P. Várlaki, *Tensor product model transformation in polytopic model–based control*. Taylor & Francis Group, LLC, 2014.

[34] J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *Journal of Process Control*, vol. 10, no. 4, pp. 363 – 385, 2000.

[35] J. Löfberg, *Minimax approaches to robust model predictive control*. Linköping University Electronic Press, 2003, vol. 812.

[36] S. Wang, D. Yu, J. Gomm, G. Page, and S. Douglas, "Adaptive neural network model based predictive control for airfuel ratio of si engines," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 2, pp. 189 – 200, 2006.

[37] M. Lawrynczuk, "On improving accuracy of computationally efficient nonlinear predictive control based on neural models," *Chemical Engineering Science*, vol. 66, no. 21, pp. 5253 – 5267, 2011.

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

[39] L. Chisci and E. Mosca, "Stabilizing i-o receding horizon control of carma plants," *Automatic Control, IEEE Transactions on*, vol. 39, no. 3, pp. 614–618, Mar 1994.

[40] S. Keerthi and E. Gilbert, "Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations," *Journal of Optimization Theory and Applications*, vol. 57, no. 2, pp. 265–293, 1988. [Online]. Available: http://dx.doi.org/10.1007/BF00938540

[41] P. Scokaert, D. Mayne, and J. Rawlings, "Suboptimal model predictive control (feasibility implies stability)," *Automatic Control, IEEE Transactions on*, vol. 44, no. 3, pp. 648–654, Mar 1999.

[42] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers; Chemical Engineering*, vol. 23, no. 45, pp. 667 – 682, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135498003019

[43] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[44] F. A. Cuzzola, J. C. Geromel, and M. Morari, "An improved approach for constrained robust model predictive control," *Automatica*, vol. 38, no. 7, pp. 1183 – 1189, 2002.

[45] N. Wada, K. Saito, and M. Saeki, "Model predictive control for linear parameter varying systems using parameter dependent lyapunov function," in *Circuits and Systems, 2004. MWSCAS'04. The 2004 47th Midwest Symposium on*, vol. 3. IEEE, 2004, pp. iii–133.

[46] E. Esmailzadeh and K. Farzaneh, "Shimmy vibration analysis of aircraft landing gears," *Journal of Vibration and Control*, vol. 5, no. 1, pp. 45–56, 1999.

[47] W. Krüger, I. Besselink, D. Cowling, D. Doan, W. Kortüm, and W. Krabacher, "Aircraft landing gear dynamics: simulation and control," *Vehicle System Dynamics*, vol. 28, no. 2-3, pp. 119–158, 1997.

[48] I. Jocelyn *et al.*, "An overview of landing gear dynamics," 1999.

[49] M. Fallah, S. Long, W. Xie, and R. Bhat, "Robust model predictive control of shimmy vibration in aircraft landing gears," *Journal of Aircraft*, vol. 45, no. 6, pp. 1872–1880, 2008.

[50] G. Somieski, "Shimmy analysis of a simple aircraft nose landing gear model using different mathematical methods," *Aerospace Science and Technology*, vol. 1, no. 8, pp. 545 – 555, 1997.

[51] A. Hajiloo and W. F. Xie, "The stochastic robust model predictive control of shimmy vibration in aircraft landing gears," *Asian Journal of Control*, vol. 17, no. 2, pp. 476–485, 2015.

[52] P. Baranyi, A. R. Várkonyi-Kóczy, Y. Yam, and R. J. Patton, "Adaptation of ts fuzzy models without complexity expansion: Hosvd-based approach," *Instrumentation and Measurement, IEEE Transactions on*, vol. 54, no. 1, pp. 52–60, 2005.

[53] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[54] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *Robotics Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, Dec 2006.

[55] Y. Wang, H. Lang, and C. de Silva, "Visual servo control and parameter calibration for mobile multi–robot cooperative assembly tasks," in *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, Sept 2008, pp. 635–639.

[56] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 331–340, April 2008.

[57] L. Ott, F. Nageotte, P. Zanne, and M. de Mathelin, "Robotic assistance to flexible endoscopy by physiological-motion tracking," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 346–359, April 2011.

[58] R. Mebarki, A. Krupa, and F. Chaumette, "2-d ultrasound probe complete guidance by visual servoing using image moments," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 296–306, April 2010.

[59] F. Chaumette and S. Hutchinson, "Visual servo control. ii. advanced approaches [tutorial]," *Robotics Automation Magazine, IEEE*, vol. 14, no. 1, pp. 109–118, March 2007.

[60] M. Keshmiri, W.-F. Xie, and A. Mohebbi, "Augmented image–based visual servoing of a manipulator using acceleration command," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5444–5452, Oct 2014.

[61] D.-H. Park, J.-H. Kwon, and I.-J. Ha, "Novel position-based visual servoing approach to robust global stability under field-of-view constraint," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4735–4752, Dec 2012.

[62] Y. Fang, X. Liu, and X. Zhang, "Adaptive active visual servoing of nonholonomic mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 486–497, Jan 2012.

[63] N. Garcia-Aracil, E. Malis, R. Aracil-Santonja, and C. Perez-Vidal, "Continuous visual servoing despite the changes of visibility in image features," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1214–1220, Dec 2005.

[64] A. Remazeilles, N. Mansard, and F. Chaumette, "A qualitative visual servoing to ensure the visibility constraint," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Oct 2006, pp. 4297–4303.

[65] P. Corke, F. Spindler, and F. Chaumette, "Combining cartesian and polar coordinates in ibvs," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, Oct 2009, pp. 5962–5967.

[66] M. Iwatsuki and N. Okiyama, "A new formulation of visual servoing based on cylindrical coordinate system with shiftable origin," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1, 2002, pp. 354–359.

[67] R. Fomena and F. Chaumette, "Improvements on visual servoing from spherical targets using a spherical projection model," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 874–886, Aug 2009.

[68] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 713–723, Aug 2004.

[69] I. Siradjuddin, L. Behera, T. McGinnity, and S. Coleman, "Image-based visual servoing of a 7-dof robot manipulator using an adaptive distributed fuzzy pd controller," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 512–523, April 2014.

[70] W. Sun, H. Gao, and O. Kaynak, "Adaptive backstepping control for active suspension systems with hard constraints," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 1072–1079, June 2013.

[71] W.-F. Xie, Z. Li, X.-W. Tu, and C. Perron, "Switching control of image-based visual servoing with laser pointer in robotic manufacturing systems," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 2, pp. 520–529, Feb 2009.

[72] B. Kadmiry and P. Bergsten, "Robust fuzzy gain scheduled visual-servoing with sampling time uncertainties," in *Intelligent Control, 2004. Proceedings of the 2004 IEEE International Symposium on*, Sept 2004, pp. 239–245.

[73] C. Lazar and A. Burlacu, "Visual servoing of robot manipulators using model–based predictive control," in *Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on*, June 2009, pp. 690–695.

[74] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 933–939, Oct 2010.

[75] T. T. Wang, W. F. Xie, G. D. Liu, and Y. M. Zhao, "Quasi-min-max model predictive control for image-based visual servoing with tensor product model transformation," *Asian Journal of Control*, vol. 17, no. 2, pp. 402–416, 2015.

[76] T. Besselmann, J. Lofberg, and M. Morari, "Explicit model predictive control for linear parameter-varying systems," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, Dec 2008, pp. 3848–3853.

[77] T. Besselmann, J. Lofberg, and M. Morari, "Explicit mpc for lpv systems: Stability and optimality," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2322–2332, 2012.

[78] E. Malis, F. Chaumette, and S. Boudet, "$2\frac{1}{2}$d visual servoing," vol. 15, no. 2, pp. 238 –250, Apr. 1999.

[79] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, ser. Cambridge books online. Cambridge University Press, 2004.

[80] M. Keshmiri and W. F. Xie, "Augmented imaged based visual servoing controller for a 6 dof manipulator using acceleration command," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 2012, pp. 556–561.

[81] W. Sun, Y. Zhao, J. Li, L. Zhang, and H. Gao, "Active suspension control with frequency band constraints and actuator input delay," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 530–537, Jan 2012.

[82] A. Hajiloo, M. Keshmiri, W.-F. Xie, and T.-T. Wang, "Robust on-line model predictive control for a constrained image based visual servoing," *IEEE Transactions on Industrial Electronics*, vol. PP, no. 99, pp. 1–1, 2015.

[83] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 933–939, Oct 2010.

[84] A. Hajiloo, N. Nariman-zadeh, and A. Moeini, "Pareto optimal robust design of fractional-order pid controllers for systems with probabilistic uncertainties," *Mechatronics*, vol. 22, no. 6, pp. 788 – 801, 2012.

[85] A. Jamali, A. Hajiloo, and N. Nariman-zadeh, "Reliability-based robust pareto design of linear state feedback controllers using a multi-objective uniform-diversity genetic algorithm (muga)," *Expert Systems with Applications*, vol. 37, no. 1, pp. 401 – 413, 2010.

[86] A. Hajiloo and W.-F. Xie, "Multi-objective control design of the nonlinear systems using genetic algorithm," in *2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings*, June 2014, pp. 27–34.

[87] A. Hajiloo and W.-F. Xie, "Multi-objective optimal fuzzy fractional-order pid controller design," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 18, no. 3, pp. 262–270, 2014.

[88] A. Hajiloo and W.-F. Xie, "Fuzzy fractional-order pid controller design using multi-objective optimization," in *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, June 2013, pp. 1445–1450.

[89] X. Blasco, J. Herrero, J. Sanchis, and M. Martnez, "A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization," *Information Sciences*, vol. 178, no. 20, pp. 3908 – 3924, 2008, ¡ce:title¿Special Issue on Industrial Applications of Neural Networks¡/ce:title¿ ¡xocs:full-name¿10th Engineering Applications of Neural Networks 2007¡/xocs:full-name¿.

[90] A. Bemporad and D. M. de la Pea, "Multiobjective model predictive control," *Automatica*, vol. 45, no. 12, pp. 2823 – 2830, 2009.

[91] V. M. Zavala and A. Flores-Tlacuahuac, "Stability of multiobjective predictive control: A utopia-tracking approach," *Automatica*, vol. 48, no. 10, pp. 2627 – 2632, 2012.

[92] J. J. V. Garca, V. G. Garay, E. I. Gordo, F. A. Fano, and M. L. Sukia, "Intelligent multi-objective nonlinear model predictive control (imo-nmpc): Towards the on-line optimization of highly complex control problems," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6527 – 6540, 2012.

[93] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182 –197, apr 2002.

[94] K. Deb, *Multi-objective optimization using evolutionary algorithms*. England: John Wiley & Sons, Ltd., 2001.

[95] K. Binmore, A. Rubinstein, and A. Wolinsky, "The nash bargaining solution in economic modelling," *The RAND Journal of Economics*, pp. 176–188, 1986.

[96] E. Van Damme, "The nash bargaining solution is optimal," *Journal of Economic Theory*, vol. 38, no. 1, pp. 78–100, 1986.

[97] S. Yu, C. Bohm, H. Chen, and F. Allgower, "Stabilizing model predictive control for lpv systems subject to constraints with parameter-dependent control law," in *American Control Conference, 2009. ACC '09.*, June 2009, pp. 3118–3123.

[98] A. Hajiloo, W. Xie, and X. Ren, "Multi-objective robust model predictive control using game theory," in *2015 IEEE International Conference on Information and Automation*, Aug 2015, pp. 2026–2030.

[99] W. M. Haddad and V.-S. Chellaboina, "Nonlinear fixed-order dynamic compensation for passive systems," in *American Control Conference, 1997. Proceedings of the 1997*, vol. 3. IEEE, 1997, pp. 2160–2164.

[100] S. Dussy and L. El Ghaoui, "Measurement-scheduled control for the rtac problem: an lmi approach," *International journal of robust and nonlinear control*, vol. 8, no. 45, pp. 377–400, 1998.

[101] B. Gao, C. Liu, and H. Cheng, "Virtual constraints based control design of an inclined translational oscillator with rotational actuator system," *Shock and Vibration*, vol. 2015, 2015.

[102] A. Choukchou-Braham, B. Cherki, M. Djemaï, and K. Busawon, "Control design schemes for underactuated mechanical systems," in *Analysis and Control of Underactuated Mechanical Systems*. Springer, 2014, pp. 55–91.

[103] R. J. Kinsey, D. L. Mingori, and R. Rand, "Nonlinear controller to reduce resonance effects during despin of a dual–spin spacecraft through precession phase lock," in *Proceedings of the 31st IEEE Conference on Decision and Control*, 1992, pp. 3025–3030 vol.4.

[104] R. T. Bupp, D. S. Bernstein, and V. T. Coppola, "A benchmark problem for nonlinear control design," *Int. J. Robust Nonlinear Control*, vol. 307–310, p. 310, 1998.

[105] P. Baranyi, D. Tikk, Y. Yam, and R. J. Patton, "From differential equations to pdc controller design via numerical transformation," *Computers in Industry*, vol. 51, no. 3, pp. 281–297, 2003.

[106] H. O. Wang, K. Tanaka, and M. Griffin, "Parallel distributed compensation of nonlinear systems by takagi-sugeno fuzzy model," in *Proceedings of 1995 IEEE International Conference on Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium.*, 1995, pp. 531–538.