Optimization-Based Control Methodologies with Applications to Autonomous Vehicle

Behnam Gholitabar Omrani

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Mechanical Engineering) at

Concordia University

Montréal, Québec, Canada

August 2010

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis proposal prepared

By: **Behnam Gholitabar Omrani**

Entitled: **Optimization-Based Control Methodologies with Applications to Autonomous Vehicle**

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

Master of Applied Science (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 committee:

_____ Dr. Lyes Kadem, Chair

_____ Dr. Peter Grogono, External examiner

_____ Dr. Chun Yi Su, Examiner

_____ Dr. Luis Rodrigues, Co-Supervisor

_____ Dr. Camille A. Rabbath, Co-Supervisor

Approved by _____

Chair of the Mechanical and Industrial Eng. Dept.

_____

Dean of Engineering

ABSTRACT

Optimization-Based Control Methodologies with Applications to Autonomous Vehicle

Behnam Gholitabar Omrani

This thesis includes two main parts. In the first part, the main contribution is to develop nonsingular rigid-body attitude control laws using a convex formulation, and implement them in an experimental set up. The attitude recovery problem is first parameterized in terms of quaternions, and then two polynomial controllers using an SoS Lyapunov function and an SoS density function are developed. A quaternion-based polynomial controller using backstepping is also designed to make the closed-loop system asymptotically stable. Moreover, the proposed quaternion-based controllers are implemented in a Quanser helicopter, and compared to the polynomial controllers and a PID controller experimentally.

The main contribution of the second part of this thesis is to analytically solve the Hamilton-Jacobi-Bellman equation for a class of third order nonlinear optimal control problems for which the dynamics are affine and the cost is quadratic in the input. One special advantage of this work is that the solution is directly obtained for the control input without the computation of a value function first. The value function can however also be obtained based on the control input. Furthermore, a Lyapunov function can be constructed for a subclass of optimal control problems, yielding a proof certificate for stability. Using the proposed methodology, experimental results of a path following problem implemented

in a Wheeled Mobile Robot (WMR) are then presented to verify the effectiveness of the proposed methodology.

Dedicated to

my Papa and Mom

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisors, Dr. Luis Rodrigues and Dr. Camille Alain Rabbath. This thesis could not have been accomplished without their guidance and support. I am very grateful to my supervisors for giving me this opportunity to come to Concordia University and join their research group. This helped me a lot to experience a new French-Canadian life in the bilingual, multicultural city of Montreal with its lovely culture and people.

My Special heartfelt thanks go to all my friends in Montreal. I had the honor of sharing the same space and time both in and outside the lab with these fantastic people. I would like to thank Miad Moarref, Arash Ashkan Alam, Hojat Izadi, Sina Nabavi, Sanaz Oliaie, Scott Casselman, Nastaran Nayebpanah, Mehdi Abedinpour, Behzad Samadi, Giancarlo Luglio, Kyungjae Baik, Ralph Koyess, Gavin Kenelly, Patrick Demers-Stoddart, Camilo Ossa, Alejandro Celis, Felipe Jaramillo, and Felipe Gomez. I would also like to Thank Miad Moarref and Gilles Huard for their help on the experimental setups. Last but not least, I would like to tank my parents and my two brothers for their unconditional and incredible love and support.

TABLE OF CONTENTS

1 Introduction **1**

1.1 Literature Review 1

1.1.1 Attitude Control of a Rigid Body 1

1.1.2 Optimal Control Problems: An Inverse Optimality Approach . . . 7

1.2 Contributions of the Thesis 10

1.3 Structure of the Thesis 11

2 Preliminaries and Experimental Setup **13**

2.1 Attitude Kinematics and Dynamics of a Rigid Body 13

2.1.1 Reference Frames 14

2.1.2 Attitude Kinematics 16

2.1.3 Quanser Helicopter 21

2.1.4 Mathematical Preliminaries 22

2.2 Path Following Control Problem 24

2.2.1 Wheeled Mobile Robot (WMR) Experimental Setup 25

3 Large Attitude Control of Rigid Bodies Using Quaternions **30**

3.1 Introduction 30

3.2 Background 31

3.3 Preliminaries on Attitude Control Problem 32

3.3.1 Why Quaternions? 33

3.3.2 Attitude Control Problem Definition 34

3.3.3 Sum of Squares (SoS) Decomposition Method 36

3.4 Attitude Control Problem Solution 38

3.4.1 SoS Lyapunov Based Control 38
3.4.2 SoS Density Function-Based Control 41
3.4.3 Backstepping Approach for MIMO Nonlinear System 44
3.5 Simulations and Experiments 47
3.5.1 Numerical Simulation 47
3.5.2 Experimental Results on Quanser Helicopter 55
3.6 Summary 58
3.7 Appendix 60

4 An Inverse Optimality Approach To A Third Order Optimal Control Problem 65

4.1 Introduction 65
4.2 Background 66
4.3 Optimal Control Problem Definition and Solution 67
4.4 Examples and Numerical Simulations 73
4.5 Experimental Results 84
4.6 Summary 88

5 Conclusions 90

LIST OF FIGURES

1.1 Structure of the Thesis 11

2.1 Illustration of the orbital F_o and inertial F_i reference frames 14
2.2 Euler Angles (roll ϕ, pitch θ, and yaw ψ) 16
2.3 Quanser Helicopter of HYCONS Laboratory in Concordia University . . 22
2.4 Wheeled Mobile Robot (WMR) 24
2.5 HYCONS Wheeled Mobile Robot 26
2.6 Experimental structure of Wheeled Mobile Robot 27
2.7 Xbee wireless communication modules and Arduino Atmega328 board . . 28
2.8 Inertial Measurement Unit and Xbee wireless modules installed on IMU . 28
2.9 Security camera (left) and rechargeable lipo battery (right) 29
2.10 WMR Identification 29

3.1 Time response of quaternions and associated constraint 54
3.2 Time response of Angular velocities 55
3.3 Time response of control inputs 56
3.4 Time response of control inputs 57
3.5 Time response of Euler angles using Quaternions and MRP parameters . . 58
3.6 Comparison of Pitch and Control Input for Quanser Helicopter 59

4.1 Time response of the states, $u(x)$, $L(x)$, and $V(x)$ for example 4.4.2 77
4.2 Comparison of LQR and HJB-based controllers for Example 4.4.2 79
4.3 Time response of states and control inputs with 85% loss in control authority 80
4.4 WMR Trajectories 82
4.5 Region of Attraction for the given WMR 83

4.6 Time response of the states,$u(x)$,$L(x)$, and $V(x)$ for path following problem 85
4.7 Control input and trajectory of the WMR path following 86
4.8 Experimental WMR Trajectories 87
4.9 Experimental control inputs for the WMR 88

LIST OF TABLES

2.1 WMR Identification Table 28

3.1 Max. overshoot and settling time due to 25 (deg) initial pitch angle 58

LIST OF SYMBOLS

Variables

Symbol	**Unit**	**Definition**
q_i	$-$	Quaternions
W	rad/s	Angular Velocity Vector
ϕ	rad	Roll Angle
θ	rad	Pitch Angle
ψ	rad	Yaw (heading) Angle
x,y,z	m	x,y,z Position
m	kg	Mass
I	$kg-m^2$	Moment of Inertia
$g = 9.81$	m/s^2	Acceleration due to Gravity

Symbol	**Mathematical Definition**
$>(<)$	Positive (Negative) Matrix
$\geq(\leq)$	Positive (Negative) Semidefinite Matrix
∇(V)	Gradient of a Scalar Valued Function ($V(x):\mathscr{R}^n \rightarrow \mathscr{R}$)
∇.(f)	Divergence of a Vector Field ($f(x):\mathscr{R}^n \rightarrow \mathscr{R}^n$)

Acronyms

Label	**Definition**
MRP	Modified Rodrigues Parameter
SoS	Sum of Squares
CLF	Control Lyapunov Function
WMR	wheeled mobile robot
PWM	Pulse-Width Modulation
IMU	Inertial Measurement Unit
HJB	Hamilton-Jacobi-Bellman
DOF	Degrees Of Freedom
LMI	Linear Matrix Inequality
ECI	Earth-Centered Inertial Frame
ECEF	Earth-Centered Earth-Fixed Frames
SDP	Semidefinite Program
MIMO	Multi-Input Multi-Output
ROA	Region of Attraction
3D	3-Dimension

Chapter 1

Introduction

This chapter includes a review of the relevant literature on two main topics of the thesis: the attitude control of a rigid body and inverse optimality method. The main contributions and the structure of this thesis are also stated in this chapter.

1.1 Literature Review

This section will be broken into two subsections. The first part presents a review of the relevant literature on the attitude control of a rigid body, and the second part will review the literature of inverse optimality approach.

1.1.1 Attitude Control of a Rigid Body

An attitude recovery maneuver is used when a malfunction affects the attitude of the rigid body and throws it into a spin. The primary task of the attitude control system is to stabilize the attitude of the rigid body, specially satellite, against external torque disturbances generated by aerodynamic drag effects, solar radiation, unwanted wind torques, a sudden seizure of a momentum wheel, and so on. In most rigid body applications such as

satellite, spacecraft, robot manipulators and high performance air vehicles, large angle maneuvers are required to be performed for different missions. To this aim, an attitude recovery should be implemented to bring back the rigid body to the zero attitude state vector, subject to any initial condition. To fully simulate an attitude problem, a rigid body is characterized by nonlinear attitude kinematics.

Attitude dynamics and its control has been an important topic in the control field since the first humans made an artificial satellite, Sputnic *I*, which was build and launched on October 4^{th}, 1957. The actual numbers of journal and conference papers, technical reports and books published in this area is in hundreds and quite overwhelming. Therefore, the literature review given in the following paragraphs will give a brief review of attitude control of a rigid body.

Several research studies have been conducted in the past few decades that investigate attitude control problem using a variety of control techniques ranging from the classical PID [1], [2] and [3], and feedback linearization control [18] , to adaptive and optimal control [4], [5] and [6], and intelligent-based attitude control approaches such as neural networks [7] and [8], and fuzzy logic [9], [10] and [11]. The attitude control problem was first developed by Meyer [12] and [13], and then was extended by several researchers. Using a Lyapunov approach, Meyer [13] focused on appropriate attitude representations of spacecraft dynamic models. In [14] Crouch extends Meyer's work, and presents necessary and sufficient conditions for the controllability of a spacecraft in the case that the gas jet actuators yield one, two, or three independent torques.

A general framework for the analysis of the attitude tracking control problem for a rigid body is presented in [15], where a large family of globally stable control laws are obtained by using the globally nonsingular quaternion representation in a Lyapunov function candidate. In [16] the rigid body attitude control problem with external torques is transformed into an equivalent linear form implementable by three double integrators.

The Linearizing transformations themselves are formulated in vector algebra, requiring no integrators for implementation. Tsiotras in [17] applies a Lyapunov function that includes a sum of quadratic and logarithmic terms in the angular velocities and kinematic parameters resulting in a linear control design. The problem of the attitude recovery of flexible spacecraft is also investigated in [18] and [19] using the feedback linearization control and generating the control error signal based on the quaternion addition.

Recently, backstepping approach, sliding mode control, nonlinear H_∞ control, optimal and adaptive control have also been applied to attitude control problems. Backstepping control approach is mostly used in attitude problems, due to its remarkable capability in designing cascaded systems. The advantage of integrator backstepping compared with other control methods lies in its design flexibility, due to its recursive use of Lyapunov functions. The main concept of backstepping control has been examined in general in [20] and [21], and then has been utilized in several attitude control problems. For example, [22] proposes a robust nonlinear attitude control method for aircraft based on partitioned backstepping. Reference [23] presents a solution to the problem of controlling relative attitude in a leader-follower spacecraft formation, with focus on optimality in rotation path for the follower spacecraft. References [24] and [25] focus on a backstepping approach for controlling the attitude of the European Student Earth Orbiter (ESEO). In these papers a tracking controller is presented to stabilize the attitude of a micro satellite via integrator backstepping and quaternion feedback. The backstepping approach was also applied to attitude control of satellites in [28], [29] and [30].

Sliding mode control is also one of the most important approaches to handle the attitude control problems with large uncertainties, nonlinearities, and bounded external disturbances. The main drawback of the sliding mode control is its discontinuous switching control law (sign function) which results in chattering. In [31], [32], [33], and [34] sliding mode controller have been investigated for attitude control problem in term of

Euler angles, Rodrigues parameters, Modified Rodrigues parameters, and Quaternions, respectively. The most recent work in [35] also studies two optimal sliding mode control laws using integral sliding mode control (ISM) for some spacecraft attitude tracking problems. In this paper, integral sliding mode control combining the first order sliding mode and optimal control is applied to quaternion-based spacecraft attitude tracking maneuvers with external disturbances and an uncertainty inertia matrix.

Using a control Lyapunov function approach, [36] designs globally stabilizing feedback laws that have desirable optimality with respect to cost functions, penalizing state errors and control effort. Their performance is also compared to the performance of previously developed proportional-derivative type control laws. It is shown that the new control laws achieve the same or greater stabilization rate with less control effort. In [37] a discrete optimal control problem for attitude dynamics of a rigid body with symmetry, applied to a 3D pendulum, is presented. The symmetry in the attitude dynamics system yields a conserved quantity, causing a fundamental singularity in optimal control problems. Using an inverse optimal adaptive Control, the attitude tracking control problem of a rigid body with external disturbances and an uncertain inertia matrix is addressed in [38]. This is achieved by the inverse optimality approach without solving the associated Hamilton JacobiIsaacs partial differential equation directly. In [6] a nonlinear optimal controller has been devised for the attitude tracking problem of spacecraft maneuvers through HamiltonJacobi formulation, applying a penalty on angular velocities and attitudinal kinematics. Reference [39] presents attitude control of a satellite using a statistical game (Minimal Cost Variance) control. Throughout the simulations, statistical game control has an extra degree of freedom to improve the performance, and reduce the overshoot compared to either H_1 control and H_2/H_1 control.

Nonlinear adaptive control is also one of the recent control approaches in attitude

control problem. Adaptive control method is a natural choice to manipulate uncertain parameters and has been applied to the attitude tracking control problem of spacecraft [40]. For instance, [41] presents a nonlinear adaptive control law for the attitude control of satellites using gyro torquers such that large rotational maneuvers can be performed. The problem of adaptive attitude tracking control for a rigid spacecraft with uncertain inertia matrix is addressed in [42] and [43]. Using MRP attitude representation and the backstepping approach, the adaptive attitude tracking control problem for a rigid spacecraft subject to inertia uncertainty is investigated in [42]. In [43] a nonlinear adaptive control law based on a backstepping design technique is derived for the control of the pitch angle.

In nonlinear optimal control theory, nonlinear H_∞ control method is a potential approach to the attitude control problem with external disturbances. To use H_∞ approach, a control problem is expressed as a mathematical optimization problem, where the desired controller is obtained by solving this optimal problem. H_∞ control mainly includes two issues. The first issue is to make a given system stable and the second one is to ensure that the L_2-gain, from the disturbance input to the controlled output, of the closed-loop system is not larger than a certain value [44]. However, the main drawback of this method is the difficulty in solving the associated Hamilton-Jacobi-Isaacs (HJI) partial differential equation, although there have been a few numerical approaches to solve the HJI equation. A linear H_∞-control method based on the linearization of a space station model is used in [45]. In [46] a state feedback H_∞-suboptimal control problem for a rigid spacecraft with three control torques and disturbances is addressed. The Hamilton-Jacobi inequality associated with a corresponding state feedback H_∞-suboptimal control problem is globally solved in this paper. Reference [47] extends the results of [45] and [46] to the attitude tracking control problem of a rigid spacecraft with external disturbances. Using the inverse optimal control method, it is shown that a nonlinear H_∞ optimality with respect to

the extended disturbance is achieved without obtaining a direct solution to the HJI equation.

The aforementioned approaches so far are mainly based on Lyapunov and storage functions for analysis. The main drawback of these approaches is that finding or constructing a Lyapunov function is not trivial, and there does not exist a general systematic method to find a Lyapunov function for a given system. Therefore, finding or constructing a Lyapunov function is inevitably restricted to some specific structure of known systems with small state dimensions. For a general nonlinear system $\dot{x} = f(x)$, in the case in which both vector field f and the Lyapunov function candidate V are polynomial, the Lyapunov conditions are basically polynomial non-negativity conditions, which can be NP hard to check [48]. However, most recently a new computationally efficient nonlinear method using sum of squares (SoS) approach was proposed by Parrilo [49]. Using this approach, the non-negativity conditions are relaxed to SoS certificate functions (of appropriate polynomials) in the form of semidefinite programming (SDP) (see 3.3.3 for more detail). Therefore, using SoS approach, not only the Lyapunov conditions are checked, but also a suitable Lyapunov function can be constructed. To convert the SoS decomposition problem to the corresponding SDP formulation, a freely-available MATLAB toolbox, the software SOSTOOLS [50] has been developed. This recent approach has so far been used for several applications including aircrafts [51]−[52], satellites [54], and Robots [55].

Two of the most recent approaches in nonlinear control are SOS Lyapunov based controller [56] and SOS density function based controller [57]. First and foremost, the key idea that enables us to utilize SoS in solving the attitude problem is that the rigid body model using either quaternions or MRP can be represented by polynomial vector fields [54]. For a general nonlinear system $\dot{x} = f(x) + g(x)u$, where $f(x)$ and $g(x)$ are polynomials, searching for a control Lyapunov function and a controller simultaneously is not a convex problem. However, using a so-called Density function $\rho(x)$ [57] leads to a

convex formulation. Moreover, for a nonlinear system in the form of $\dot{x} = f(x)x + g(x)u$, an SOS Lyapunov based controller can also be used to find a polynomial controller satisfying conditions of the Lyapunov's stability theorem [20]. As the most recent work in [54], an SOS Lyapunov based control has been used to design a polynomial controller for a rigid-body attitude problem, using Modified Rodrigues Parameters (MRP).

1.1.2 Optimal Control Problems: An Inverse Optimality Approach

The sufficient condition for solving an optimal control problem is to find the solution of the Hamilton-Jacobi-Bellman (HJB) equation. There is no systematic analytical solution at present for HJB equation, which is a nonlinear partial differential equation. Therefore, finding a value function that satisfies the HJB equation for a nonlinear system is quite challenging. Avoiding solving the HJB directly, inverse optimality is an alternative method to solve the nonlinear optimal control problem. The inverse optimal problem is different from the direct one in the point that the latter seeks a controller which minimizes a given cost, while the former is concerned with finding a controller which minimizes some meaningful cost dependent on the controller [69]. Using inverse optimality approach, it can be shown that a controller $u(x)$ is inverse optimal with respect to a cost functional

$$J(x,u) = \lim_{t\to\infty}\left\{\int_0^t \left(l(x) + u^T R(x)u\right)\right\} \tag{1.1}$$

where x and u are the state vector and the control input vector, respectively, $l(x)$ is positive definite and radially unbounded, and $R(x)$ is a positive definite matrix for all x. Inverse optimal control method not only finds a stabilizing control law, but also determines $l(x)$ and $R(x)$ yielding a meaningful cost function. Therefore, the inverse optimal problem is easier than the direct one in which $l(x)$ and $R(x)$ are given, and also where one has to solve an HJI partial differential equation.

Optimal control problems and inverse optimality have been studied in the sixties

focusing mostly on linear quadratic problems driven by aerospace applications (see for example [70] and [71]). Nonlinear optimal control problems based on the concept of the inverse optimality have been revisited by several researchers such as [72]-[75] and [21]. In terms of applications, [72] presents an inverse optimal control approach for regulation of a rotating rigid spacecraft by solving a HJB equation. The resulting design includes a penalty on the angular velocity, orientation, and the control torque, where the weight in the penalty on the control depends on the current state and decreases for states away from the origin. Inverse optimal stabilization of a class of nonlinear systems is also investigated in [73] resulting in a controller optimal with respect to a meaningful cost function. The main drawback of the inverse optimality approach used in [72] and [73] is that the one requires the knowledge of a control Lyapunov function and a stabilizing control law of a particular form.

Focusing on the inverse optimal control of nonlinear systems with a structural uncertainty, [76] derives a Lyapunov-based theorem for a globally asymptotic stability which yields a less conservative condition for the inverse optimal control problem. In [74] an optimal feedback controller for bilinear systems is obtained that minimizes a quadratic cost function. The proposed inverse optimal control design is also applied to the problem of the stabilization of an inverted pendulum on a cart with horizontal and vertical movement, where the control performance of the system can be easily tuned using the proposed quadratic cost function.

Using a control Lyapunov function (CLF) and Sontag's formula, inverse optimal tracking control is experimentally applied to a nonholonomic mobile robots with two actuated wheels and an autonomous surveillance aerial blimp in [69] and [77], respectively. The proposed optimal controllers minimizing a meaningful cost function guarantee the robustness of these systems with respect to large uncertainties. In [78] an inverse optimal adaptive controller, based on a Lyapunov analysis, is developed to asymptotically

minimize a meaningful performance index. Using the resulting adaptive controller based on inverse optimality, the generalized coordinates of a nonlinear Euler-Lagrange system asymptotically track a desired time-varying trajectory despite LP (linear in the parameters) uncertainty linear in the dynamics. A Lyapunov analysis is also provided to derive a cost functional with a positive integrand that penalizes the states and control, and has a terminal penalty on the parameter estimation error.

A sufficient condition for an optimal control problem is to find the solution to a Hamilton-Jacobi-Bellman (HJB) equation [79], which is a nonlinear partial differential equation and difficult to solve analytically. Therefore, Optimal control problems are generally solved by numerical techniques. However, there is an explicit solution for a given general format of the control input as a derivative of the value function if the dynamic model is affine and the cost is quadratic in the input. This idea was first used in [80] to solve a class of second order problems, and will then be extended to a class of third order optimal problems in chapter 4.

The experimental motivation of this theoretical work comes from the dynamics model of a Wheeled Mobile Robot (WMR) on the $x-y$ plane for path following of the straight line $y=0$ at a constant velocity, as shown in $Fig.2.4$. Given this nonlinear system, our interest is to simultaneously search for a general form of a control input, in terms of the states and the nonlinear term $sin(\psi)$, and a function $Q(x)$ that together satisfy the HJB equation. Therefore, it is assumed that the cost function is the sum of a quadratic term in the input and the states and an unknown term $Q(x)$ that should be determined. Moreover, the resulting value function will also be a local Lyapunov function that proves the asymptotic stability of the WMR dynamic model. This path following problem and its experimental setup is discussed in more detail in Chapter 2. The proposed method in this paper is more general for a class of third order nonlinear systems, where the path following problem is an special case of our optimal control.

Departing from previous methods, the proposed method in this thesis can directly find a solution for the control input without the computation of a value function (see Chapter 4 for more detail). The value function can however also be obtained based on the control input. Furthermore, a Lyapunov function can be constructed for a subclass of optimal control problems, yielding a proof certificate for stability. The proposed methodology will then be applied to the dynamic model of a Wheeled Mobile Robot (WMR) on the $x-y$ plane for path following of the straight line $y=0$ at a constant velocity.

1.2 Contributions of the Thesis

The main contributions of this thesis are the following:

- To develop nonsingular rigid-body attitude control laws using a convex formulation, and implement them in an experimental set up. The thesis proposes and compares, both numerically and experimentally, two Sum of Squares (SoS)-based controller design approaches for large attitude recovery of rigid bodies. The proposed quaternion-based controllers are also implemented in a Quanser helicopter, and compared to the polynomial controllers and a PID controller experimentally.

- To analytically solve the Hamilton-Jacobi-Bellman equation for a class of third order nonlinear optimal control problems for which the dynamics are affine and the cost is quadratic in the input. The proposed solution method is based on the notion of inverse optimality with a variable part of the cost to be determined in the solution. One special advantage of the proposed method is that the solution is directly obtained for the control input without the computation of a value function first. The value function can however also be obtained based on the control input. Furthermore, a Lyapunov function can be constructed for a subclass of optimal

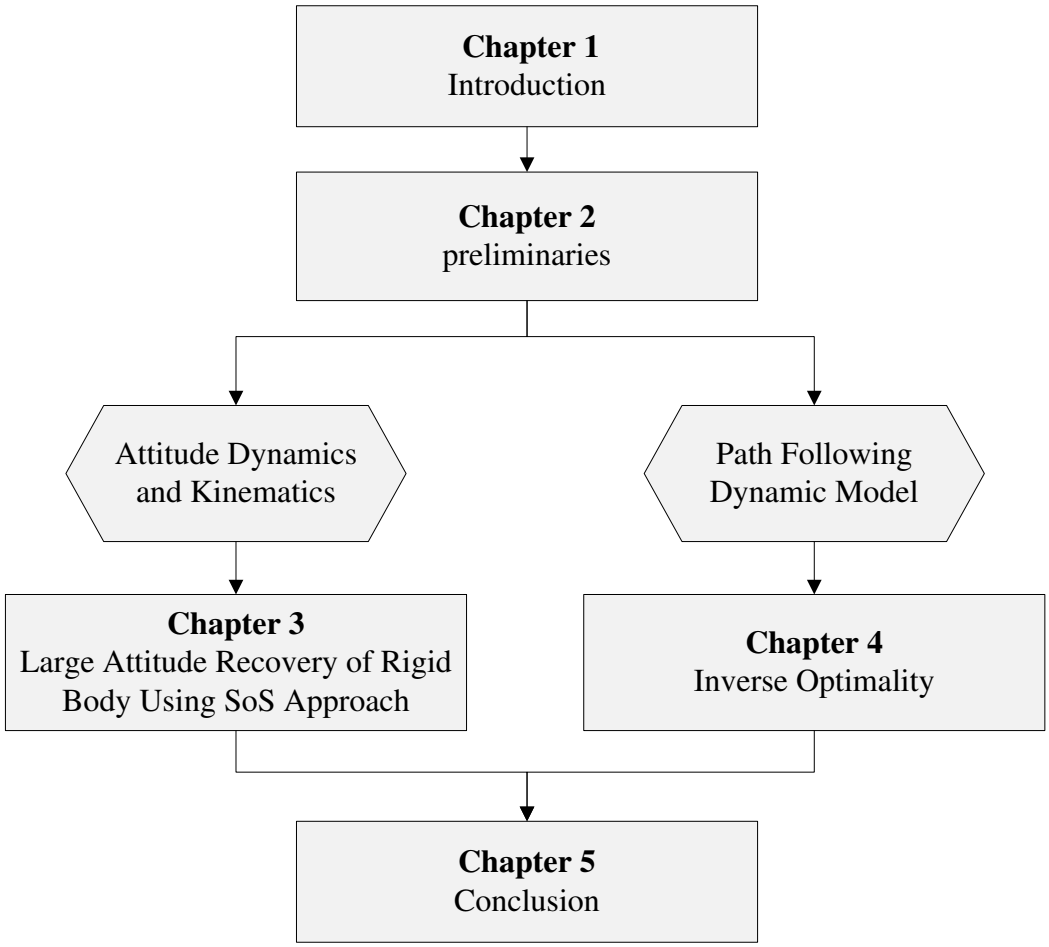

Figure 1.1: Structure of the Thesis

control problems, yielding a proof certificate for stability. The proposed approach is also implemented in a Wheeled Mobile Robot (WMR) for path following of a line to experimentally verify the effectiveness of this methodology.

1.3 Structure of the Thesis

The thesis is structured as shown in *Fig*.1.1. In Chapter 2, a brief review of the kinematics equations of motion for a rigid body is given. The experimental setup for 1-DOF model of a Quanser helicopter and path following of a Wheeled Mobile Robot (WMR) are also

explained. Moreover, the dynamic model for path following of the straight line $y = 0$ of a WMR on a plane is stated in this chapter. Next, the rigid-body attitude problem is first parameterized in terms of quaternions. Then polynomial controllers based on an SoS Lyapunov function, an SOS density function and a backstepping controller are proposed to make the closed-loop system asymptotically stable. A practical application implemented in a Quanser helicopter is also presented to verify the numerical simulation results in Chapter 3. Subsequently, using an inverse optimality method a class of third order nonlinear optimal control problems is analytically solved in Chapter 4. A practical application to a WMR path following problem is also presented to experimentally verify the effectiveness of the proposed methodology. Finally, conclusions are drawn in Chapter 5. Chapter 4, and part of Chapter 2 are mainly based on the following paper:

- Behnam Gholitabar Omrani, Camille Alain Rabbath, and Luis Rodrigues, "An Inverse Optimality Method to Solve a Class of Third Order Optimal Control Problems", accepted to be published in the *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, Georgia, December 15-17, 2010

Chapter 2

Preliminaries and Experimental Setup

This chapter includes two main sections. In the first section,kinematics equations of motion for a rigid body is described using different common representations such as quaternions, Modified Rodrigues Parameter (MRP) and Euler angles. The attitude dynamics and experimental set up of a 1-DOF rigid body for Quanser helicopter is then described. In the second section, the dynamic model for path following problem of a Wheeled Mobile Robot (WMR) is given. A system identification and an experimental setup of a Wheeled Mobile Robot (WMR) is also presented.

2.1 Attitude Kinematics and Dynamics of a Rigid Body

In this section, reference frames are first defined, and then kinematics equations of motion for a rigid body using different representations are briefly discussed. Moreover, a brief review of the gradient and the divergence properties is given, which will be used throughout this thesis.

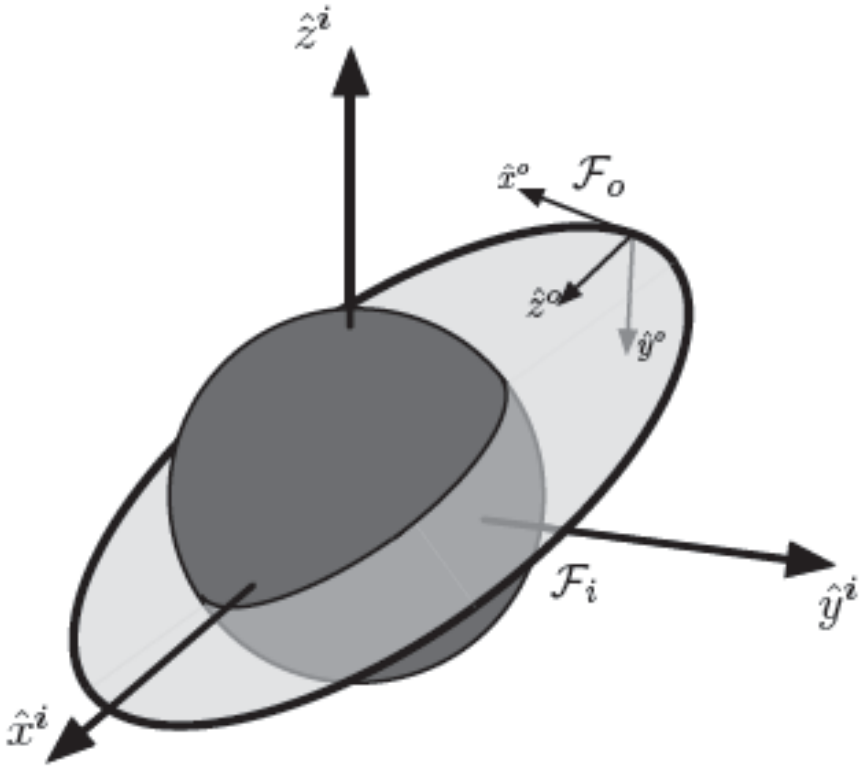

Figure 2.1: Illustration of the orbital F_o and inertial F_i reference frames (adopted from [66])

2.1.1 Reference Frames

Since attitude dynamics refers to the orientation of one reference frame with respect to another due to external forces and torques, the definition of reference frames, or coordinate systems, are important. To fully describe an attitude, a set of reference frames are defined here. The most common reference frames used for describing the attitude of a rigid body, specially satellites, are the inertial frame, the orbital frame, the body frame, and the principal axis frame [66].

Inertial Frame

An inertial frame is a non-rotating reference frame in a fixed space. A common representation of an inertial frame is Earth-Centered Inertial (ECI) frame, in contrast to the Earth-centered Earth-fixed (ECEF) frames which rotate in an inertial space in order

to remain fixed with respect to the surface of the Earth. ECI frame is illustrated in *Fig.* (2.1). The $\hat{i}_x$ axis points from the center of the Earth to the vernal equinox, the $\hat{i}_z$ axis is aligned with the Earth's rotation axis and perpendicular to the equatorial plane, and $\hat{i}_y$ is in the equatorial plane completing a right-hand triad. The hats also denote unit vectors.

Orbital Frame

The orbital frame is a non-inertial frame attached to the center of mass of the rigid body, and moves with the body in orbit. The motion of the frame depends only on the orbit and is not effected by body rotations. As illustrated in *Fig.* 2.1, the $\hat{O}_z$ axis points the direction from the spacecraft to the Earth (nadir direction), $\hat{O}_y$ is the direction opposite to the orbit normal, and $\hat{O}_y$ completes the orthonormal triad to $\hat{O}_z$ and $\hat{O}_x$. Note that this frame is non-inertial because of orbital acceleration and the rotation of the reference frame.

Body Frame

A body frame has its origin at the center of a rigid body. Since this frame is fixed to the rigid body, it is a non-inertial frame. Body frames are useful for relating objects on a rigid body relative to one another. It also describes how a rigid body is oriented with respect to an external frame (such as the orbital or inertial frames).

Principal Axis

This frame is a specific body-fixed reference frame with the axes aligned such that the moment of inertia matrix is diagonal. These moments of inertia are called the principal moments of inertia. In dynamic modeling, it is useful to describe the system in the principal axes frame.

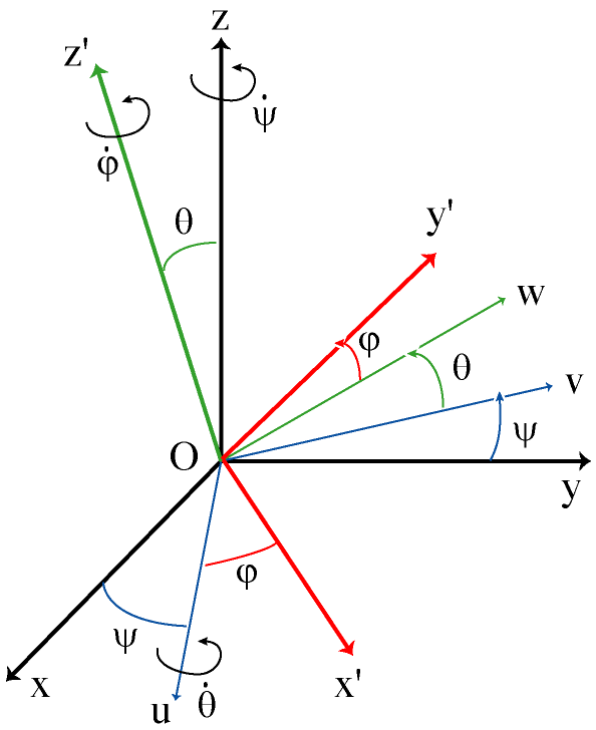

Figure 2.2: Euler Angles (roll ϕ, pitch θ, and yaw ψ) [62]

2.1.2 Attitude Kinematics

There are several common ways to describe the attitude of a rigid body like Direction cosine matrix, Euler axis and angle, Modified Rodrigues Parameter (MRP), Euler angles, and quaternions [65]. The three commonly used representations used in a rigid body attitude control is briefly discussed here: Euler Angles, Modified Rodrigues Parameters, and Quaternions.

Euler Angles

The Euler angle rotation is defined as successive angular rotations about the three orthogonal frame axes. The first rotation is about any axis. The second rotation is about either of two axes not used for the first rotation, and the last rotation is about either of two axes not used for the second rotation. There are totally 12 sets of order combination by which the rotation can be performed. However, It is common to define the Euler roll angle

(ϕ) about the x body axis, the pitch angle (θ) about the y body axis, and the yaw angle (ψ) about the z body axis. Note that the transformation from one reference frame to another is non-unique, and also that any other definition is acceptable as long as it follows the correct order of the rotations. Suppose we will perform the orientation of a body frame F_b relative to a fixed inertial frame F_i using the transformation $\psi \rightarrow \theta \rightarrow \phi$ successively about the z, y, and x body axes, as shown in $Fig.$ 2.2. The corresponding principal rotation matrices are expressed in the following matrix form [65]

$$R_x(\phi)\begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & S\phi \\ 0 & -S\phi & C\phi \end{bmatrix}, R_y(\theta)\begin{bmatrix} C\theta & 0 & -S\theta \\ 0 & 1 & 0 \\ S\theta & 0 & C\theta \end{bmatrix}, R_z(\psi)\begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

where $S\alpha = Sin(\alpha)$, and $C\alpha = Cos(\alpha)$. For this transformation, the rotation matrix will be described by

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \left[R_\psi\right]\left[R_\theta\right]\left[R_\phi\right]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \left[R_{\psi\theta\phi}\right]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.2}$$

where

$$\left[R_{\psi\theta\phi}\right] = \begin{bmatrix} C\theta C\psi & C\theta S\psi & -S\theta \\ -C\phi S\psi + S\phi S\theta C\psi & C\phi C\psi + S\phi S\theta S\psi & S\phi C\theta \\ S\phi S\psi + C\phi S\theta C\psi & -S\phi C\psi + C\phi S\theta S\psi & C\phi C\theta \end{bmatrix} \tag{2.3}$$

The roll-pitch-yaw derivatives are then transformed to the body angular rates w_x, w_y, and w_z by the following equation

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \left[R_\phi\right]\left[R_\theta\right]\left[R_\psi\right]\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \left[R_\phi\right]\left[R_\theta\right]\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \left[R_\phi\right]\begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}. \tag{2.4}$$

Finally, the attitude kinematics of a rigid body using Euler angles for the roll-pitch-yaw ($\psi \rightarrow \theta \rightarrow \phi$) transformation is given by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi)sec(\theta) & cos(\phi)sec(\theta) \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}. \tag{2.5}$$

As seen in (2.5) using Euler angles for the representation of attitude kinematics results in singularities at $\theta = \pm 90^{\circ}$, making the Euler angles impractical and inconvenient for describing large angle rotations.

Quaternions

The attitude determination of rigid bodies by use of the quaternion parameters has several advantages over the use of other representations. Instead of trigonometric functions, quaternions uses algebraic relations to determine the elements of the rotation matrix. Moreover, the computations are faster and there are no singularities as may occur in the MRP representation or Euler formulation. Fewer multiplications are also required for propagating successive incremental relations [58]. Thus, using quaternions has a better numerical properties [84]. However, a disadvantage is that one of the four elements is redundant, and that in general there is no obvious physical interpretation of the rotation geometry ([64] and [59]) (see subsection 3.3.1 for more details).

A quaternion is a scalar plus a vector, totaling four elements. While the vector (with three elements) defines an axis of the rotation, the scalar element defines the magnitude of the rotation angle about the axis of the rotation. The formulation is based on Euler's theorem which states that any rotation of a body (or coordinate system) with respect to another can be described by a single rotation through some angle about single fixed axis [59]. The four-element quaternion set, q, can then be determined from the Euler axis and angle, $(\overrightarrow{e}, \lambda)$, where λ denotes the principal angle, and $\overrightarrow{e} = (e_1, e_2, e_3)^T$ denotes the

principal unit vector corresponding to Euler's theorem [64]. The quaternion vector

$$q = \begin{bmatrix} q_o & \widetilde{q}^T \end{bmatrix}^T = \begin{bmatrix} q_o & q_1 & q_2 & q_3 \end{bmatrix}^T \tag{2.6}$$

can then be written as [64]

$$\widetilde{q} = \overrightarrow{e} \ \sin(\frac{\lambda}{2}) \quad , \quad q_o = \cos(\frac{\lambda}{2}), \tag{2.7}$$

where the condition

$$\Omega = |q| = q_o^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

is automatically satisfied, and can be used for numerical control of machine computations. The inverse rotation is also given by the complex conjugate of q as

$$\bar{q} = \begin{bmatrix} q_o & -\widetilde{q}^T \end{bmatrix}^T. \tag{2.8}$$

Note that if q represents a given attitude of a rigid body, then $-q$ represents the same attitude. Therefore, although $q \neq -q$ mathematically, they both represent the same physical attitude [24]. The kinematics equation in terms of quaternions can then be expressed as [66]

$$\dot{q} = \frac{1}{2} \begin{bmatrix} -\widetilde{q}^T \\ q^* + q_o I_{3\times 3} \end{bmatrix} W \tag{2.9}$$

$$q^* = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{2.10}$$

where $W = (W_x, W_y, W_z)^T$ is the vector of the angular velocities of the rigid body. The kinematics equation of the attitude recovery problem in terms of quaternions is finally obtained as

$$\dot{q} = \Gamma(q) W \tag{2.11}$$

where

$$\Gamma(q) = \frac{1}{2}\begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_o & -q_3 & q_2 \\ q_3 & q_o & -q_1 \\ -q_2 & q_1 & q_o \end{bmatrix}. \tag{2.12}$$

Note that the matrix $\Gamma(q)$ is linear, while the polynomial matrix entries using MRP representation $\Omega(\sigma)$ (2.17) are nonlinear in a quadratic form. This indicates that quaternions requires fewer computational operations rather than MRP. Moreover, to convert the Euler angles to quaternions the following conversion algorithm is used

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} cos(\phi/2)cos(\theta/2)cos(\psi/2) + sin(\phi/2)sin(\theta/2)sin(\psi/2) \\ sin(\phi/2)cos(\theta/2)cos(\psi/2) - cos(\phi/2)sin(\theta/2)sin(\psi/2) \\ cos(\phi/2)sin(\theta/2)cos(\psi/2) + sin(\phi/2)cos(\theta/2)sin(\psi/2) \\ cos(\phi/2)cos(\theta/2)sin(\psi/2) - sin(\phi/2)sin(\theta/2)sin(\psi/2) \end{bmatrix}. \tag{2.13}$$

Modified Rodrigues Parameters

Modified Rodrigues Parameters (MRP) is the most recent method of describing a rigid body attitude. MRP is also not a unique representation to the transformation. The MRP vector (σ) is defined by using the principal rotation elements as

$$\sigma = \hat{e}\tan(\frac{\Phi}{4}). \tag{2.14}$$

MRP can also be defined in terms of quaternions elements as

$$\sigma = \begin{bmatrix} q_1/(1+q_o) \\ q_2/(1+q_o) \\ q_3/(1+q_o) \end{bmatrix}. \tag{2.15}$$

As seen in 2.15 MRP has geometry singularities at $\Phi = \pm 360$, which corresponds to $q_o = -1$. Thus, for any rotation more than a complete revolution MRP representation

encounters a singularity. The attitude kinematics of a rigid body in terms of the MRP can be expressed as

$$\dot{\sigma} = \Omega(\sigma)w \tag{2.16}$$

where $w = (w_x, w_y, w_z)^T$ is the vector of the angular velocities of the rigid body about the principal body axes,

$$\Omega(\sigma) = \frac{1}{4}\begin{bmatrix} 1-\sigma^2+2\sigma_1^2 & 2(\sigma_1\sigma_2-\sigma_3) & 2(\sigma_1\sigma_3+\sigma_2) \\ 2(\sigma_2\sigma_1+\sigma_3) & 1-\sigma^2+2\sigma_2^2 & 2(\sigma_2\sigma_3-\sigma_1) \\ 2(\sigma_3\sigma_1-\sigma_2) & 2(\sigma_3\sigma_2+\sigma_1) & 1-\sigma^2+2\sigma_3^2 \end{bmatrix}, \tag{2.17}$$

and $\sigma^2 = \sigma_1^2+\sigma_2^2+\sigma_3^2$. Note also that the polynomial matrix entries for $\Omega(\sigma)$ (2.17) is quadratic with cross terms, which numerically poses more computational challenges rather than a linear matrix. Three different methods of attitude kinematics representations have been discussed in this subsection. Now the attitude dynamics and experimental set up of a 1-DOF rigid body for Quanser helicopter is described in the next subsection.

2.1.3 Quanser Helicopter

The Quanser helicopter [68] is shown in *Fig*.2.3. Using this experimental set up, the objective is to implement the SoS controller synthesis proposed in Chapter 3 for stabilizing the pitch angle of the Quanser helicopter. The quaternion-based attitude parameterization for a one Degree of Freedom (1-DOF) rigid body is the simplified version of (3.2), where $q_1 = q_3 = W_x = W_z = 0$, given by

$$\begin{bmatrix} \dot{q}_o \\ \dot{q}_2 \\ \dot{W}_y \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\frac{q_2}{2} \\ 0 & 0 & \frac{q_o}{2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_o \\ q_2 \\ W_y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{I_y} \end{bmatrix} M_y, \tag{2.18}$$

where I_y for the Quanser helicopter is $0.028(kg.m^2)$. This set up is used to both apply

Figure 2.3: Quanser Helicopter of HYCONS Laboratory in Concordia University [68]

and compare the proposed controllers in Chapter 3. Moreover, It was shown in [68] that the encoder, which measures the pitch angle, works with the stated accuracy of ± 0.0293 degrees. A filter has also been designed by Quanser Inc. to remove any noisy inputs and outputs. See [68] for more detail about the Quanser helicopter setup.

2.1.4 Mathematical Preliminaries

This subsection briefly reviews mathematical preliminaries which will be used throughout this thesis. The gradient (∇V) and the divergence ($\nabla . f$) are defined as follows.

$$\nabla V = \Big[\frac{\partial V}{\partial x_1}, \cdots, \frac{\partial V}{\partial x_n}\Big], \quad V(x) : \mathscr{R}^n \to \mathscr{R} \tag{2.19}$$

$$\nabla . f = \frac{\partial f_1}{\partial x_1} + \cdots + \frac{\partial f_n}{\partial x_n}, \quad f(x) : \mathscr{R}^n \to \mathscr{R}^n \tag{2.20}$$

These two vector mathematical operators also have some basic properties. The divergence is a linear operator, i.e

$$\nabla.\big(aA + bB\big) = a\nabla.A + b\nabla.B \tag{2.21}$$

where (a,b) and (A,B) are real numbers and vector fields, respectively. The divergence operator also satisfies the product rule as follows

$$\nabla.\big(\varphi\, f\big) = \nabla\big(\varphi\big).f + \varphi\big(\nabla.f\big) \tag{2.22}$$

where φ and f are a scalar valued function and a column vector field, respectively. As one of the gradient properties we have

$$\nabla\big(\varphi^{\alpha}\big) = \alpha\varphi^{\alpha-1}\nabla(\varphi) \tag{2.23}$$

where α is a real number. Moreover, given $y = F(x)$ written explicitly as

$$y = \begin{bmatrix} F_1(x) & F_2(x) & \cdots & F_m(x) \end{bmatrix}^T \tag{2.24}$$

where $\big\{F(x) : \mathscr{R}^n \longrightarrow \mathscr{R}^m\big\}$, the jacobian matrix is defined by

$$J(x_1,\cdots,x_n) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}_{m\times n} . \tag{2.25}$$

The symbol $\nabla^2 f(x)$ denotes the Hessian matrix for a scalar valued function $f(x)$ of a state vector $x \in \mathscr{R}^n$, defined as follows

$$\nabla^2 f(x) = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right] = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}_{n\times n} . \tag{2.26}$$

The function $f(x)$ is convex if $\nabla^2 f(x)$ is a positive semidefinite matrix for every $x \in \mathscr{R}^n$. Moreover, the function $f(x)$ is stictly convex if $\nabla^2 f(x)$ is positive definite. We will use the above notation and properties throughout this thesis.

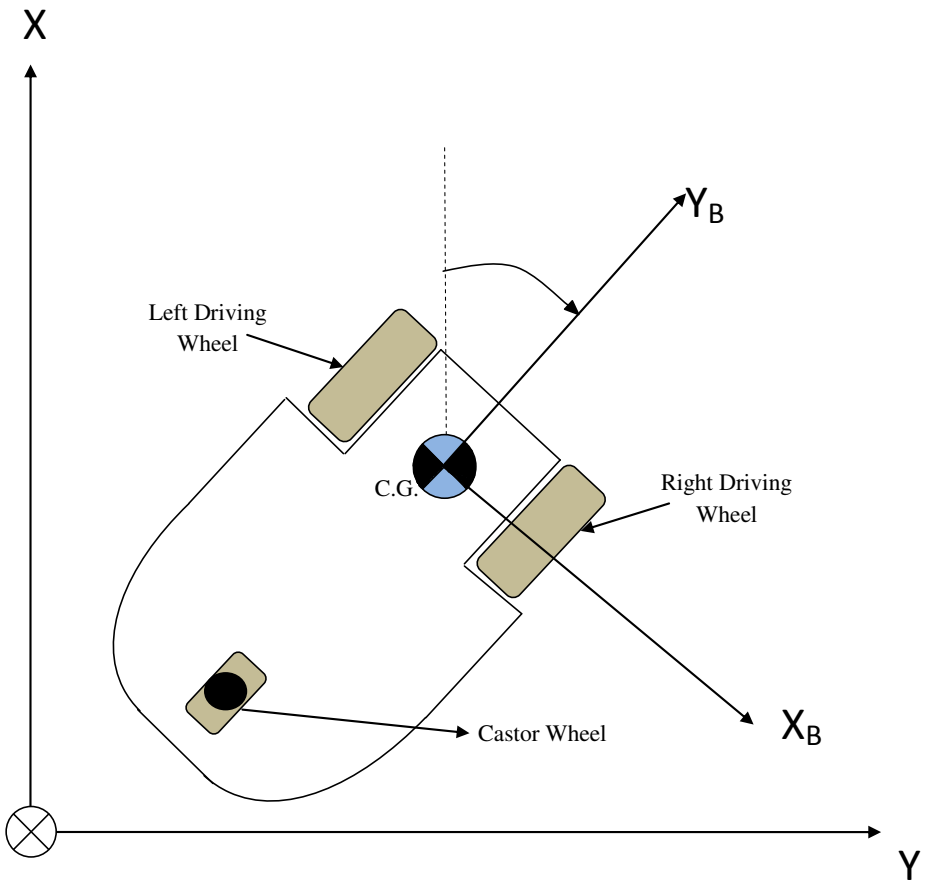

Figure 2.4: Wheeled Mobile Robot (WMR)

2.2 Path Following Control Problem

Path following control problems are primarily concerned with the design of control laws that drive an object, such as robot arm, wheeled mobile robot, ship, aircraft, to reach and follow a specified geometric path, where the time is not important [26]. Note also that Path following is more flexible than reference-tracking, where the vehicle is required to follow a reference signal which is a given function of time. In path following control problem smoother convergence to the path is achieved and the control signals are less likely pushed into saturation, when compared to trajectory-tracking [27]. Thus, the path following control problem is defined as follows.

Definition 2.2.1. *The control objective of the path following problem is to force the output to follow a geometric path without a timing law assigned to it. Therefore, the vehicle is required to converge to and follow a certain path that is specified.*

Consider now the wheeled mobile robot (WMR) shown in *Fig*.2.4, where the center of gravity (*C*.*G*.) of the WMR coincides with the origin of the body frame, located midway between the two driving wheels. The heading angle of the WMR is also given by ψ. The objective is to design a controller for the WMR to follow the straight line $y = 0$ at a constant velocity. The dynamic model for path following of the straight line $y = 0$ of a WMR on the $X - Y$ plane is represented as follows

$$\begin{aligned} \dot{y}(t) &= V \sin(\psi) \\ \dot{\psi}(t) &= \omega \\ \dot{\omega}(t) &= \frac{1}{I_z} u \\ x(0) &= x_0, \quad u \in \mathscr{U} \end{aligned} \tag{2.27}$$

where V is the constant velocity of the WMR, and I_z is the moment of inertia of the WMR for rotation around the z axis. The control input u is also the torque generated about the z-axis. Therefore, the state vector

$$x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T = \begin{bmatrix} y & \psi & w \end{bmatrix}^T \tag{2.28}$$

contains the position y, the heading angle ψ, and the angular velocity $\dot{\psi}$, respectively.

2.2.1 Wheeled Mobile Robot (WMR) Experimental Setup

Fig.2.5 shows the experimental Wheeled Mobile Robot (WMR) in HYCONS lab, Concordia University. The experimental set up includes a camera (*Fig*.2.9), an Inertial Measurement Unit (IMU) 3DM-GX1 (*Fig*.2.8), Xbee wireless communication modules [81], and one Arduino Atmega328 board [82] as well as a WMR and a server computer. The experimental structure of wheeled mobile robot is also shown in *Fig*.2.6, which illustrate how each part of set up communicate with the rest of the system. The camera is

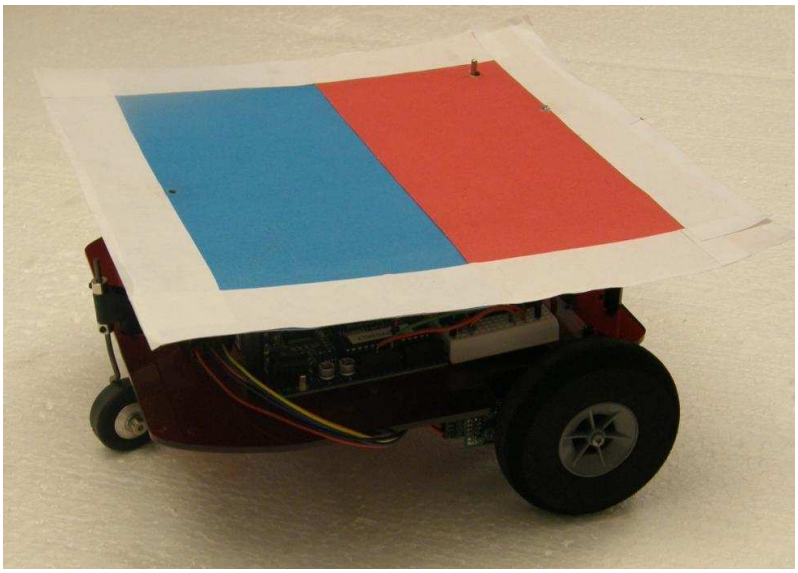

Figure 2.5: HYCONS Wheeled Mobile Robot

directly connected to the computer giving the positions x and y of the WMR after digital image processing [83]. The other two states including the heading angle ψ and the angular velocity ω are measured by the IMU with frequency ($50Hz$), and are then sent to the server wirelessly using the Xbee modules. The server computer processes all the data using *MATLAB* and a *MEX* file (written in $C++$), and then sends the resulting control input 4.51 to the Arduino board ($Fig.2.7$) installed on the WMR and connected to the servos. Moreover, to power the system including servos, Arduino board, and Xbee wireless communication, a rechargeable Lithium-ion polymer (lipo) battery is used ($Fig.2.9$). It is also worthwhile to mention that, due to hardware and wireless communication limitations, the maximum frequency that the total system can handle is $50Hz$. The experimental results indicates that using this sampling rate of data is quite satisfying to implement the proposed controllers on WMR set up.

It is also assumed that the forward velocity $V = \frac{V_1+V_2}{2} = constant$, where V_1 and V_2 are the velocity of the left wheel and the right wheel, respectively. Moreover, due to saturation (as seen in $Table2.1$), the control input range is 600 (PWM) changing between ± 300 (PWM), where $+300$ (PWM) and -300 (PWM) indicate the maximum and the

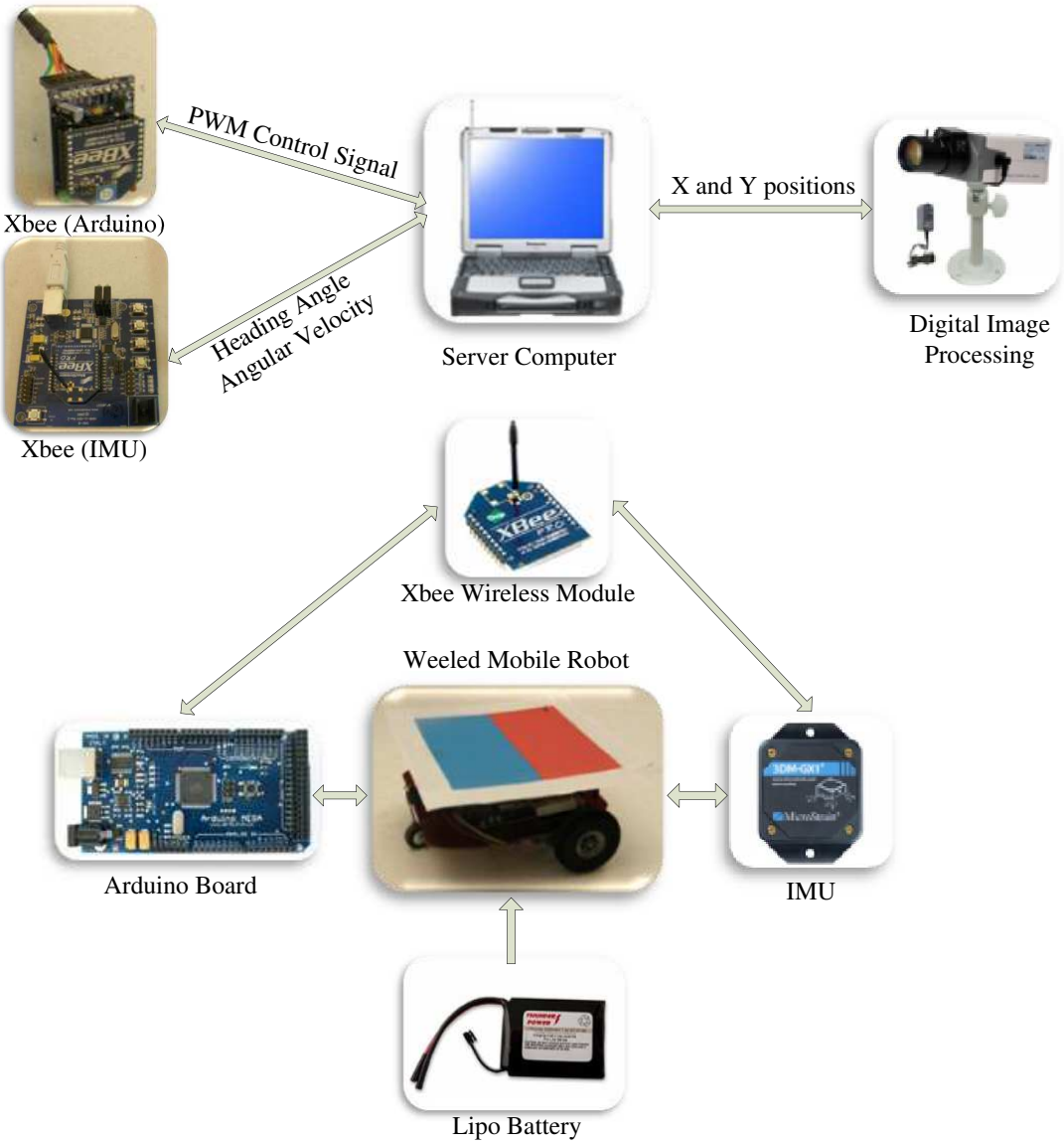

Figure 2.6: Experimental structure of Wheeled Mobile Robot

minimum possible turning speed, respectively. Therefore, if there is no control input ($u = 0$), the WMR follows the line $y = 0$. Moreover, *Table*2.1 and *Fig*.2.10 show the system identification for the WMR that we use to implement the optimal control problem defined in section 4. The forward velocity of our WMR is also 0.083 (m/s).

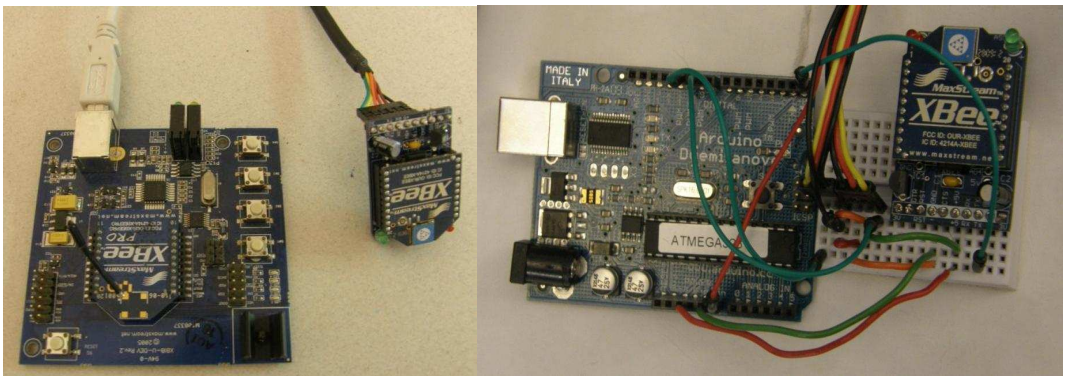

Figure 2.7: Two Xbee wireless communication modules connected to the server computer (left) and Arduino Atmega328 board connected to Xbee modules (right)

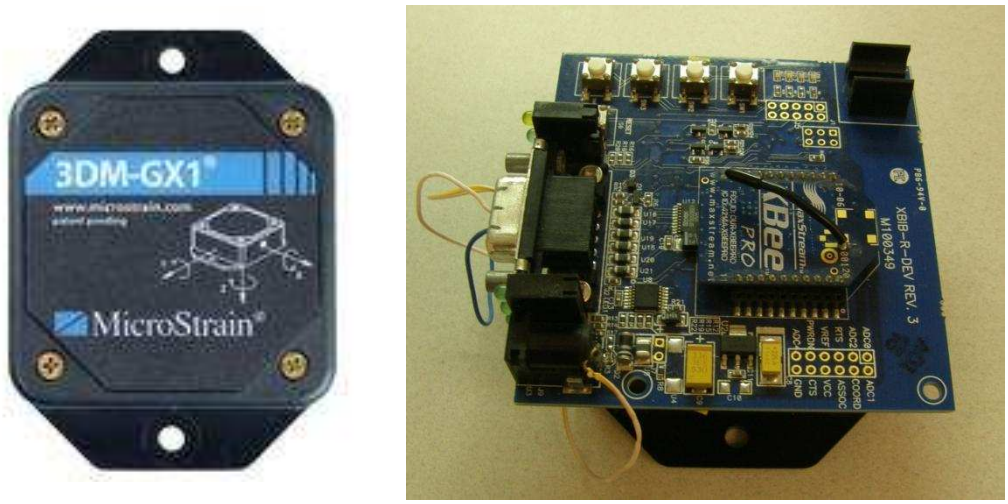

Figure 2.8: Inertial Measurement Unit (IMU) 3DM-GX1 (left) and Xbee wireless communication modules installed on IMU (right)

Table 2.1: WMR Identification Table

Control Input u (PWM)	Period T (sec)	Angular Velocity ψ ($\frac{2\pi}{T}$)
50	12.5	0.5024
100	7	0.8971
150	5.5	1.1418
200	4.5	1.3959
250	4	1.57
300	3.5	1.7943

Figure 2.9: Security camera (left) and rechargeable lipo battery (right)

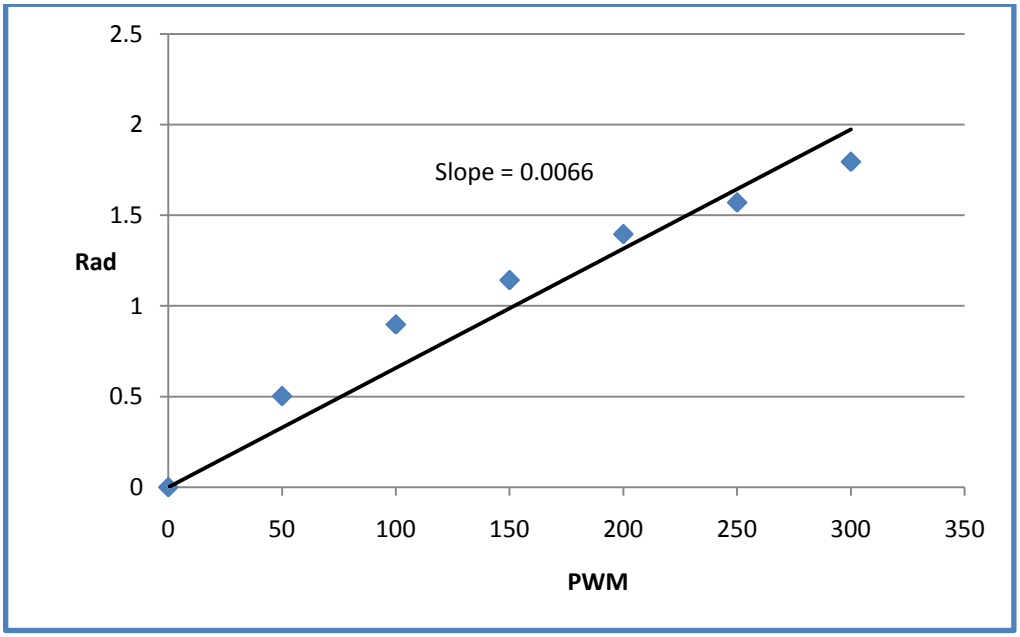

Figure 2.10: WMR Identification

Chapter 3

Large Attitude Control of Rigid Bodies Using Quaternions

3.1 Introduction

The main contribution of this chapter is to develop nonsingular rigid-body attitude control laws using a convex formulation, and implement them in an experimental set up. In work, Gollu *et al*. in [54] solved the attitude control problem with singularity, where a Modified Rodrigues Parameter(MRP)-based polynomial controller has been designed. The objective of this chapter is to tackle the same attitude problem without singularity, using a quaternion-based attitude model. To solve the attitude recovery problem without singularity, this chapter proposes not only Sum of Squares (SoS) Lyapunov based control law (the same method proposed by Gollu *et al*. in [54]) but also an SOS density function based controller. First and foremost, the key idea that enables us to use SoS technique in solving the attitude problem is that the rigid body model using either quaternions or MRP can be represented by polynomial vector fields, which was first used in [54]. Thus, the rigid-body attitude model is first parameterized in terms of quaternions, and then quaternion-based

polynomial controllers using an SOS Lyapunov function and density function are proposed to make the closed-loop system asymptotically stable. A polynomial controller based on backstepping is also developed. All these methods are then compared in a numerical simulation for a satellite with given specific parameters. Moreover, a practical application implemented in a Quanser helicopter is presented to verify effectiveness of the proposed methodology. The proposed SoS Lyapunov based control using both quaternion and MRP representations is applied to a Quanser Helicopter, and then is compared with a PID controller.

3.2 Background

Two of the most recent approaches in nonlinear control are Sum of Squares (SoS) Lyapunov based controllers [56] and SOS density function based controllers [57]. For a general nonlinear system $\dot{x} = f(x) + g(x)u$, where $f(x)$ and $g(x)$ are polynomials, searching for a control Lyapunov function and a controller simultaneously is not a convex problem. However, using a so-called density function $\rho(x)$ [57] leads to a convex formulation. Moreover, for a nonlinear system in the form of $\dot{x} = f(x)x + g(x)u$, an SoS Lyapunov function [56] can also be used to find a polynomial controller satisfying conditions of Lyapunov's second stability theorem [20]. These control methods use the Sum of Squares (SoS) decomposition technique to find the required control input for some specific nonlinear systems. First and foremost, the key idea that enables us to use SoS in solving the attitude recovery problem is that the rigid body model using either quaternions or Modified Rodrigues Parameters (MRP) can be represented by polynomial vector fields, which was first used in [54]. Using an SoS Lyapunov function, Gollu *et al*. in [54] solved the attitude control problem with singularity, where a MRP-based polynomial controller has

been designed. The objective of this chapter is to tackle the same attitude problem without singularity, using a quaternion-based attitude model. To solve the attitude recovery problem without singularity, this chapter proposes not only the same method of [54] but also an SOS density function-based controller first proposed by Rantzer *et al*. in [57]. A quaternion-based controller using backstepping for MIMO nonlinear systems is also designed. The proposed methods using both MRP and quaternion representation are then compared in a numerical simulation implemented in a Quanser helicopter. To the best of the author's knowledge, this is the first time that an SoS-based polynomial controller has been implemented. Here we are interested in quaternions rather than MRPs because i) the latter has a geometric singularity while the former one has a nonsingular representation; ii) the polynomial matrix entries using quaternions are linear while they are nonlinear for the MRP representation.

The remainder of this chapter is organized as follows. Section 3.3 presents a discussion of why we are interested in quaternions rather than in a MRP representation, and then the state space model of a rigid body in terms of quaternions is given in the general nonlinear form $\dot{x} = f(x) + g(x)u$. The control objective and a brief review of the Sum of Squares (SoS) Decomposition Method are also given in Section 3.3. In section 3.4, quaternion-based polynomial controllers using an SoS Lyapunov function, an SoS density function, and backstepping are developed to asymptotically stabilize the closed-loop system. A numerical simulation of a satellite as well as an implementation in the Quanser Helicopter will also be presented in section 3.5. Finally, a summary is given.

3.3 Preliminaries on Attitude Control Problem

This section first presents a discussion of why we are interested in quaternions rather than MRP representation, and then the quaternion-based attitude control problem is stated. A

brief introduction to the Sum of Squares (SoS) Decomposition Method is also given in this section.

3.3.1 Why Quaternions?

There are several common ways to represent the attitude of a rigid body such as direction cosine matrix, Euler axis and angle, MRP, Euler angles, and quaternions. However, the attitude determination of rigid bodies using the quaternion parameterization has several advantages over the use of other representations. First and foremost, the key idea that enables us to use SoS in solving the attitude problem is that the rigid body model using either quaternions or MRP can be represented by polynomial vector fields. Thus, since in this chapter an SoS approach is explored, only quaternions and MRP representations are considered. We are interested in quaternions rather than MRP because of the following advantages:

1. While quaternions avoid singularity, any three-parameter attitude representation like MRP has always a singularity [64], which implies that they should be avoided in situations where large-angle recovery maneuvers are present. From a practical point of view, singularity avoidance during rigid body missions, specifically for an attitude maneuver of a satellite, is critical, and thus quaternions are widely used to determine the attitude [65].

2. Quaternions have a better numerical properties [84]. The polynomial matrix entries using quaternions are of first order (linear) while they are nonlinear in a quadratic form for the MRP representation, meaning that using quaternions requires fewer computational operations when implemented in a microprocessor.

3. It will be shown that using SoS polynomial controllers based on the quaternion parameterization stabilizes the closed-loop system in less settling time with smaller overshoot rather than using MRP representation-based controllers, both numerically and experimentally (refer to 3.5). Note that these results are based on the particular simulations of this chapter with respect to different initial conditions.

3.3.2 Attitude Control Problem Definition

As discussed in section 3.3.1, the quaternion representation of rigid bodies has several advantages over the use of other polynomial representations. Therefore, a quaternion-based attitude problem is presented here. The kinematics equation of the attitude recovery problem in terms of quaternions has been obtained in equation (2.11). Assuming X, Y, and Z are the principal axes of inertia, the attitude dynamics derived by Euler's moment equations [65] can be expressed in the form of $\dot{w} = f_a + g_a u$ as

$$\begin{bmatrix} \dot{w}_x \\ \dot{w}_y \\ \dot{w}_z \end{bmatrix} = \begin{bmatrix} (\frac{I_y - I_z}{I_x}) w_y w_z \\ (\frac{I_z - I_x}{I_y}) w_z w_x \\ (\frac{I_x - I_y}{I_z}) w_x w_y \end{bmatrix} + \begin{bmatrix} \frac{1}{I_x} & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}, \tag{3.1}$$

where $u = (M_x, M_y, M_z)^T$ is the vector of the control torques acting on the rigid body, and the principal moments of inertia I_x, I_y, and I_z are the components of the inertia tensor $I = diag(I_x, I_y, I_z)$. Combining dynamics and kinematics equations (2.11) and (3.1), the state space model of the rigid body is now represented in the general nonlinear form of

$\dot{x} = f(x) + gu$ as

$$\begin{bmatrix} \dot{w}_x \\ \dot{w}_y \\ \dot{w}_z \\ \dot{q}_o \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} + \begin{bmatrix} \frac{1}{I_x} & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & \frac{1}{I_z} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}, \tag{3.2}$$

where

$$\begin{aligned} f_1 &= \left[\frac{I_y - I_z}{I_x}\right] w_y w_z \\ f_2 &= \left[\frac{I_z - I_x}{I_y}\right] w_z w_x \\ f_3 &= \left[\frac{I_x - I_y}{I_z}\right] w_x w_y \\ f_4 &= \frac{1}{2}\Big[- q_1 w_x - q_2 w_y - q_3 w_z\Big] \\ f_5 &= \frac{1}{2}\Big[q_0 w_x - q_3 w_y + q_2 w_z\Big] \\ f_6 &= \frac{1}{2}\Big[q_3 w_x + q_o w_y - q_1 w_z\Big] \\ f_7 &= \frac{1}{2}\Big[- q_2 w_x + q_1 w_y + q_o w_z\Big] \end{aligned}$$

and the state vector $x = (w_x, w_y, w_z, q_o, q_1, q_2, q_3)^T$ contains the angular velocities, and the quaternions. The control input vector $u = (M_x, M_y, M_z)^T$ contains the required moments generated about axes *x*, *y*, and *z*, respectively. Moreover, we assume that the desired set-point for the system (3.2) is where the angular velocities $w_d = (w_x, w_y, w_z)^T$ and Euler angles $\theta_d = (Roll, Pitch, Yaw)_T$ are zero. Using the expression 2.13, the point $\theta_d = (0,0,0)^T$ equivalently transforms to $q_d = (1,0,0,0)^T$. Therefore, through this chapter the desired set-point is assumed to be $x_d = (0,0,0,1,0,0,0)^T$. The attitude control problem to be solved is now stated as follows.

Problem Statement 3.3.1. *Given the attitude dynamics of the rigid body (3.2), design a nonlinear attitude controller to asymptotically stabilize the closed-loop system around the desired set-point with respect to any initial condition.*

The next section will give a brief review of the sum of squares decomposition method.

3.3.3 Sum of Squares (SoS) Decomposition Method

As proposed by Parrilo [49], in the case of polynomial functions a tractable sufficient condition of positive definiteness is the existence of a sum of squares decomposition. In fact, the condition that $P(x)$ is a Sum of Squares is computationally tractable while non-negativity is not. A polynomial $P(x), x \in \mathscr{R}^N$, is a sum of squares if there exists polynomials $f_1(x), ..., f_m(x)$ such that [48]

$$P(x) = \sum_{i=1}^{m} f_i^2(x). \tag{3.3}$$

Moreover, being SoS is equivalent to the existence of a positive semidefinite matrix Q, and a properly chosen vector of monomials $Z(x)$ such that [48]

$$P(x) = Z^T(x)\, Q\, Z(x). \tag{3.4}$$

Note also that $P(x)$ being SoS implies that $P(x) \geq 0$, but the converse is not generally correct, i.e. if a polynomial function $P(x)$ is not SoS, it does not necessarily imply the negative definiteness of $P(x)$. Using SoS decomposition method, for a given polynomials $P(x)$ and $\varphi(x)$, where $\varphi(x)$ is positive definite, the following expression

$$P(x) - \varphi(x) \quad \text{is a SoS} \tag{3.5}$$

guarantees the positive definiteness of $P(x)$. The proof is straightforward as follows. The expression $P(x) - \varphi(x)$ being SoS implies that $P(x) \geq \varphi(x)$. Therefore, since $\varphi(x) > 0$,

the polynomial $P(x)$ is positive definite. Using SoS decomposition method, a extension of Lyapunov's stability theorem for handling systems with equalities [48], which follows from the application of the Positivstellensatz Theorem, is now presented. Consider a general nonlinear system

$$\dot{x} = f(x,u) \tag{3.6}$$

with the following equalities

$$\Omega_i = 0, \quad \text{for} \quad i = 1, \cdots, N \tag{3.7}$$

where $x \in \mathscr{R}^n$ and $u \in \mathscr{R}^m$ are the states and the control inputs of the system, respectively. It is also assumed $f(0,0) = 0$.The following theorem, which is the simplified version of a theorem from [48], can be used to prove that the above system is asymptotically stable.

Proposition 3.3.1. *[48] Suppose that for the above system there exist polynomial functions $V(x)$, $a(x)$, and a positive definite function $\varphi(x)$ such that*

$$V(x) - \varphi(x) \quad \textit{is SoS} \tag{3.8}$$

$$-\frac{\partial V}{\partial x} f(x,u) + \sum_{i=0}^{n} a_i \Omega_i - \varphi(x) \quad \textit{is SoS.} \tag{3.9}$$

Then, the origin is asymptotically stable.

Proof. See [48]. □

To convert the SoS decomposition problem to the corresponding Semidefinite Program (SDP) formulation, a freely-available MATLAB toolbox, the software SOSTOOLS [50] has been developed. The above definitions are now used to solve the attitude control problem.

3.4 Attitude Control Problem Solution

The objective of this section is to propose three different methods to solve the attitude control problem 3.3.1, and to compare them. These methods include SOS Lyapunov based control, SOS density function based control, and backstepping. Thus, in this section quaternion-based polynomial controllers using an SOS Lyapunov function and an SOS density function are proposed to make the closed-loop system asymptotically stable. A backstepping method for the MIMO nonlinear system is also developed. All these methods are then compared in simulations and experiments.

3.4.1 SoS Lyapunov Based Control

The rigid body model (3.2) is parameterized as the following state dependent linear-like form

$$\dot{x} = A(x)x + g\,u \tag{3.10}$$

with

$$A(x) = \begin{bmatrix} 0 & a_{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{23} & 0 & 0 & 0 & 0 \\ a_{31} & 0 & 0 & 0 & 0 & 0 & 0 \\ -q_1/2 & -q_2/2 & -q_3/2 & 0 & 0 & 0 & 0 \\ q_o/2 & -q_3/2 & q_2/2 & 0 & 0 & 0 & 0 \\ q_3/2 & q_o/2 & -q_1/2 & 0 & 0 & 0 & 0 \\ -q_2/2 & q_1/2 & q_o/2 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.11}$$

where

$$a_{12} = \left[\frac{I_y - I_z}{I_x}\right] W_z \tag{3.12}$$

$$a_{23} = \left[\frac{I_z - I_x}{I_y}\right] W_x \tag{3.13}$$

$$a_{31} = \left[\frac{I_x - I_y}{I_z}\right] W_y \,. \tag{3.14}$$

To design a controller using an SoS Lyapunov function, the polynomial vector fields should have an equilibrium point at the origin [52], i.e. all state variables should converge to zero. Each state variable should, therefore, be shifted from its trim condition (or desired set-point) to the origin. Let us denote this transformation by

$$\hat{x} = x - x^*, \tag{3.15}$$

where x and x^* denote the original state and the trim point of the original state, respectively. Note that the derivatives of the new shifted variables are the same as the original ones. Therefore, using the shifted state vector $\hat{x} = (\hat{w}_x, \hat{w}_y, \hat{w}_z, \hat{q}_o, \hat{q}_1, \hat{q}_2, \hat{q}_3)^T$, the dynamic model becomes

$$\dot{\hat{x}} = f(\hat{x}) + g\,u = A(\hat{x})\,\hat{x} + g\,u, \tag{3.16}$$

where the polynomial matrix $A(\hat{x})$ is given by

$$A(\hat{x}) = \begin{bmatrix} 0 & a_{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{23} & 0 & 0 & 0 & 0 \\ a_{31} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\hat{q}_1/2 & -\hat{q}_2/2 & -\hat{q}_3/2 & 0 & 0 & 0 & 0 \\ (\hat{q}_o+1)/2 & -\hat{q}_3/2 & \hat{q}_2/2 & 0 & 0 & 0 & 0 \\ \hat{q}_3/2 & (\hat{q}_o+1)/2 & -\hat{q}_1/2 & 0 & 0 & 0 & 0 \\ -\hat{q}_2/2 & \hat{q}_1/2 & (\hat{q}_o+1)/2 & 0 & 0 & 0 & 0 \end{bmatrix},$$

with the quaternion constraint

$$\Omega = \hat{q}_o^2 + \hat{q}_1^2 + \hat{q}_2^2 + \hat{q}_3^2 = 1\,. \tag{3.17}$$

Note that, since the desired set-point is $x_d = (0,0,0,1,0,0,0)$, the only state which needs to be shifted is related to the scalar components of the quaternions q_o. Therefore, expressions a_{12}, a_{32}, and a_{31} are the same as in (3.12), (3.13), and (3.14), respectively. For the rigid body model (3.16), a Lyapunov-based controller will now be designed to asymptotically stabilize the closed-loop system. The objective is now to find a quaternion-based state feedback controller for the rigid body nonlinear model (3.16) which guarantees asymptotic stability. For this, the following theorem, which is the simplified version of a theorem from [56], is stated.

Theorem 3.4.1. *[56] Given $A(\hat{x})$ and g for the system (3.16) with the quaternion constraint Ω (3.17) if one can find a symmetric matrix P and a polynomial matrix $K(\hat{x})$ such that $\varepsilon_2(\hat{x})$ is a sum of squares and*

$$F^T(P-\varepsilon_1 I)F \quad \textit{is SoS} \tag{3.18}$$

$$-F^T(PA^T(\hat{x})+A(\hat{x})P+(gK(\hat{x}))^T+gK(\hat{x})+\varepsilon_2(\hat{x})I)F+a(x)\Omega-\varphi(x) \quad \textit{is SoS} \tag{3.19}$$

where ε_1 and $a(x)$ are a constant and a polynomial multiplier, respectively, then the state feedback controller which stabilizes the closed-loop system is given by

$$u(\hat{x}) = K(\hat{x})P^{-1}\hat{x}. \tag{3.20}$$

Proof. It follows from the proof of [56] with $P(\hat{x}) = P$, $Z(\hat{x}) = \hat{x}$, and $M = I$. Moreover, since the quaternion-based model needs to satisfy the constraint Ω (3.17), using the proposition 3.3.1 a polynomial expression $a(x)\Omega$ is also added to the SoS relaxation, where $a(x)$ is a polynomial multiplier. □

Therefore, given the nonlinear system (3.16) and solving the SoS problem in Theorem 3.4.1, one can find the control input (3.20), which makes the closed-loop system asymptotically stable.

3.4.2 SoS Density Function-Based Control

It is well-known that for a general nonlinear system, the joint search for a controller and a Lyapunov function is not convex. For the case of nonlinear systems with polynomial or rational vector fields, a so-called density function $\rho(x)$, which is also interpreted as a dual to the Lyapunov function, has first been proposed in [57], and has been extended in [52]-[53]. The main result of [52], which formulates the joint search as a convex problem with constraints, is stated as follows.

Theorem 3.4.2. *[52] Given the system $\dot{x} = f(x) + g(x)u$ with a constraint $\Omega(x)$, where $(f+gu)(x) \in C^1(\mathscr{R}^n, \mathscr{R}^n)$, $(f+gu)(0) = 0$, and $a(x)$ is a polynomial multiplier, suppose there exists a non-negative function $\rho(x) \in C^1(\mathscr{R}^n - \{0\}, \mathscr{R}^n)$, referred to as the density function, such that $\rho(x)(f+gu)(x)/|x|$ is integrable on $\{x \in \mathscr{R}^n : |x| \geq 1\}$, and for almost all x*

$$\nabla.\big[\rho(f+gu)\big](x) + a(x)\Omega(x) > 0\,. \tag{3.21}$$

Then, for almost all initial states $x(0)$, the trajectory x(t) exists for $t \in [0, \infty)$ and tends to zero as $t \longrightarrow \infty$. Moreover, if the closed-loop equilibrium $x = 0$ is stable, then the conclusion remains valid even if $\rho(x)$ takes negative values.

Proof. It follows from the proof of [57]. Moreover, since the system is subject to a constraint $\Omega(x)$, using the application of the Positivstellensatz Theorem a polynomial $a(x)\Omega(x)$ is also added to the resulting expression, where $a(x)$ is a polynomial multiplier. □

In order to jointly search for the density function and the controller, the following parameterization is considered [57]

$$\rho(x) = \frac{p(x)}{t(x)^s} \quad , \quad u(x) = \frac{w(x)}{p(x)} \tag{3.22}$$

where $p(x), t(x)$, and $w(x)$ are polynomials, $t(x)$ is positive, and s is chosen to satisfy the integrability condition in Theorem 3.4.2. By plugging (3.22) in (3.21) and using the gradient and the divergence properties in (2.22)-(2.23), the first component of condition (3.21) is written as [57]

$$\begin{aligned}
\nabla.\big[\rho(f+gu)\big](x) &= \nabla.\left[\frac{1}{t(x)^s}(fp+gw)(x)\right] \\
&= \nabla\left(\frac{1}{t(x)^s}\right)(fp+gw)(x) + \frac{1}{t(x)^s}\nabla.(fp+gw)(x) \\
&= -\frac{s}{t(x)^{s+1}}\nabla\big(t(x)\big)(fp+gw)(x) + \frac{1}{t(x)^s}\nabla.(fp+gw)(x) \\
&= \frac{1}{t(x)^{s+1}}\left[t(x)\nabla.(fp+gw)(x) - s\nabla\big(t(x)\big).(fp+gw)(x)\right] > 0\,.
\end{aligned} \tag{3.23}$$

Since $t(x)$ is positive, we only need to satisfy the following inequality

$$t(x)\nabla.(fp+gw)(x) - s\nabla\big(t(x)\big).(fp+gw)(x) > 0\,. \tag{3.24}$$

Assuming that $f(x)$ and $g(x)$ in the above equation are polynomials, using SoS relaxation the inequality (3.21) is satisfied if

$$t(x)\nabla.(fp+gw)(x) - s\nabla\big(t(x)\big).(fp+gw)(x) + a(x)\Omega(x) - \varphi(x) \quad \text{is SoS} \tag{3.25}$$

where $\varphi(x)$ is a positive definite polynomial function. Note also that since all state variables should converge to zero, each state variable should be shifted by the transformation in (3.15). A good first candidate for $t(x)$ is the Control Lyapunov Function (CLF) for the linearized system

$$\dot{x} = Ax + Bu\,. \tag{3.26}$$

Given linear dynamics (A,B), to find CLF we consider the following candidate Lyapunov function

$$t(x) = V(x) = x^T Rx, \tag{3.27}$$

where x is the state vector, and R is a symmetric positive definite matrix which needs to be obtained. Obviously, since $R > 0$, V(x) is positive definite for all x. However, to guarantee that the closed-loop system is asymptotically stable, $\dot{V}(x)$ needs to be negative definite. Now assuming $Q = R^{-1}$ and the control input

$$u = GRx, \tag{3.28}$$

the derivative of $V(x)$ with respect to time along the trajectories of 3.26 is given by

$$\begin{aligned}\dot{V}(x) =& \dot{x}^T Rx + x^T R\dot{x} \\ =& x^T \left(A^T R + RG^T B^T R + RA + RBGR\right)x \\ =& x^T R\left(QA^T + G^T B^T + AQ + BG\right)Rx. \end{aligned} \tag{3.29}$$

Then, to satisfy $\dot{V}(x) < 0$, we only need to solve the following Linear Matrix Inequalities (LMIs)

$$AQ + QA^T + BG + G^T B^T < 0 \quad , \quad Q = Q^T > 0. \tag{3.30}$$

The SoS density function approach for solving the attitude control problem is now summarized in the following algorithm.

Algorithm 1. *Using SoS density function approach, the following steps are proposed to obtain a polynomial control input for the attitude control problem (3.16):*

1. *Given a linearized model (A,B), matrices Q and G are obtained by solving the LMIs (3.30)*
2. *A positive definite function $t(x)$ will then be given by $t(x) = V(x) = x^T Q^{-1} x$*
3. *Search for polynomials $p(x)$ and $w(x)$, as defined in (3.22), to satisfy the SoS problem (3.25)*

4. *The control input $u(x) = \frac{w(x)}{p(x)}$ is then obtained, which makes the closed loop system (3.16) asymptotically stable.*

3.4.3 Backstepping Approach for MIMO Nonlinear System

Consider the general backstepping system

$$\dot{z} = \alpha(z) + \beta(z)\zeta \tag{3.31}$$

$$\dot{\zeta} = f_a(z,\zeta) + g_a(z,\zeta)U \tag{3.32}$$

where $\left[z^T, \zeta^T\right]^T \in \mathscr{R}^{n+m}$ is the state vector, and $U \in \mathscr{R}^m$ is the control input vector. The functions $\alpha \in \mathscr{R}^n$, $\beta \in \mathscr{R}^{n\times m}$, $f_a \in \mathscr{R}^m$, and $g_a \in \mathscr{R}^{m\times m}$ are smooth, and α and f_a vanish at the origin. The attitude control problem (3.16) can be written in a cascade connection of two subsystems, as shown in (3.31)-(3.32). Therefore, the objective of this subsection is to stabilize the system (3.31)-(3.32) using backstepping approach. Using the control input

$$U = g_a(z,\zeta)^{-1}\Big[u - f_a(z,\zeta)\Big], \tag{3.33}$$

where g_a is a nonsingular diagonal matrix, the system (3.31)-(3.32) can be reduced to the following system

$$\dot{z} = \alpha(z) + \beta(z)\zeta \tag{3.34}$$

$$\dot{\zeta} = u \tag{3.35}$$

where $u \in \mathscr{R}^m$ is the control input vector, which needs to be obtained. When the state ζ is scalar (and consequently the input u is also scalar), the system (3.34)-(3.35) is reduced to the integrator backstepping system as shown in [20]. Here we consider a MIMO nonlinear system. To stabilize the system (3.34) and (3.35) at the origin, the following backstepping approach is given.

Theorem 3.4.3. *Given the system (3.34) and (3.35), suppose there is a stabilizing state feedback law $\zeta = \phi(z) \in \mathscr{R}^m$ for the subsystem (3.34) such that $\phi(0) = 0$. Let $V(z)$ be a Lyapunov function for the subsystem (3.34) such that $V(z)$ is positive definite and*

$$\frac{\partial V(z)}{\partial z}\Big[\alpha(z) + \beta(z)\phi(z)\Big] \leq -W(z) \tag{3.36}$$

where $W(z)$ is positive definite, and $\frac{\partial V}{\partial z} = \nabla V = \Big[\frac{\partial V}{\partial z_1} \ \cdots \ \frac{\partial V}{\partial z_n}\Big]$. Then, the feedback law

$$u = \frac{\partial \phi}{\partial z}\Big[\alpha(z) + \beta(z)\phi(z)\Big] - \Big(\frac{\partial v}{\partial z}\beta(z)\Big)^T - ky \tag{3.37}$$

where $k > 0$ and

$$y = \zeta - \phi(z) = \big[\zeta_1 - \phi_1(z) \ \cdots \ \zeta_m - \phi_m(z)\big]^T, \tag{3.38}$$

stabilizes the origin $(z^T = 0 \ , \ \zeta^T = 0)$, and a Lyapunov function for the closed-loop system is

$$V_c(z, \zeta) = V(z) + \frac{1}{2}y^T y\,. \tag{3.39}$$

Proof. Suppose the subsystem (3.34) can be stabilized asymptotically by a state feedback control $\zeta = \phi(z)$ with $\phi(0) = 0$. Suppose, moreover, that there is a Lyapunov function $V(z)$ for the subsystem (3.34) such that $V(z)$ is positive definite and satisfies (3.36). By adding and subtracting $\beta(z)\phi(z)$, and performing the change of variable $y = \zeta - \phi(z)$, we obtain the system

$$\dot{z} = \Big[\alpha(z) + \beta(z)\phi(z)\Big] + \beta(z)y \tag{3.40}$$

$$\dot{y} = u - \dot{\phi}(z) \tag{3.41}$$

where

$$\dot{\phi}(z) = \frac{\partial \phi}{\partial z}\big[\alpha(z) + \beta(z)\zeta\big] \tag{3.42}$$

and $\frac{\partial \phi}{\partial z} \in \mathscr{R}^{m\times n}$ is the jacobian matrix as defined in (2.25). Letting $v = u - \dot{\phi}$ reduces the

system (3.40) and (3.41 to

$$\dot{z} = \Big[\alpha(z) + \beta(z)\phi(z)\Big] + \beta(z)y \tag{3.43}$$

$$\dot{y} = v \tag{3.44}$$

which has the same form as the system we started from with the exception that we now know the subsystem (3.43) is asymptotically stable to the origin when $y = 0$. Now, for the system (3.43) and (3.44), let us consider the candidate Lyapunov function (3.39) which is positive definite. Then, the derivative of V_c is

$$\begin{aligned}\dot{V}_c &= \frac{\partial V(z)}{\partial z}\Big[\alpha(z) + \beta(z)\phi(z)\Big] + \frac{\partial V(z)}{\partial z}\beta(z)y + v^T y \\ &\leq -W(z) + \frac{\partial V(z)}{\partial z}\beta(z)y + v^T y\,.\end{aligned} \tag{3.45}$$

Choosing

$$v = -\Big(\frac{\partial V}{\partial z}\beta(z)\Big)^T - ky \tag{3.46}$$

where $k > 0$, implies that

$$\dot{V}_c \leq -W(z) - ky^T y. \tag{3.47}$$

This also shows that the origin $(z^T = 0\ ,\ \zeta^T = 0)$ is asymptotically stable. Finally, combining (3.46) and $u = v + \dot{\phi}$ results in the control input (3.37). This finishes the proof. □

The attitude control problem (3.16) can now be written in the general backstepping format (3.31)-(3.32). The functions $\alpha(z)$ and $\beta(z)$ are expressed as

$$\alpha(z) = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}^T \tag{3.48}$$

and

$$\beta(z) = \frac{1}{2}\begin{bmatrix} -\hat{q}_1 & -\hat{q}_2 & -\hat{q}_3 \\ (\hat{q}_o + 1) & -\hat{q}_3 & \hat{q}_2 \\ \hat{q}_3 & (\hat{q}_o + 1) & -\hat{q}_1 \\ -\hat{q}_2 & \hat{q}_1 & (\hat{q}_o + 1)\end{bmatrix} \tag{3.49}$$

where z and ζ are the vectors of shifted quaternions and angular velocities, respectively. Moreover, $f_a(\hat{q}, w)$ and $g_a(\hat{q}, w)$ are the same as in expressions (3.1).

In summary, to solve the attitude recovery problem (3.16) using backstepping, a Lyapunov function $V(x)$, where $V > 0$ and its derivative is negative definite, should first be found for the subsystem (3.31). Once one can find this Lyapunov function, the control input (3.33) can be obtained, which makes the overall system (3.31)-(3.32) asymptotically stable.

3.5 Simulations and Experiments

This section presents two examples of attitude control problems, including a numerical simulation and an experimental result on the pitch control of a Quanser helicopter [68]. The numerical simulation of a rigid satellite with specific inertial elements is covered in the first subsection, and then the theoretical results of the thesis are applied to a Quanser Helicopter in the second subsection. In the numerical simulations, given a satellite with specific parameters, the proposed polynomial control laws in this chapter are designed and compared. Moreover, the SoS Lyapunov function-based controllers using quaternion and MRP representations are implemented in the Quanser helicopter, and then are compared with a PID controller.

3.5.1 Numerical Simulation

The proposed controllers in Sections 3.4.1, 3.4.2 and 3.4.3 are now applied to a rigid satellite with the inertia tensor $I = diag(1, 1.2, 0.8)(kg.m^2)$, considered as a small satellite. The objective is to make the satellite asymptotically stable subject to a nonzero initial attitude. Since there is no singularity using quaternions, this controller design is applicable to any arbitrary initial states. Let us assume the initial angular velocities and the initial

Euler angles are as follows

$$w_o = (0,0,0)^T \quad , \quad \theta_0 = (Roll, Pitch, Yaw)^T = (80^\circ, 50^\circ, -120^\circ)^T .$$

The desired Euler angles are $\theta_d = (0,0,0)^T$. Using the expression 2.13, the initial and desired orientations of the satellite in terms of quaternions correspond to

$$\theta_o = (0.1119, 0.5717, -0.3426, -0.7371)^T \quad , \quad \theta_d = (1,0,0,0)^T .$$

Now, given the attitude recovery dynamic model (3.16) with the quaternion constraint Ω (3.17) and the numerical values of the parameters, the quaternion-based polynomial controllers using different methods are obtained as follows.

SoS Lyapunov Based Control

Substituting the given inertia tensor in (3.16), the matrices $A(\hat{x})$ and g become

$$A(\hat{x}) = \begin{bmatrix} 0 & 0.4w_z & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.16w_x & 0 & 0 & 0 & 0 \\ -0.25w_y & 0 & 0 & 0 & 0 & 0 & 0 \\ -\hat{q}_1/2 & -\hat{q}_2/2 & -\hat{q}_3/2 & 0 & 0 & 0 & 0 \\ (\hat{q}_o+1)/2 & -\hat{q}_3/2 & \hat{q}_2/2 & 0 & 0 & 0 & 0 \\ \hat{q}_3/2 & (\hat{q}_o+1)/2 & -\hat{q}_1/2 & 0 & 0 & 0 & 0 \\ -\hat{q}_2/2 & \hat{q}_1/2 & (\hat{q}_o+1)/2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$g = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.83 & 0 \\ 0 & 0 & 1.25 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Given these matrices and the state feedback controller structure (3.20), the SOSTOOLS Toolbox [50] will now be used to solve the SoS Lyapunov-based control in theorem 3.4.1. Using the MATLAB code in Appendix 3.7), the diagonal matrix

$$P = diag(1.05, 0.99, 1.11, 1.19, 1.23, 1.22, 1.23),$$

where $P > 0$, and the following polynomial control input vector $M = (M_x, M_y, M_z)^T$ are obtained

$$\begin{aligned} M(x) = & -0.62x_1^3 - 0.63x_1x_2^2 - 0.04x_1x_2x_6 - 0.61x_1x_3^2 \\ & - 0.08x_1x_3x_7 - 0.51x_1(x_4-1)^2 + 0.31x_1(x_4-1) \\ & - 0.5x_1x_6^2 - 0.53x_1x_7^2 - 0.37x_1 - 0.05x_2x_3(x_4-1) \\ & - 0.19x_2x_3 - 0.02x_2(x_4-1)x_7 - 0.11x_2x_5x_6 \\ & - 0.02x_2x_7 + 0.03x_3^2x_5 + 0.05x_3x_5x_7 - 0.03x_3x_6 \\ & + 0.01(x_4-1)x_5 - 0.43x_5 - 0.59x_1x_5^2 \end{aligned}$$

$$
\begin{aligned}
M(y) = &-0.76x_1^2x_2 - 0.02x_1^2x_6 + 0.06x_1x_2x_5 - 0.04x_1x_3(x_4-1) \\
&- 0.16x_1x_3 - 0.01x_1(x_4-1)x_7 - 0.1x_1x_5x_6 - 0.01x_1x_7 \\
&- 0.77x_2^3 - 0.8x_2x_3^2 - 0.62x_2(x_4-1)^2 + 0.41x_2(x_4-1) \\
&- 0.72x_2x_6^2 - 0.61x_2x_7^2 - 0.43x_2 + 0.01x_3(x_4-1)x_5 \\
&+ 0.03x_3x_5 - 0.04x_3x_6x_7 + 0.01(x_4-1)x_6 - 0.5x_6 \\
&- 0.59x_2x_5^2
\end{aligned}
$$

$$
\begin{aligned}
M(z) = &-0.5x_1^2x_3 + 0.05x_1x_2(x_4-1) + 0.27x_1x_2 + 0.03x_1x_3x_5 \\
&- 0.05x_1x_5x_7 - 0.02x_1x_6 - 0.49x_2^2x_3 - 0.03x_2^2x_7 \\
&+ 0.02x_2x_3x_6 + 0.02x_2(x_4-1)x_5 + 0.04x_2x_5 \\
&- 0.43x_3(x_4-1)^2 + 0.18x_3(x_4-1) - 0.44x_3x_5^2 \\
&- 0.48x_3x_7^2 - 0.33x_3 - 0.36x_7 - 0.44x_3x_6^2 \\
&- 0.49x_3^3 + 0.01(x_4-1)x_7\,.
\end{aligned}
$$

SoS Density Function-Based Control

Since (f,g) in (3.2) are polynomial vector fields, a polynomial controller based on the SoS density function for the same satellite can also be designed. As discussed in Section 3.4.2, a positive polynomial function $t(\hat{x})$ should first be found. A good candidate for $t(\hat{x})$ is a Control Lyapunov function for the linearized system of (3.16) around the desired set-point. Linearizing the nonlinear system (3.16) yields a linear state space model in the form $\dot{\hat{x}} = A_l\hat{x} + gu$ with

$$
A_l = \begin{bmatrix} O_{4*3} & O_{4*4} \\ 0.25I_{3*3} & O_{3*4} \end{bmatrix},
$$

where matrices O and I are the zero and identity matrices, respectively. Solving LMIs (3.30) and using YALMIP and SeDuMi, a symmetric positive definite matrix R is obtained

as follows

$$R = \begin{bmatrix} 1.02 & 0 & 0 & 0 & 0.46 & 0 & 0 \\ 0 & 1.02 & 0 & 0 & 0 & 0.46 & 0 \\ 0 & 0 & 1.02 & 0 & 0 & 0 & 0.46 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0.46 & 0 & 0 & 0 & 1.02 & 0 & 0 \\ 0 & 0.46 & 0 & 0 & 0 & 1.02 & 0 \\ 0 & 0 & 0.46 & 0 & 0 & 0 & 1.02 \end{bmatrix}.$$

The positive definite polynomial $t(\hat{x})$ will thus be $V(\hat{x}) = \hat{x}^T R \hat{x}$. The next step is to search for polynomials $p(\hat{x})$ and $w(\hat{x})$, as defined in (3.22), to satisfy (3.25). For this, $p(\hat{x})$ and $w(\hat{x})$ are assumed to be polynomials. It is assumed that $p(\hat{x})$ is a constant while the three elements of vector $w(\hat{x})$ are second-degree polynomials. Using the SOSTOOLS Toolbox [50] and assuming

$$s \geq 3 \quad , \quad p = 10^{-6} \quad , \quad \varphi = w_x^2 + w_y^2 + w_z^2 , \tag{3.50}$$

the following control inputs are obtained

$$\begin{aligned} M(x) =& 0.003q_1(q_o - 1) - 0.14q_1 - 0.002q_2q_3 + 0.004q_2w_z \\ &+ 0.006q_3w_y - 0.03(q_o - 1)w_x - 0.35w_x - 0.004w_yw_z \\ M(y) =& -0.01q_1w_z + 0.003q_2q_0 - 0.17q_2 + 0.004q_3w_x \\ &- 0.04(q_o - 1)w_y + 0.003w_xw_z - 0.42w_y \\ M(z) =& 0.002q_1q_2 - 0.01q_1w_y - 0.17q_3 - 0.03(q_o - 1)w_z \\ &+ 0.004w_xw_y - 0.42w_z . \end{aligned}$$

Backstepping

The attitude control problem (3.16) can also be solved by the backstepping method outlined in theorem 3.4.3. Rewriting attitude problem (3.16) in the general backstepping

format (3.31)- (3.32), the functions $\alpha(z)$, $\beta(z)$, and $(f_a(\hat{q},w), g_a(\hat{q},w))$ are the same as in expressions (3.48), (3.49), and (3.1), respectively. Now, given the subsystem

$$\dot{\hat{q}} = f(\hat{q}) + g(\hat{q})w = \frac{1}{2}\begin{bmatrix} -\hat{q}_1 & -\hat{q}_2 & -\hat{q}_3 \\ (\hat{q}_o+1) & -\hat{q}_3 & \hat{q}_2 \\ \hat{q}_3 & (\hat{q}_o+1) & -\hat{q}_1 \\ -\hat{q}_2 & \hat{q}_1 & (\hat{q}_o+1) \end{bmatrix} w, \tag{3.51}$$

where w is the control input, a stabilizing control law should be obtained to make this subsystem asymptotically stable. For this, the SoS problem in theorem 3.4.1 is solved to find an SoS Lyapunov based controller (3.20). Solving this SoS problem using the SOSTOOLS Toolbox [50], the controller law is obtained as

$$w = \phi(\hat{q}) = \begin{bmatrix} -\mu_1\hat{q}_1 & -\mu_2\hat{q}_2 & -\mu_3\hat{q}_3 \end{bmatrix}^T \tag{3.52}$$

where μ_1, μ_2, and μ_3 are positive constants. Replacing the control input (3.52) in the first subsystem (3.51), the closed-loop system for the first subsystem is given by

$$\dot{\hat{q}} = \begin{bmatrix} \hat{q}_2^2 + \hat{q}_3^2 + \hat{q}_4^2 \\ -\hat{q}_2(\hat{q}_o+1) \\ -\hat{q}_3(\hat{q}_o+1) \\ -\hat{q}_4(\hat{q}_o+1) \end{bmatrix}. \tag{3.53}$$

Now, using the Lyapunov function

$$V(\hat{q}) = \frac{1}{2}\left(\hat{q}_o^2 + \hat{q}_1^2 + \hat{q}_2^2 + \hat{q}_3^2\right) > 0, \tag{3.54}$$

the derivative of $V(\hat{q})$ with respect to time along the trajectories of (3.53) is calculated as

$$\begin{aligned} \dot{V}(\hat{q}) =& \frac{\partial V(\hat{q})}{\partial \hat{q}}\dot{\hat{q}} = \left(\hat{q}_o\dot{\hat{q}}_o + \hat{q}_1\dot{\hat{q}}_1 + \hat{q}_2\dot{\hat{q}}_2 + \hat{q}_3\dot{\hat{q}}_3\right) \\ =& -\frac{1}{2}\left(\mu_1\hat{q}_1^2 + \mu_2\hat{q}_2^2 + \mu_3\hat{q}_3^2\right). \end{aligned} \tag{3.55}$$

The polynomial function $\dot{V}$ appears to be negative semidefinite in $S = \{x|\hat{q}_1 = \hat{q}_2 = \hat{q}_3 = 0, \hat{q}_o \in \mathscr{R}\}$. However, due to the quaternion constraint $\Omega = (\hat{q}_o + 1)^2 + \hat{q}_1^2 + \hat{q}_2^2 + \hat{q}_3^2 = 1$, $\dot{V}$ is not defined over the line $S = \{x|\hat{q}_1 = \hat{q}_2 = \hat{q}_3 = 0, \hat{q}_o \in R\}$, except at the origin. Therefore, $\dot{V}$ becomes zero only at the point $(\hat{q}_o = \hat{q}_1 = \hat{q}_2 = \hat{q}_3 = 0)$ and is negative definite. Thus, the closed-loop system (3.53) is asymptotically stable. Using the resulting $\phi(\hat{q})$ and $V(\hat{q})$, the control law (3.37) can now be obtained to stabilize the overall system. Assuming $k = \mu_1 = \mu_2 = \mu_3 = 1$ and the given inertia tensor, the following control inputs are obtained.

$$\begin{aligned}
u_1 =& -2w_x - 2.5q_1 - (q_o - 1)w_x + q_3 w_y - q_2 w_z \\
& - 0.4 w_y w_z - 2q_o q_1 \\
u_2 =& 1.2\Big(-2w_y - 2.5q_2 - q_3 w_x - (q_o - 1)w_y \\
& + q_1 w_z + 0.16 w_z w_x - 2q_o q_2 \Big) \\
u_3 =& 0.8\Big(-2w_z - 2.5q_3 + q_2 w_x - q_1 w_y - (q_o - 1)w_z \\
& + 0.25 w_x w_y - 2q_o q_3 \Big)
\end{aligned}$$

The state trajectories for the three approaches are shown in *Fig*.3.1, *Fig*.3.2, *Fig*.3.3, and *Fig*.3.4. *Fig*.3.1 shows that all the quaternion elements converge to zero, and that the quaternion constraint is always verified. *Fig*.3.2 shows the time response of the angular velocities converging to zero. The time response of the control inputs is also shown in *Fig*.3.3 and *Fig*.3.4. It is important to notice the small magnitude of the required torques for stabilizing the satellite. From a practical point of view, it implies that the satellite will require lower power.

Fig.3.5 compares the time response of the Euler angles using MRP-based and Quaternion-based controllers. It shows that using state-feedback controllers based on the quaternion parameterization stabilizes the closed-loop system in less settling time with a

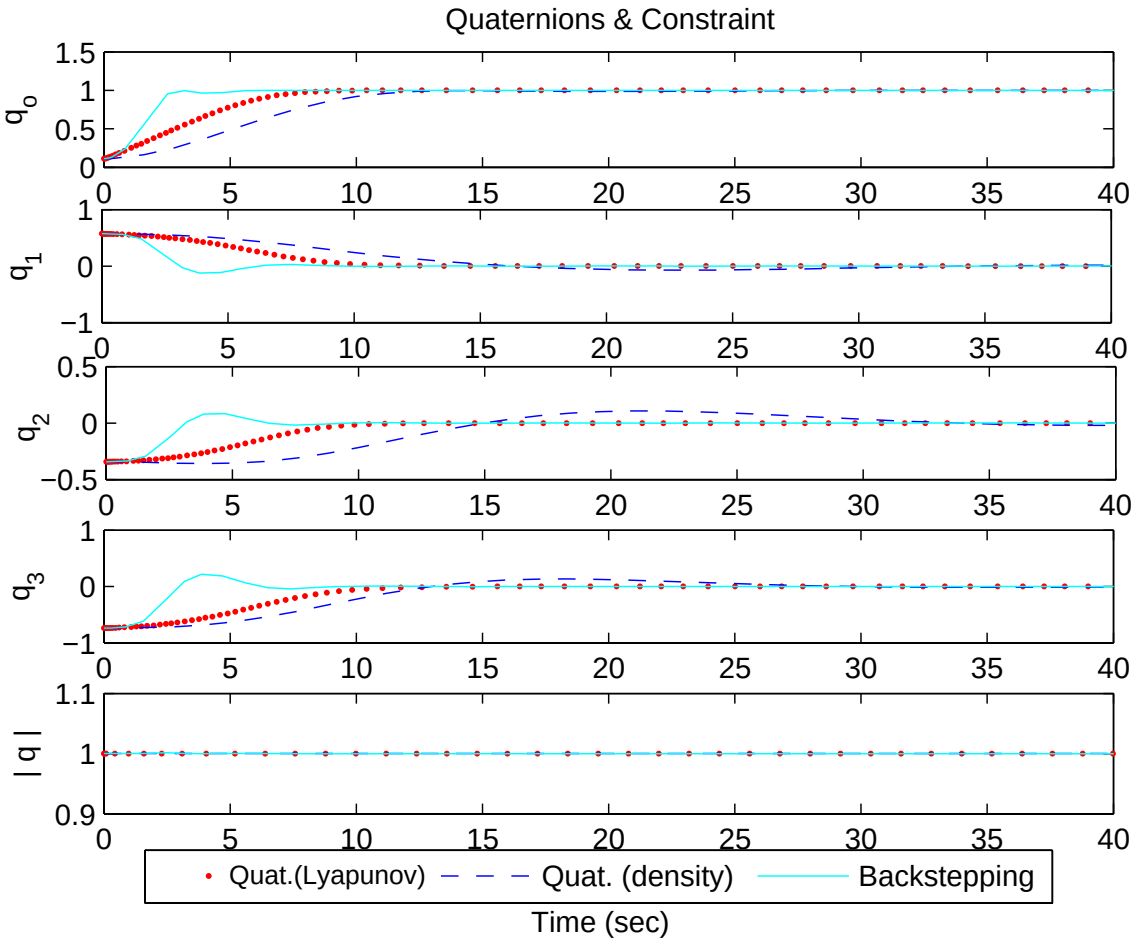

Figure 3.1: Time response of quaternions and associated constraint for SoS Lyapunov based control, SoS density function based control, and backstepping approach

smaller overshoot rather than using MRP-based controllers. Moreover, using MRP representation results in a singularity which is not desirable in practical applications, specifically satellite applications. Moreover, from *Fig*.3.5 it is observed that the quaternion-based polynomial controller using SoS Lyapunov-based approach has the best responses in terms of settling time, overshoot and smoothness. *Fig*.3.5 also shows that using the backstepping controller makes the closed-loop system asymptotically stable. As shown in *Fig*.3.3 and *Fig*.3.4, the control inputs for backstepping approach is much bigger than other two SoS-based approaches, resulting in higher overshoot responses compared with other two SoS-based approaches. It is worthwhile to note that these results are based on our particular simulations with respect to different initial conditions.

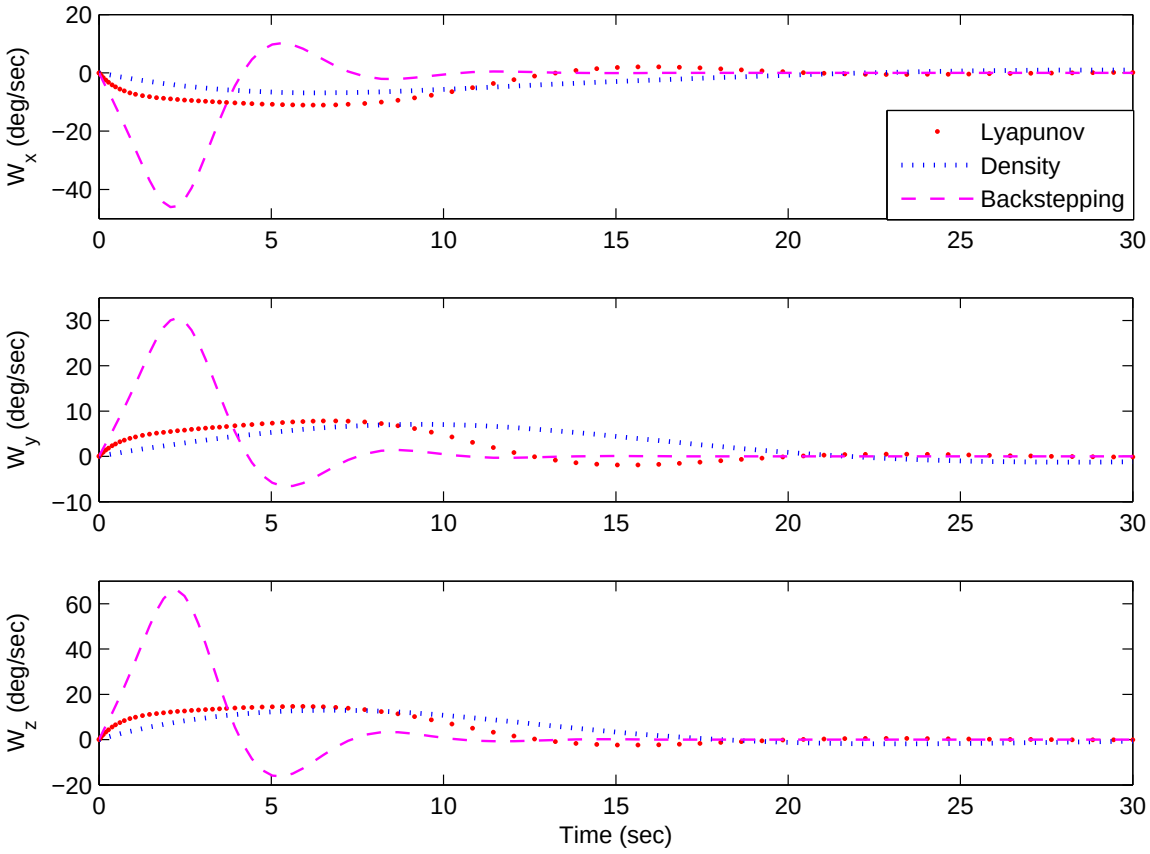

Figure 3.2: Time response of angular velocities for SoS Lyapunov based control, SoS density function based control, and backstepping approach

3.5.2 Experimental Results on Quanser Helicopter

The proposed SoS Lyapunov based control (3.20) using both quaternion and MRP representations is now applied to a Quanser Helicopter [68]. The objective is to design a quaternion-based polynomial controller for stabilizing the pitch angle of the Quanser helicopter, and then compare it with the nonlinear MRP-based controller and the PID controller. The quaternion-based attitude parameterization for a one Degree of Freedom (DOF) rigid body is the simplified version of (3.16) as

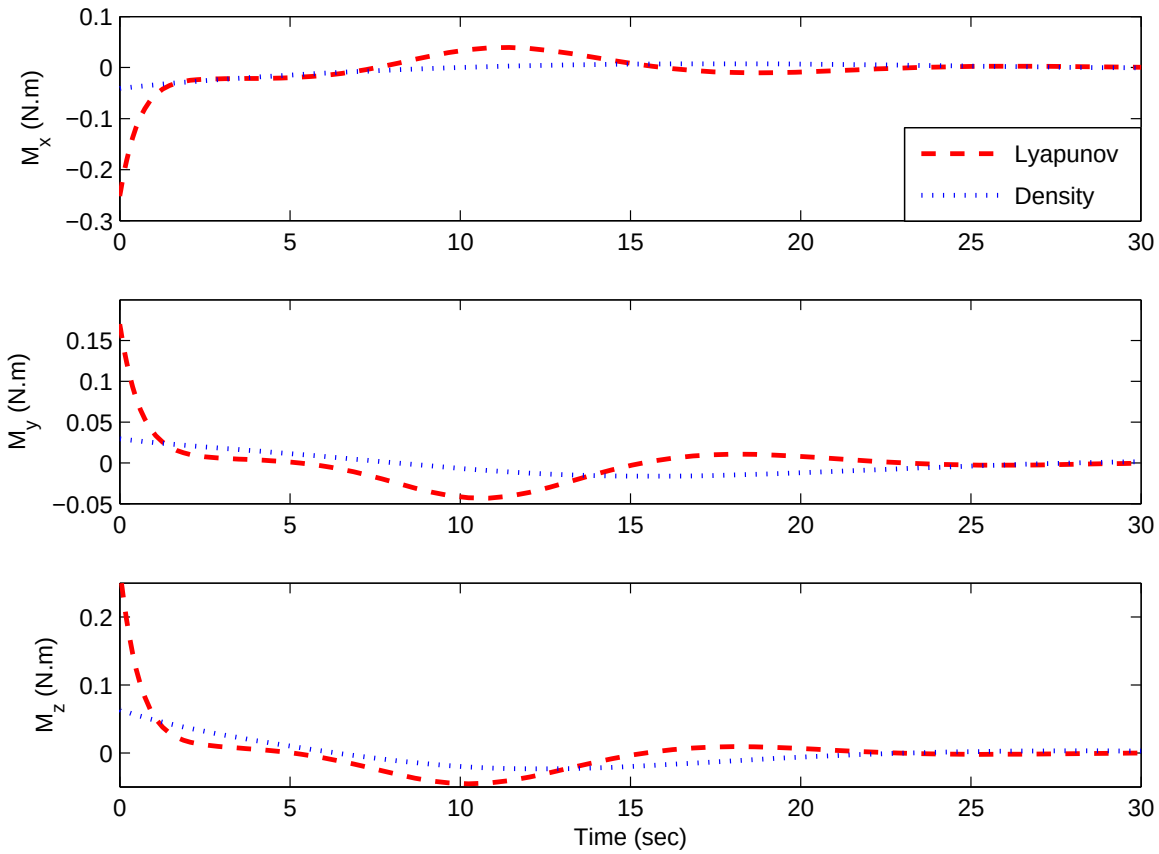

Figure 3.3: Time response of control inputs for SoS Lyapunov based control and SoS density function based control

$$\begin{bmatrix} \dot{\hat{q}}_o \\ \dot{q}_2 \\ \dot{w}_y \end{bmatrix} = \begin{bmatrix} 0 & 0 & -q_2/2 \\ 0 & 0 & (\hat{q}_o+1)/2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{q}_o \\ q_2 \\ w_y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/I_y \end{bmatrix} M_y, \tag{3.56}$$

where I_y for the Quanser helicopter is $0.028(kg.m^2)$. Given (3.56) and assuming $\varepsilon = 0.01$,

the following quaternion-based polynomial control input is obtained

$$M_y = -0.02w_y^3 - 0.03w_y(q_o-1)^2 - 0.02w_yq_2^2$$
$$-0.03w_y - 0.01q_2\,.$$

The Quanser helicopter experimental results for the pitch angle and the control input are shown in *Fig*.3.6. It shows that the time trajectory of the pitch angle and the control input for the PID controller is more oscillatory than both quaternion-based and

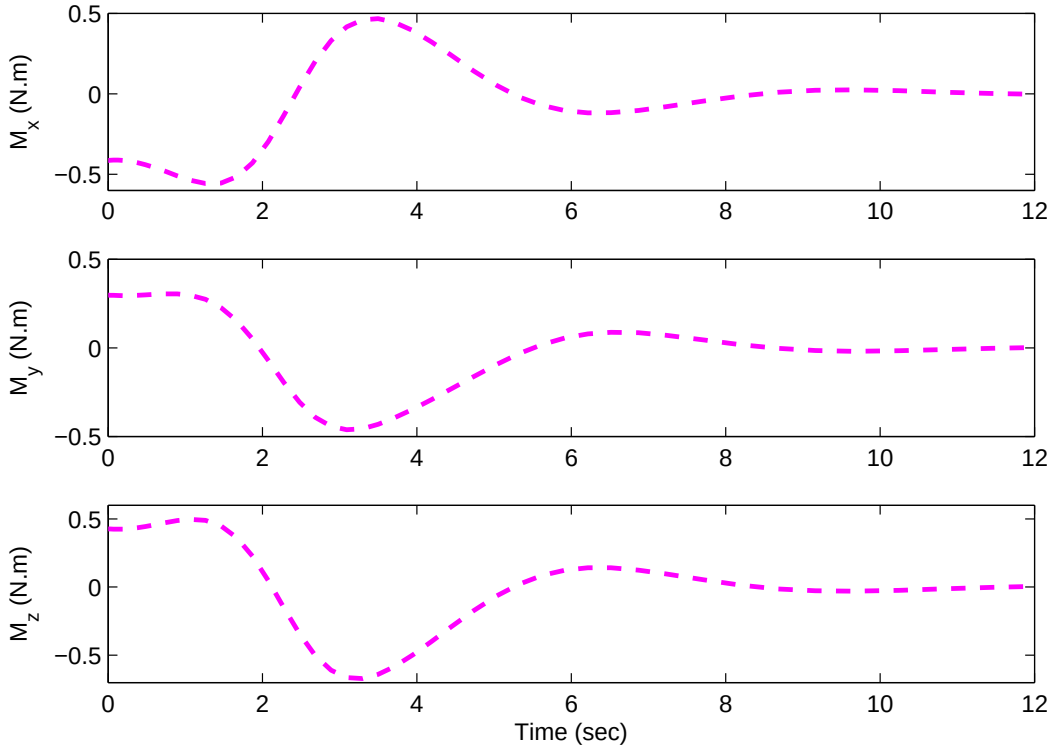

Figure 3.4: Time response of control inputs for backstepping approach

MRP-based SoS controllers. While the two quaternion-based and MRP-based controllers have been designed using the SoS technique, a tuned PID controller has also been found experimentally. Note that, contrary to the tuned linear controller, the resulting quaternion-based and MRP-based controllers have been implemented on Quanser helicopter without being tuned. *Table*3.1 also shows the maximum overshoot and settling time due to a 25 (deg) initial pitch angle for the three different controllers implemented in the Quanser Helicopter. The results demonstrate experimentally that the quaternion-based state feedback controller stabilizes the closed-loop system in less settling time and with a smaller overshoot than the MRP-based SoS controller and the PID controller.

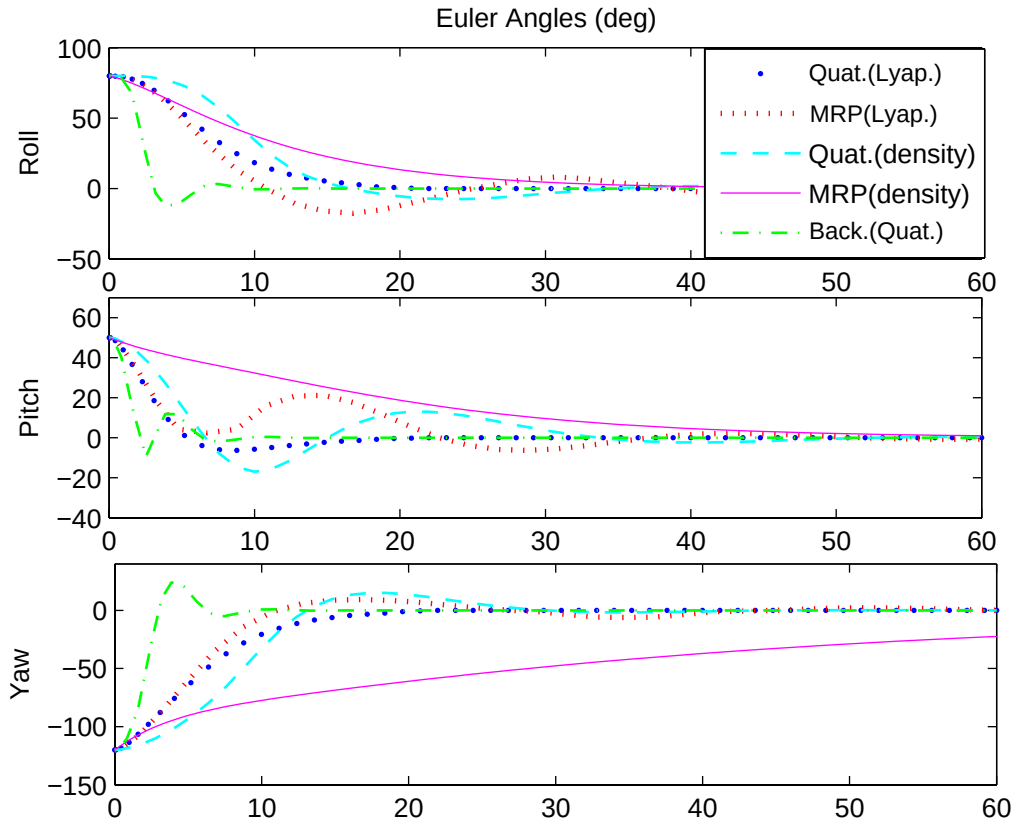

Figure 3.5: Time response of Euler angles using Quaternions and MRP parameters for SoS Lyapunov based control, SoS density function based control, and backstepping

3.6 Summary

The objective of this chapter was to develop nonsingular rigid-body attitude control laws using a convex formulation, and to implement them in an experimental set up. The attitude recovery problem was first parameterized in terms of quaternions, and then two polynomial controllers using an SoS Lyapunov function and an SoS density function were

Table 3.1: Max. overshoot and settling time due to 25 (deg) initial pitch angle for two SoS Lyapunov-based controllers and a PID controller implemented in the Quanser Helicopter

Control Method	Max. Overshoot(deg)	Settling Time (sec)
SoS (quaternions)	5	5
SoS (MRPs)	10	11
PID	13	16

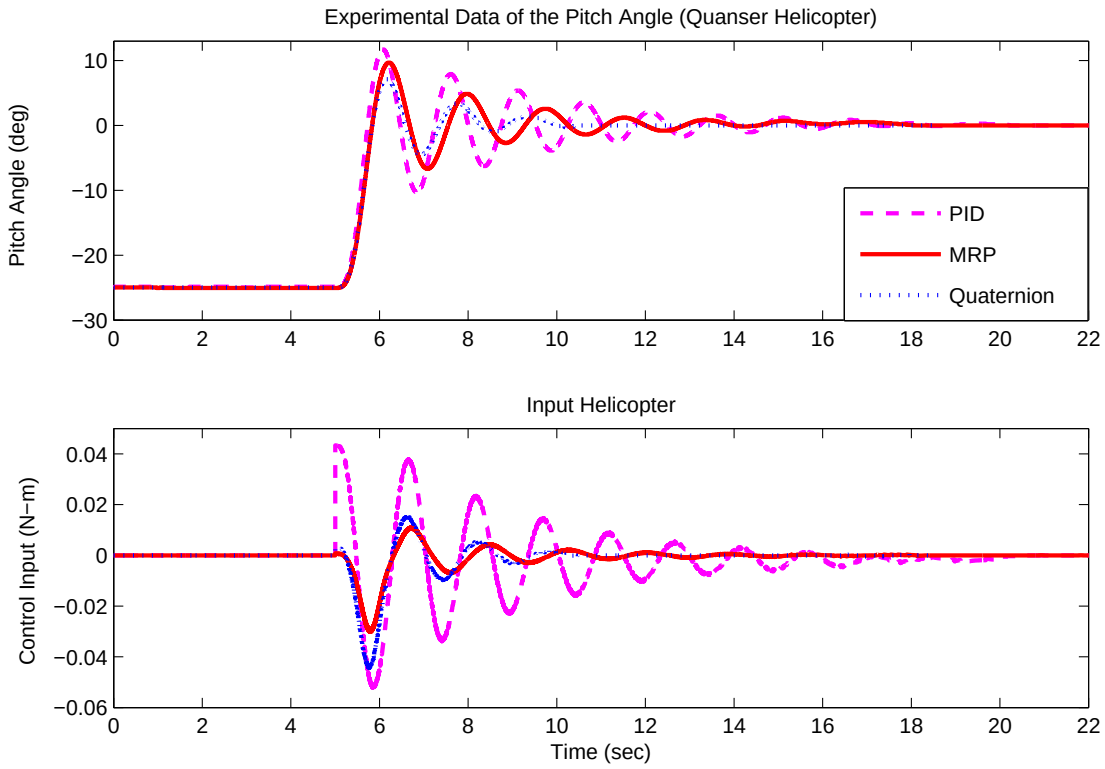

Figure 3.6: Comparison of time trajectory of pitch angle and Control Input for Quanser Helicopter using Quaternion-based and MRP-based polynomial controllers (SoS Lyapunov-based approach), and PID controller

developed. A quaternion-based polynomial controller using backstepping has also been designed. The simulation results show that the proposed nonlinear controllers guarantee the asymptotic stability of states subject to any initial condition. Moreover, the numerical simulation as well as the experimental results implemented in a Quanser Helicopter verify that the quaternion-based controller stabilizes the closed-loop system in less settling time and with smaller overshoot than the MRP-based controller. These results have been observed based on our specific simulations and experimental results on the Quanser helicopter with respect to a set of different initial conditions.

3.7 Appendix

The MATLAB code for designing an SoS Lyapunov-Based Controller using SOSTOOLS and SeDuMi is as follows:

```
clc
clear all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Define Performance Parameters and Polynomial Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% x1=wx ; x2=wy ; x3=wz ; x4=qo ; x5=q1 ; x6=q2 ; x7=q3
pvar x1 x2 x3 x4 x5 x6 x7
X = [x1;x2;x3;x4;x5;x6;x7];
n = length(X);
pvar v1 v2 v3 v4 v5 v6 v7
V = [v1;v2;v3;v4;v5;v6;v7];
%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Define System Dynamics
%%%%%%%%%%%%%%%%%%%%%%%%%%
J = diag([1,1.2,0.8]);
A = [      0      ((J(5)-J(9))/J(1))*x3              0              0   0   0   0
           0                 0            ((J(9)-J(1))/J(5))*x1   0   0   0   0
((J(1)-J(5))/J(9))*x2      0                       0              0   0   0   0
       -x5/2             -x6/2                   -x7/2            0   0   0   0
     (x4+1)/2            -x7/2                    x6/2            0   0   0   0
        x7/2           (x4+1)/2                  -x5/2            0   0   0   0
       -x6/2              x5/2                 (x4+1)/2          0   0   0   0];
B = [1/J(1)     0        0
     0       1/J(5)      0
     0          0     1/J(9)
     0          0        0
     0          0        0
```

```
      0          0       0
      0          0       0];
m = size(B,2);
Program = sosprogram([X;V]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Define Lyapunov Function and Controller Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Define  P=P'
for i = 1:n,
 for j = 1:n,
   if i>=j,
      eval(['pvar p' num2str(i) '_' num2str(j)]);
      eval(['Program=sosdecvar(Program,p' num2str(i) '_' num2str(j) ');']);
      eval(['P(i,j) = p' num2str(i) '_' num2str(j) ';']);
      eval(['P(j,i) = p' num2str(i) '_' num2str(j) ';']);
   end
 end
end
% Define K(x)
for i = 1:m,
    for j = 1:n,
        [Program, Kij]=sospolyvar(Program,monomials(X,0:2));
        K(i,j) = Kij;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Define Inequalities and run SOS program
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V(x)=x^TPx  , Q=P^{-1}
I = eye(size(P));
e1=0.1;
```

```
[Program , e2]=sospolyvar(Program,monomials(X,2));
Program = sosineq(Program,V'*(P-e1*I)*V);
Program = sosineq(Program,e2);
[Program,phi]=sospolyvar(Program,[x1^2;x2^2;x3^2;x4^2;x5^2;x6^2;x7^2]);
const = x4^2+x5^2+x6^2+x7^2-1;
[Program , mono] = sospolyvar(Program,monomials(X,1));
V_dot=-V'*(P*A'+A*P+K'*B'+B*K+e2*I)*V+mono*const;
Program = sosineq(Program,V_dot);
Program = sossolve(Program);
sosgetsol(Program,e2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Retrieve Solution Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Retrieve P
for i = 1:n,
    for j = 1:n,
        P(i,j) = sosgetsol(Program,P(i,j));
    end
end
P = double(P)
% Retrieve K
for i = 1:m,
    for j = 1:n,
        K(i,j) = sosgetsol(Program,K(i,j));
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Control Input
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
U=K*inv(P)*X
```

The MATLAB code for designing an SoS density function-based Controller using SOSTOOLS and SeDuMi is also as follows:

```
clc
clear all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Variabla and Constant Definition
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pvar wx wy wz qo q1 q2 q3
x=[wx ; wy ; wz ; qo ; q1 ; q2 ; q3];
Ix = 1;Iy = 1.2;Iz = 0.8;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Dynamic Equations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f = [((Iy-Iz)/Ix)*wy*wz ;
     ((Iz-Ix)/Iy)*wz*wx ;
     ((Ix-Iy)/Iz)*wx*wy ;
     0.5*(-q1*wx-q2*wy-q3*wz)       ;
     0.5*( (qo+1)*wx-q3*wy+q2*wz)   ;
     0.5*( q3*wx+(qo+1)*wy-q1*wz)   ;
     0.5*(-q2*wx+q1*wy+(qo+1)*wz) ];
g = [ 1/Ix    0     0;
      0     1/Iy    0;
      0      0     1/Iy;
      0      0      0;
      0      0      0;
      0      0      0;
      0      0      0; ];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% SoS Programming
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
p = 10^-6;
s = 5;
program = sosprogram(x);
[program, w(1,1)] = sospolyvar(program,monomials(x,0:2));
[program, w(2,1)] = sospolyvar(program,monomials(x,0:2));
[program, w(3,1)] = sospolyvar(program,monomials(x,0:2));
p_lin=[1.0226   -0.0000    0.0000   0.0000   0.4593   0.0000   -0.0000
      -0.0000    1.0226   -0.0000   0.0000   0.0000   0.4593   -0.0000
       0.0000   -0.0000    1.0226   0.0000  -0.0000   0.0000    0.4593
       0.0000    0.0000    0.0000   1.0000   0.0000   0.0000    0.0000
       0.4593    0.0000   -0.0000   0.0000   1.0226  -0.0000   -0.0000
       0.0000    0.4593    0.0000   0.0000  -0.0000   1.0226    0.0000
      -0.0000   -0.0000    0.4593   0.0000  -0.0000   0.0000    1.0226];
t = x'*p_lin*x;
[program,phi]=sospolyvar(program,[wx^2;wy^2;wz^2;qo^2;q1^2;q2^2;q3^2]);
fg = f*p+g*w;
divergent = diff(fg(1),wx)+diff(fg(2),wy)+diff(fg(3),wz)+ ...
            diff(fg(4),qo)+diff(fg(5),q1)+diff(fg(6),q2)+ ...
            diff(fg(7),q3);
gradian = [diff(t,wx) diff(t,wy) diff(t,wz) diff(t,qo) ...
           diff(t,q1) diff(t,q2) diff(t,q3)];
const = qo^2+q1^2+q2^2+q3^2-1;
[program , mono] = sospolyvar(program,monomials(x,1));
final_f = t*divergent-s*gradian*fg-phi+mono*const;
program = sosineq(program,final_f);
program = sossolve(program);
sosgetsol(program,w(1))
sosgetsol(program,w(2))
sosgetsol(program,w(3))
```

Chapter 4

An Inverse Optimality Approach To A Third Order Optimal Control Problem

4.1 Introduction

The main contribution of this chapter is to analytically solve the Hamilton-Jacobi-Bellman equation for a class of third order nonlinear optimal control problems for which the dynamics are affine and the cost is quadratic in the input. The proposed solution method is based on the notion of inverse optimality with a variable part of the cost to be determined in the solution. The main idea was first proposed in [80] to solve a class of second order problems. This chapter will extend the work in [80] to solve a class of nonlinear third order optimal control problems. One special advantage of this work is that the solution is directly obtained for the control input without the computation of a value function first. The value function can however also be obtained based on the control input. Furthermore, a Lyapunov function can be constructed for a subclass of optimal control problems, yielding a proof certificate of stability. Finally, using the proposed methodology, experimental results for a path following problem implemented in a Wheeled Mobile Robot (WMR)

are then presented to verify the effectiveness of the proposed methodology.

4.2 Background

Optimal control problems are generally solved by numerical techniques since the optimal controller is the solution of the Hamilton-Jacobi-Bellman (HJB) equation [79], which is a nonlinear partial differential equation that is difficult to solve analytically. However, there is an explicit solution for the input as a derivative of the value function if the dynamic model is affine and the cost is quadratic in the input. This idea was first used in [80] to solve a class of second order problems. This chapter will extend the work in [80] to solve a class of nonlinear third order optimal control problems.

Departing from previous methods, the proposed method in this chapter can directly find a solution for the control input without the computation of a value function. The value function can however also be obtained based on the control input. Furthermore, a Lyapunov function can be constructed for a subclass of optimal control problems, yielding a proof certificate for stability. The method can be applied to a class of third order nonlinear systems that will be defined in the next section. It is assumed that the cost function is the sum of a quadratic term in the input and the states and an unknown term $Q(x)$ that should be determined. For a third order nonlinear system in the assumed class, our interest is then to simultaneously search for a controller and a cost function term $Q(x)$ that together satisfy the HJB equation. The methodology will be applied to the dynamic model of a Wheeled Mobile Robot (WMR) on the $x-y$ plane for path following of the line $y=0$ at a constant velocity, as shown in *Fig*.2.4. This path following problem will be investigated in section 4.4.

The remainder of this chapter is organized as follows. In section 4.3 a third order nonlinear optimal control problem is defined, and then the main result is derived. An

interesting special case of the general optimal control problem is also presented in that section. The effectiveness of the proposed method will be shown in several examples in section 4.4. Using the proposed methodology, experimental results for a path following problem implemented in a Wheeled Mobile Robot (WMR) are then presented, followed by some concluding remarks.

4.3 Optimal Control Problem Definition and Solution

Consider the following optimal control problem

$$\begin{aligned} min \quad & J(x,u) = \int_0^\infty (q_1 x_1^2 + q_2 x_2^2 + q_3 x_3^2 + Q(x) + ru^2)\,\mathrm{d}t \\ s.t. \quad & \dot{x}_1(t) = f_1(x_2) \\ & \dot{x}_2(t) = f_2(x_3) \\ & \dot{x}_3(t) = cu \\ & x(0) = x_0, \quad u \in \mathscr{U} \end{aligned} \tag{4.1}$$

where $c > 0$, $q_1 \geq 0$, $q_2 \geq 0$, $q_3 > 0$, $r > 0$, $x(t) = [x_1 \quad x_2 \quad x_3]^T \in \mathscr{W}^3 \subset \mathscr{R}^3$ is the state vector, where $\mathscr{W}^3$ includes a neighborhood of the origin. The scalar input u belongs to the set $\mathscr{U}$ of Lebesgue integrable functions. The function $f_1(x_2)$ is class $\mathscr{C}^1$ with a bounded derivative and $f_2(x_3)$ is continuous. These functions $f_1(x_2)$ and $f_2(x_3)$ are not identically zero and are assumed to be zero at $x = 0$ $\big(f_1(0) = f_2(0) = 0\big)$. The term

$$L(x_1,x_2,x_3,u) = q_1 x_1^2 + q_2 x_2^2 + q_3 x_3^2 + Q(x) + ru^2 \tag{4.2}$$

which is a function of all the states and the input, is called the running cost. The optimal control problem formulated here is to find, if possible, a control law $u(x)$ and a cost function $L(x_1,x_2,x_3,u)$ such that u minimizes the performance index $J(x,u) = \int_0^\infty L\,\mathrm{d}t$,

and (4.2) is nonnegative and has a minimum at $x_1 = x_2 = x_3 = u = 0$. Let the optimal cost function be defined by

$$V(x_1, x_2, x_3) = \inf_u \int_0^\infty L(x_1, x_2, x_3, u)\, \mathrm{d}t \tag{4.3}$$

The main result is now stated.

Theorem 4.3.1. *Given the optimal control problem (4.1), if there exist gains k_1, k_2, k_3, k_4, and k_5 satisfying*

$$k_1^2 = \frac{q_1}{r} \quad , \quad k_2^2 = \frac{q_2}{r} \quad , \quad k_3^2 = \frac{q_3}{r}, \tag{4.4}$$

$$k_1 = c k_3 k_4 \quad , \quad k_2 = c k_3 k_5 \tag{4.5}$$

$$k_4^2 f_1^2(x_2) + k_5^2 f_2^2(x_3) - 2c^{-1} k_4 f_1'(x_2) x_3 f_2(x_3) \geq 0 \tag{4.6}$$

and

$$2rc^{-1} k_5 \int f_2(x_3)\, \mathrm{d}x_3 + 2rc^{-1} k_4 x_3 f_1(x_2) + 2r k_4 k_5 \int f_1(x_2)\, \mathrm{d}x_2 + \gamma \geq 0 \tag{4.7}$$

where γ is an integration constant verifying

$$\gamma = -2rc^{-1} \left[\int k_5 f_2(x_3)\, \mathrm{d}x_3 + c k_4 k_5 \int f_1(x_2)\, \mathrm{d}x_2 \right]_{x_2 = x_3 = 0} \tag{4.8}$$

then the control input

$$u = -k_1 x_1 - k_2 x_2 - k_3 x_3 - k_4 f_1(x_2) - k_5 f_2(x_3) \tag{4.9}$$

solves the HJB equation for problem (4.1) with

$$\begin{aligned} Q(x) \; = & r k_4^2 f_1^2(x_2) + r k_5^2 f_2^2(x_3) - 2rc^{-1} k_4 x_3 f_1'(x_2) f_2(x_3) \\ & + 2r k_1 k_2 x_1 x_2 + 2r k_1 k_3 x_1 x_3 + 2r k_2 k_3 x_2 x_3 \end{aligned} \tag{4.10}$$

which yields the nonnegative running cost

$$\begin{aligned} L(x_1, x_2, x_3, u) \; = & r k_4^2 f_1^2(x_2) + r k_5^2 f_2^2(x_3) - 2rc^{-1} k_4 x_3 f_1'(x_2) f_2(x_3) \\ & + r(k_1 x_1 + k_2 x_2 + k_3 x_3)^2 + r u^2 \end{aligned} \tag{4.11}$$

with a minimum at $x_1 = x_2 = x_3 = u = 0$. *The resulting optimal cost function* $V(x) = J(x,u^*)$, *where* u^* *is the optimal controller, will be given by*

$$\begin{aligned} V(x) &= r\left(\sqrt{k_1k_4}x_1 + \sqrt{k_2k_5}x_2 + \sqrt{k_3c^{-1}}x_3\right)^2 \\ &\quad + 2rc^{-1}\left(k_5\int f_2(x_3)\,\mathrm{d}x_3 + k_4x_3f_1(x_2)\right) \\ &\quad + 2rk_4k_5\int f_1(x_2)\,\mathrm{d}x_2 + \gamma. \end{aligned} \tag{4.12}$$

The function $V(x)$ *will also be a local Lyapunov function for the system (4.1) provided it is positive definite. Furthermore, the trajectories will converge to one of the minimizers of* $L(x_1,x_2,x_3,u(x_1,x_2,x_3))$, *i.e, to a point* (x_1,x_2,x_3) *such that*

$$L(x_1,x_2,x_3,u(x_1,x_2,x_3)) = 0\,. \tag{4.13}$$

If $L(x_1,x_2,x_3,u(x_1,x_2,x_3))$ *is convex, then the trajectories will converge to the origin for all initial conditions.*

Proof. To solve the optimal control problem (4.1), the HJB equation

$$\inf_u \quad H(x_1,x_2,x_3,V_{x_1},V_{x_2},V_{x_3},u) = 0 \tag{4.14}$$

where

$$\begin{aligned} H &= q_1x_1^2 + q_2x_2^2 + q_3x_3^2 + Q(x) + V_{x_1}f_1(x_2) \\ &\quad + V_{x_2}f_2(x_3) + V_{x_3}cu + ru^2 \end{aligned} \tag{4.15}$$

with

$$V_{x_i} = \frac{\partial V(x)}{\partial x_i}, \quad i = 1,2,3 \tag{4.16}$$

should be solved. A necessary condition for optimality is

$$\frac{\partial H}{\partial u} = 0 \;\longrightarrow\; V_{x_3} = -2rc^{-1}u(x)\,. \tag{4.17}$$

Using a structure of the control input as in (4.9), the integral of expression (4.17) yields

$$\begin{aligned} V(x) =& -2rc^{-1}\int u(x)\,\mathrm{d}x_3 + h(x_1,x_2) \\ =& +2rc^{-1}x_3\big(k_1x_1+k_2x_2+k_4f_1(x_2)\big) \\ &+2rc^{-1}\int\big(k_3x_3+k_5f_2(x_3)\big)\,\mathrm{d}x_3 + h(x_1,x_2) \end{aligned} \tag{4.18}$$

where $h(x_1,x_2)$ is an arbitrary function of x_1 and x_2. Differentiating with respect to x_1 and x_2 yields

$$V_{x_1} = 2rc^{-1}k_1x_3 + h_{x_1} \tag{4.19}$$

$$V_{x_2} = 2rc^{-1}x_3\big(k_2+k_4f_1'(x_2)\big)+h_{x_2} \tag{4.20}$$

where $f_1'(x_2)$, h_{x_1}, h_{x_2} are the derivatives of $f_1(x_2)$ and $h(x_1,x_2)$ with respect to x_2, x_1, and x_2, respectively. Replacing (4.9), (4.17), (4.19) and (4.20) in (4.14) yields

$$\begin{aligned} &Q(x)+(q_1-rk_1^2)x_1^2+(q_2-rk_2^2)x_2^2+(q_3-rk_3^2)x_3^2 \\ &+2rc^{-1}k_1x_3f_1+2rc^{-1}x_3f_2(k_2+k_4f_1')+h_{x_1}f_1+h_{x_2}f_2 \\ &-2rk_1k_2x_1x_2-2rk_1k_3x_1x_3-2rk_2k_3x_2x_3 \\ &-2rk_4k_3x_3f_1-2rk_5k_3x_3f_2-r(k_4^2f_1^2+k_5^2f_2^2) \\ &-2rk_4f_1(k_1x_1+k_2x_2)-2rk_5f_2(k_1x_1+k_2x_2) \\ &-2rk_3k_4x_3f_1-2rk_3k_5x_3f_2=0 \end{aligned} \tag{4.21}$$

where the arguments were removed for simplicity. Choosing

$$h_{x_2} = 2rk_4k_5f_1+2rk_5(k_1x_1+k_2x_2), \tag{4.22}$$

yields

$$\begin{aligned} h(x_1,x_2) =& 2rk_4k_5\int f_1(x_2)\,\mathrm{d}x_2+2rk_5k_1x_1x_2 \\ &+rk_5k_2x_2^2+g(x_1) \end{aligned} \tag{4.23}$$

and

$$h_{x_1} = 2rk_5k_1x_2 + g'(x_1) \tag{4.24}$$

where we choose

$$g'(x_1) = 2rk_4k_1x_1\,. \tag{4.25}$$

Finally, replacing (4.22)and (4.24) in (4.21) yields after rearranging

$$\begin{aligned} &Q(x) + (q_1 - rk_1^2)x_1^2 + (q_2 - rk_2^2)x_2^2 + (q_3 - rk_3^2)x_3^2 \\ &+ 2rx_3 f_1(k_1c^{-1} - k_3k_4) + 2rx_2 f_1(k_5k_1 - k_2k_4) \\ &+ 2rx_3 f_2(k_2c^{-1} - k_3k_5) - 2rk_1k_2x_1x_2 - 2rk_1k_3x_1x_3 \\ &- 2rk_2k_3x_2x_3 + 2rc^{-1}k_4x_3 f_1' f_2 - rk_4^2 f_1^2 - rk_5^2 f_2^2 = 0\,. \end{aligned} \tag{4.26}$$

Using (4.4) and (4.5) in (4.26) leads to the expression (4.10). Combining (4.2), (4.6) and (4.10) yields the nonnegative running cost (4.11). Replacing (4.23) in (4.18) and taking into account (4.5) yields the value function (4.12). Notice that V is class $\mathscr{C}^1$ given the continuity and smoothness assumptions on the functions $f_1(x_2)$ and $f_2(x_3)$. Therefore, the optimal cost $V(x)$ in (4.12) is finite for any bounded initial condition x. Moreover, since $V(x) = \int_0^\infty L\,\mathrm{d}t$, where $L \geq 0$, $V(x)$ needs to be a nonnegative function. The condition (4.7) implies $V(x) \geq 0$. Notice also that $\dot{V} = -L(x_1, x_2, x_3, u) \leq 0$. Therefore, the cost function $V(x)$ becomes a Lyapunov function for the system dynamics in (4.1) provided it is positive definite. Finally, since the optimal cost $V(x)$ is finite for all initial conditions, then the trajectories will converge to one of the minimizers of $L(x_1, x_2, x_3, u(x_1, x_2, x_3))$ because $L \geq 0$ and $\lim_{t\to\infty} L = 0$ (since the integral of L is finite). If L is convex, then the trajectories must converge to the origin because the origin is the only minimizer of L. Expression (4.8) makes the cost function $V(x)$ zero at the equilibrium point $x = 0$ satisfying the boundary condition of the HJB $x(\infty) = 0$. This finishes the proof. □

Remark 4.3.1. *It is interesting that the square of the nonlinearity terms $f_1^2(x_2)$ and $f_2^2(x_3)$ appear in the running cost function, though this would be difficult to predict. In fact, in most of the research papers on optimal control the cost usually includes only quadratic terms on the states.*

An interesting special case of the optimal control problem (4.1) is presented in the following corollary where $f_2(x_3) = a x_3$. This case can be applied to some important mobile robotics such as path following problems, which will be shown in the next section.

Corollary 1. *For the following optimal control problem*

$$\begin{aligned} min \quad & J(x,u) = \int_0^\infty (q_1 x_1^2 + q_2 x_2^2 + q_3 x_3^2 + Q(x) + ru^2)\,\mathrm{d}t \\ s.t. \quad & \dot{x}_1(t) = f(x_2) \\ & \dot{x}_2(t) = a x_3 \\ & \dot{x}_3(t) = cu \\ & x(0) = x_0, \quad u \in \mathscr{U} \end{aligned} \tag{4.27}$$

where $a > 0$, $c > 0$, $f(0) = 0$, $q_1 \geq 0$, $q_2 \geq 0$, $q_3 > 0$, and $r > 0$, if there exist gains k_i for $i = 1, \cdots, 5$ verifying (4.4), (4.5),

$$k_4 > 0 \quad , \quad f'(x_2) \leq 0.5 a c k_5^2 k_4^{-1} \tag{4.28}$$

and

$$r a c^{-1} k_5 x_3^2 + 2 r c^{-1} k_4 x_3 f(x_2) 2 r k_4 k_5 \int f(x_2)\,\mathrm{d}x_2 + \gamma \geq 0 \tag{4.29}$$

where γ is an integration constant verifying $V(0) = 0$, then the control input (4.9) solves the HJB equation corresponding to (4.27) with the nonnegative running cost

$$\begin{aligned} L(x_1,x_2,x_3,u) = & r(k_1 x_1 + k_2 x_2 + k_3 x_3)^2 + r k_4^2 f^2(x_2) \\ & + r x_3^2 \big(a^2 k_5^2 - 2 a c^{-1} k_4 f'(x_2)\big) + r u^2 . \end{aligned} \tag{4.30}$$

Moreover, the resulting value function is

$$\begin{aligned} V(x) \quad &= \quad r\left(\sqrt{k_1k_4}\,x_1+\sqrt{k_2k_5}\,x_2+\sqrt{k_3c^{-1}}\,x_3\right)^2 \\ &\quad + rac^{-1}k_5x_3^2+2rc^{-1}k_4x_3f(x_2) \\ &\quad + 2rk_4k_5\int f(x_2)\,\mathrm{d}x_2+\gamma. \end{aligned} \tag{4.31}$$

The function V is also a local Lyapunov function provided it is positive definite. Furthermore, the trajectories will converge to one of the minimizers of $L(x_1,x_2,x_3,u(x_1,x_2,x_3))$, i.e, to a point (x_1,x_2,x_3) such that $L(x_1,x_2,x_3,u(x_1,x_2,x_3))=0$. If $L(x_1,x_2,x_3,u(x_1,x_2,x_3))$ is convex, then the trajectories will converge to the origin for all initial conditions.

Proof. Making $f_2(x_3)=ax_3$ expressions (4.11) and (4.12) result in (4.30) and (4.31), respectively. Moreover, since $V(x)=\int_0^\infty L\,\mathrm{d}t$, where $L\geq 0$, $V(x)$ needs to be a positive semidefinite function. The condition (4.29) implies $V(x)\geq 0$. If the function $V(x)$ is positive definite, it will be a Lyapunov function since constraints (4.28) imply that $\dot{V}=-L(x_1,x_2,x_3,u)\leq 0$. The rest of the proof follows the same argument as the proof of theorem 4.3.1. □

In the next section, the effectiveness of the proposed method will be shown in several examples.

4.4 Examples and Numerical Simulations

Example 4.4.1. Linear System

Consider a triple backstepping integrator system with

$$c=1 \quad , \quad f_1(x_2)=x_2 \quad , \quad f_2(x_3)=x_3\,. \tag{4.32}$$

This linear system has open-loop equilibrium points at

$$x_1 = constant \quad , \quad x_2 = x_3 = 0\,. \tag{4.33}$$

Assuming $q_1 = q_3 = r = 1$ and $q_2 = 4$, the gains satisfying (4.4)-(4.7) are obtained as follows

$$k_1 = k_3 = k_4 = 1\,, k_2 = k_5 = 2\,. \tag{4.34}$$

This results in the control input

$$u = -x_1 - 3x_2 - 3x_3$$

and the running cost is given by

$$L(x_1, x_2, x_3, u) = (x_1 + 2x_2 + x_3)^2 + x_2^2 + 2x_3^2 + u^2\,.$$

This running cost function $L(x)$ is strictly convex because the Hessian matrix of $L(x)$ is positive definite as follows

$$\nabla^2 L(x) = \begin{bmatrix} 4 & 10 & 8 \\ 10 & 28 & 22 \\ 8 & 22 & 24 \end{bmatrix} > 0\,. \tag{4.35}$$

The value function is also

$$\begin{aligned} V(x) =& (x_1 + 2x_2 + x_3)^2 + x_3^2 + x_2^2 + (x_2 + x_3)^2 \\ =& x^T P x = x^T \begin{bmatrix} 1 & 2 & 1 \\ 2 & 6 & 3 \\ 1 & 3 & 3 \end{bmatrix} x\,. \end{aligned}$$

Obviously, the function $V(x)$ is positive definite since $P > 0$. This value function is also radially unbounded. Moreover, the derivative of the value function is

$$\begin{aligned} \dot{V}(x) =& -(x_1 + 2x_2 + x_3)^2 - x_2^2 - 2x_3^2 - (x_1 + 3x_2 + 3x_3)^2 \\ =& x^T Z x \end{aligned}$$

where

$$Z = \begin{bmatrix} -2 & -5 & -4 \\ -5 & -14 & -11 \\ -4 & -11 & -12 \end{bmatrix} \tag{4.36}$$

Thus, $\dot{V}(x)$ is negative definite since $Z < 0$. Therefore, the value function is a global Lyapunov function, and the system is globally asymptotically stable. Note that for a triple integrator model using the control input (4.9), an usual quadratic form of the states $V(x) = x_1^2 + x_2^2 + x_3^2$ cannot be a Lyapunov function, and thus there should exist cross terms in the states.

Example 4.4.2. Nonlinear System

Consider a nonlinear system with

$$c = 1 \quad , \quad f_1(x_2) = 2x_2 + \sin(x_2) \quad , \quad f_2(x_3) = x_3^3 + x_3 \, . \tag{4.37}$$

This nonlinear system has the open-loop equilibrium points at

$$x_1 = constant \quad , \quad x_2 = x_3 = 0 \, . \tag{4.38}$$

The control gains corresponding to

$$q_1 = q_3 = r = 1 \quad , \quad q_2 = 9 \tag{4.39}$$

are

$$3k_1 = k_2 = 3k_3 = 3k_4 = k_5 = 3 \tag{4.40}$$

which satisfy all the constraints (4.4)-(4.7). Using (4.9), the control input is given by

$$u = -x_1 - 5x_2 - 4x_3 - \sin(x_2) - 3x_3^3 \tag{4.41}$$

with the running cost

$$\begin{aligned} L(x_1,x_2,x_3,u) =&(x_1+3x_2+x_3)^2+9(x_3^3+x_3)^2+u^2 \\ &-2(2+\cos(x_2)x_3^2(x_3^2+1) \\ =&(x_1+3x_2+x_3)^2+\big(2x_2+\sin(x_2)\big)^2+u^2 \\ &+x_3^2(x_3^2+1)\bigg(9(x_3^2+1)-2\big(2+\cos(x_2)\big)\bigg). \end{aligned} \tag{4.42}$$

Since the resulting running cost is a sum of squares, and has only one minimizer at the origin satisfying (4.13), it is positive definite for all the states $x \in \mathscr{R}^3$. Moreover, the value function with γ given by (4.8) is obtained as

$$\begin{aligned} V(x) =&(x_1+3x_2+x_3)^2+1.5x_3^4+3x_3^2+6x_2^2 \\ &+6\big(1-\cos(x_2)\big)+2x_3\big(2x_2+\sin(x_2)\big) \\ =&(x_1+3x_2+x_3)^2+x_3^2+2x_2^2+1.5x_3^4 \\ &+(x_3+2x_2)^2+\big(x_3+\sin(x_2)\big)^2 \\ &+4sin^2(\tfrac{x_2}{2})\big(2+sin^2(\tfrac{x_2}{2})\big) \\ \geq& W(x) = x1^2+x2^2+x3^4, \end{aligned}$$

Since $V(x) \geq W(x) > 0$, the resulting value function is positive definite. The value function $V(x)$ is also radially unbounded. Therefore, Since $\dot{V}(x) = -L(x,u) < 0$, this value function is a global Lyapunov function. The equilibrium point $x = 0$ is globally asymptotically stable. *Fig*.4.1 shows the time response of the states $x = [x_1 \quad x_2 \quad x_3]^T$, the running cost $L(x,u)$, and the Lyapunov function $V(x)$ subject to the initial condition $x_o = [10 \quad \pi/2 \quad 1]^T$. As seen, all the states asymptotically converge to the origin, where $V(x) > 0$, $V(\infty) = 0$, and the value function has the minimum at $x_1 = x_2 = x_3 = u = 0$.

For comparison, an LQR controller is also designed for a linearized model of the

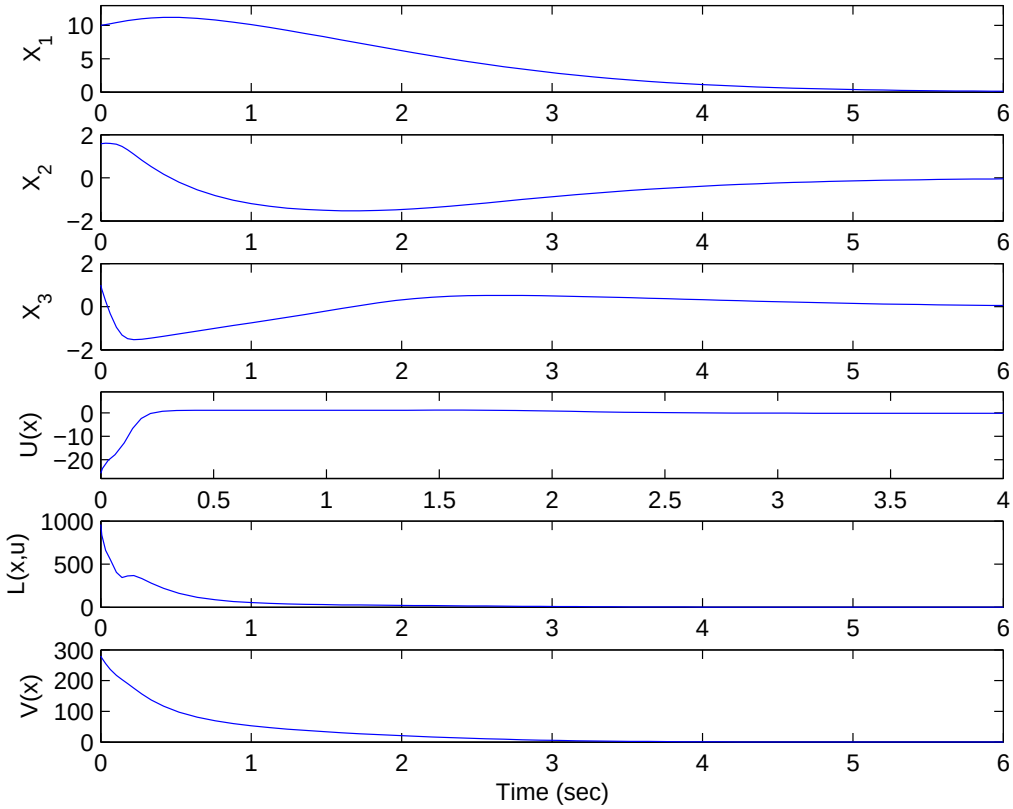

Figure 4.1: Time response of the states, the input, the running cost $L(x,u)$, and the Lyapunov function $V(x)$ for Example 4.4.2 subject to initial condition $x_o = [10 \quad \pi/2 \quad 1]^T$

given nonlinear system with the weighting matrices in the running cost function corresponding to a second order Taylor series approximation of the running cost. The linearized model is given by $\dot{x} = Ax + Bu$, where

$$A = \begin{bmatrix} 0 & 3 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

and $B = [0 \quad 0 \quad 1]^T$. Using a second order Taylor series approximation of the resulting running cost (4.42) yields

$$\begin{aligned} L(x_1,x_2,x_3,u) =& x_1^2 + 18x_2^2 + 4x_3^2 + 6x_1x_2 + 2x_1x_3 + 6x_2x_3 + u^2 \\ =& x^T Qx + u^T Ru \end{aligned} . \tag{4.43}$$

Now we consider two weighting matrices with and without having cross terms in the quadratic Lyapunov function $L(x)$ (4.43) as follows

$$Q_{LQR_{diag}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad Q_{LQR_{cross}} = \begin{bmatrix} 1 & 3 & 1 \\ 3 & 18 & 3 \\ 1 & 3 & 4 \end{bmatrix}$$

and $R = 1$. The $LQR(A, B, Q, R)$ controllers are obtained as

$$\begin{aligned} K_{LQR_{diag}} &= [1.0000 \quad 4.6766 \quad 3.2176] \\ K_{LQR_{cross}} &= [1.0000 \quad 5.4233 \quad 3.8531]. \end{aligned} \tag{4.44}$$

The time response of the states and the control input for the LQR and the proposed HJB-based controller (4.9) subject to the initial condition $x_o = [10 \quad \pi \quad 1]^T$ is shown in *Fig*.4.2. It indicates that, compared to the LQR controllers, using the proposed optimal controller stabilizes the system with less overshoot in the states, and also with less control effort. Simulation results also show that the bigger the initial condition is, the more oscillatory is the response of the system using LQR controllers. On the other hand, using the proposed optimal controller the change of initial conditions do not affect the smoothness of state responses and the control effort.

These controllers (optimal and LQR controllers) are also compared in the presence of a partial loss of control authority, shown in *Fig*.4.3. *Fig*.4.3 shows the time response of states and control inputs with 85% loss of control authority subject to the initial condition $x_o = [10 \quad \pi \quad 1]^T$. Simulation results show that the proposed HJB-based control works with up to 85% loss of control authority, whereas the LQR controller designed for a linear model of the given nonlinear system with the weighting matrices in the running cost function corresponding to a second order Taylor series approximation of the running cost cannot stabilize the system at this percentage of failure.

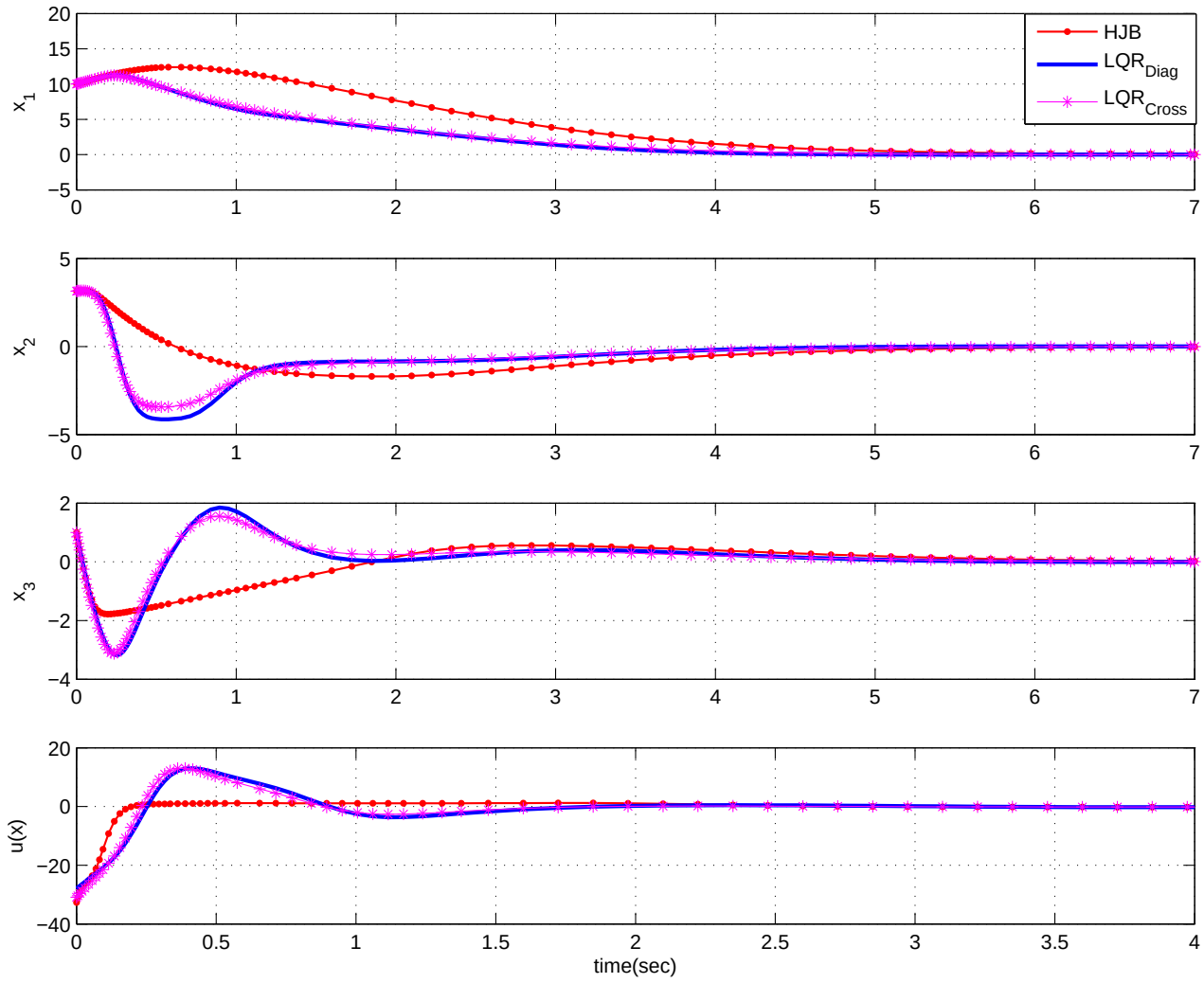

Figure 4.2: Time response of the states and the control input for LQR and HJB-based controllers for Example 4.4.2 subject to the initial condition $x_o = [10 \quad \pi \quad 1]^T$

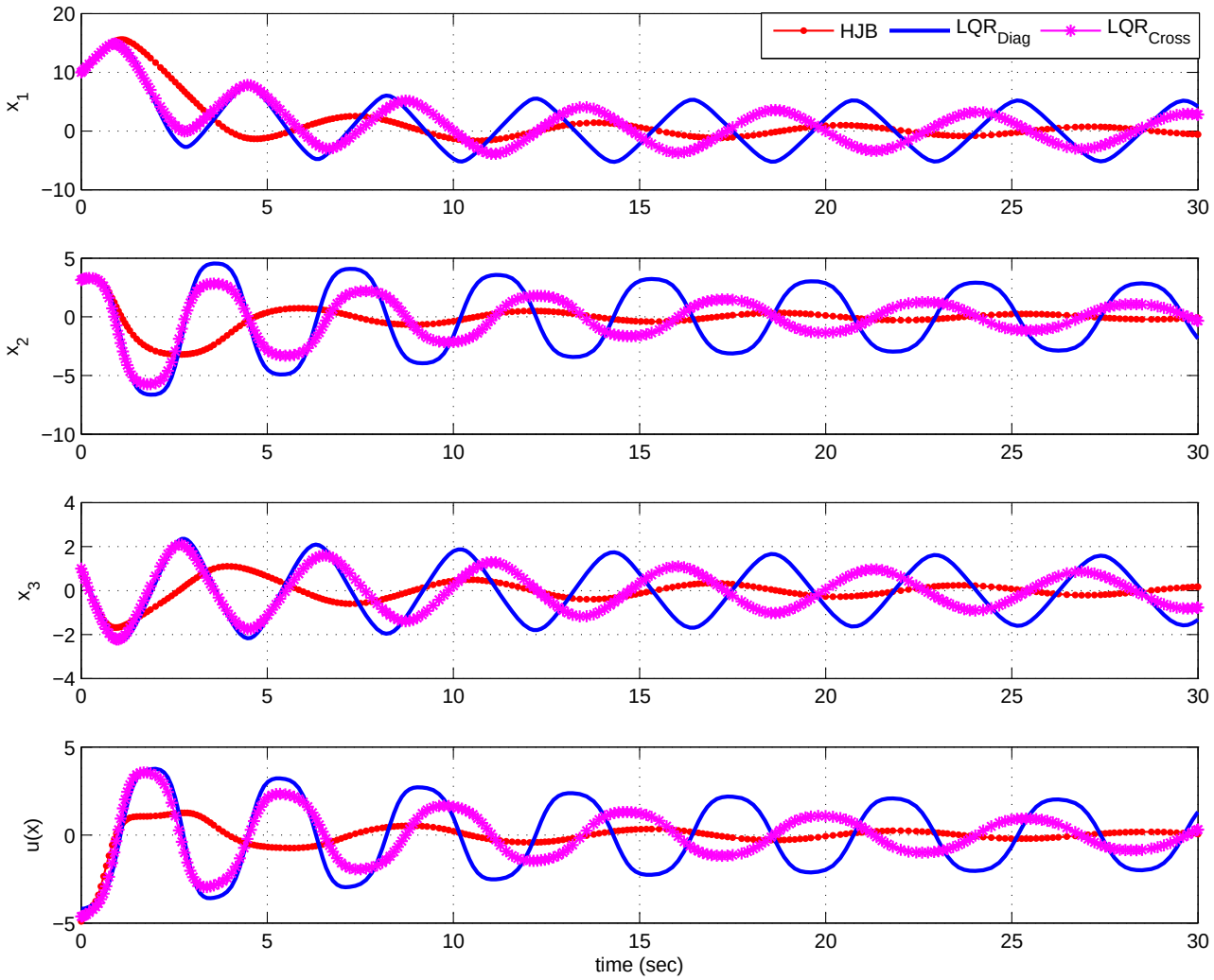

Figure 4.3: Time response of states and control inputs with 85% loss of control authority for HJB-based controller and LQR controller subject to the initial condition $x_o = [10 \quad \pi \quad 1]^T$

Example 4.4.3. Wheeled Mobile Robot (WMR) Simulation

Following corollary 1, let $a = c = 1,\ f(x_2) = \sin(x_2)$. This system is the dynamic model for path following of the line $y = 0$ of a WMR on a plane, moving at a constant unitary velocity. According to *Fig*.2.4, the state vector

$$x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T = \begin{bmatrix} y & \psi & w \end{bmatrix}^T \tag{4.45}$$

contains the position y, the heading angle $\psi \in (-\pi, \pi]$, and the angular velocity $\dot{\psi}$, respectively. This nonlinear system has the open-loop equilibrium points at

$$y = constant \quad , \quad w = 0 \quad , \quad \psi = 0, \pi\,. \tag{4.46}$$

for $k = 0, 1, 2, \cdots$. The optimal controller corresponding to $q_1 = q_3 = r = 1$ and $q_2 = 4$ is

$$u = -x_1 - 2x_2 - 3x_3 - \sin(x_2)\,,$$

with the following running cost

$$\begin{aligned} L(x_1, x_2, x_3, u) =& (x_1 + 2x_2 + x_3)^2 + \sin^2(x_2) \\ &+ x_3^2\big(4 - 2\cos(x_2)\big) + u^2\,. \end{aligned} \tag{4.47}$$

The running cost $L(x)$ is a nonnegative function since it is a sum of squares. Note that for $x_2 \in (-\pi, \pi]$ the resulting running cost function $L(x_1, x_2, x_3, u(x))$ has two minimizer at the origin $(0,0,0)$ and the point $(-2\pi, \pi, 0)$. Furthermore, the value function with γ given by (4.8) is obtained as

$$\begin{aligned} V(x) =& (x_1 + 2x_2 + x_3)^2 + 2x_3^2 + 2x_3 \sin(x_2) \\ &+ 4\big(1 - \cos(x_2)\big) \\ =& (x_1 + 2x_2 + x_3)^2 + \big(x_3 + \sin(x_2)\big)^2 + x_3^2 \\ &+ 4\sin^2\big(\tfrac{x_2}{2}\big)\big(1 + \sin^2\big(\tfrac{x_2}{2}\big)\big) \end{aligned}$$

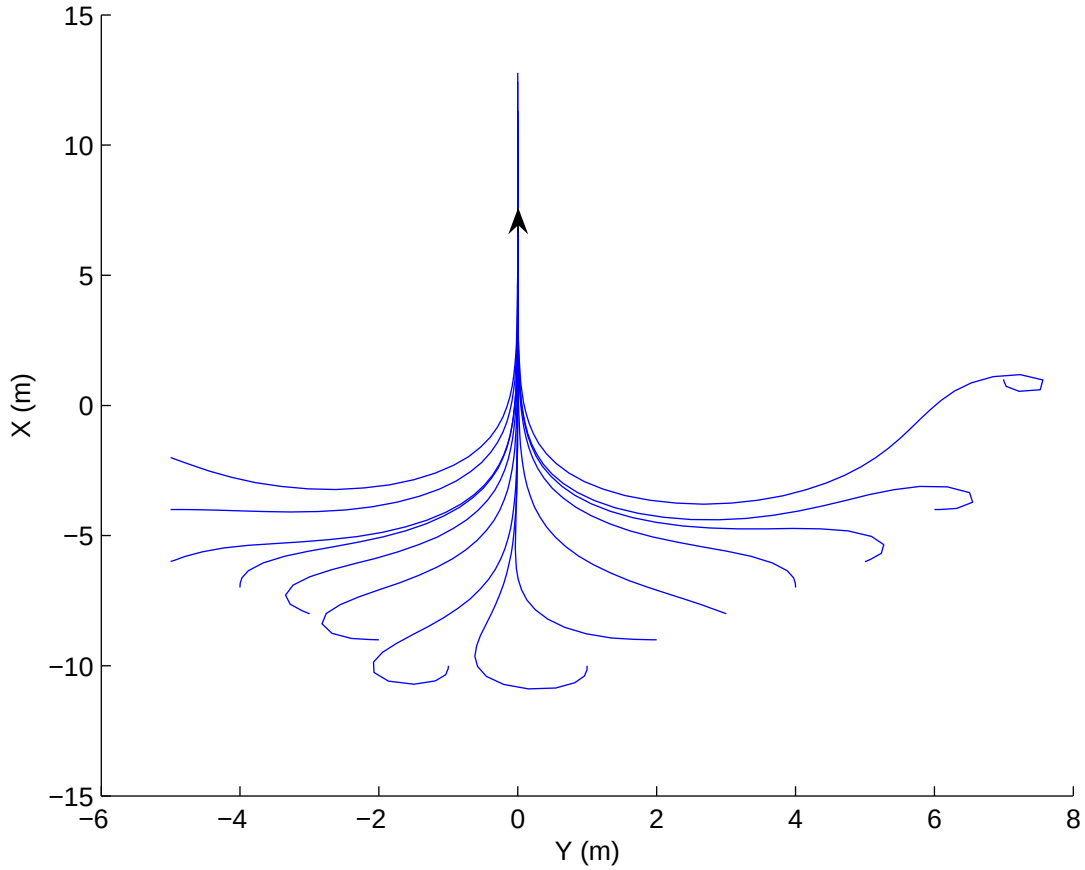

Figure 4.4: WMR Trajectories

where $V(x)$ is positive definite for $x_2 \in (-\pi, \pi)$ since the resulting value function is a sum of squares, and is equal to zero only at the origin. Moreover, the derivative of the value function

$$\dot{V}(x) = -L(x) = -(x_1 + x_2 + x_3)^2 - \sin^2(x_2) - x_3^2\big(4 - 2\cos(x_2)\big) - \big(x_1 + 2x_2 + 3x_3 + \sin(x_2)\big)^2$$

is negative definite for $x_2 \in (-\pi, \pi)$ because $L(x)$ is nonnegative, and is equal to zero only at the origin. Therefore, the value function is a local Lyapunov function in the largest invariant set of $D = \{x \,|\, (x_1, x_3) \in \mathscr{R}^2, |x_2| < \pi\}$. Note however that there is no guarantee that the closed-loop system converges to the origin subject to any initial condition in the set D.

Fig.4.4 shows the trajectories of the path following system of the WMR subject to different initial conditions. It can be seen from the figure that the trajectories converge

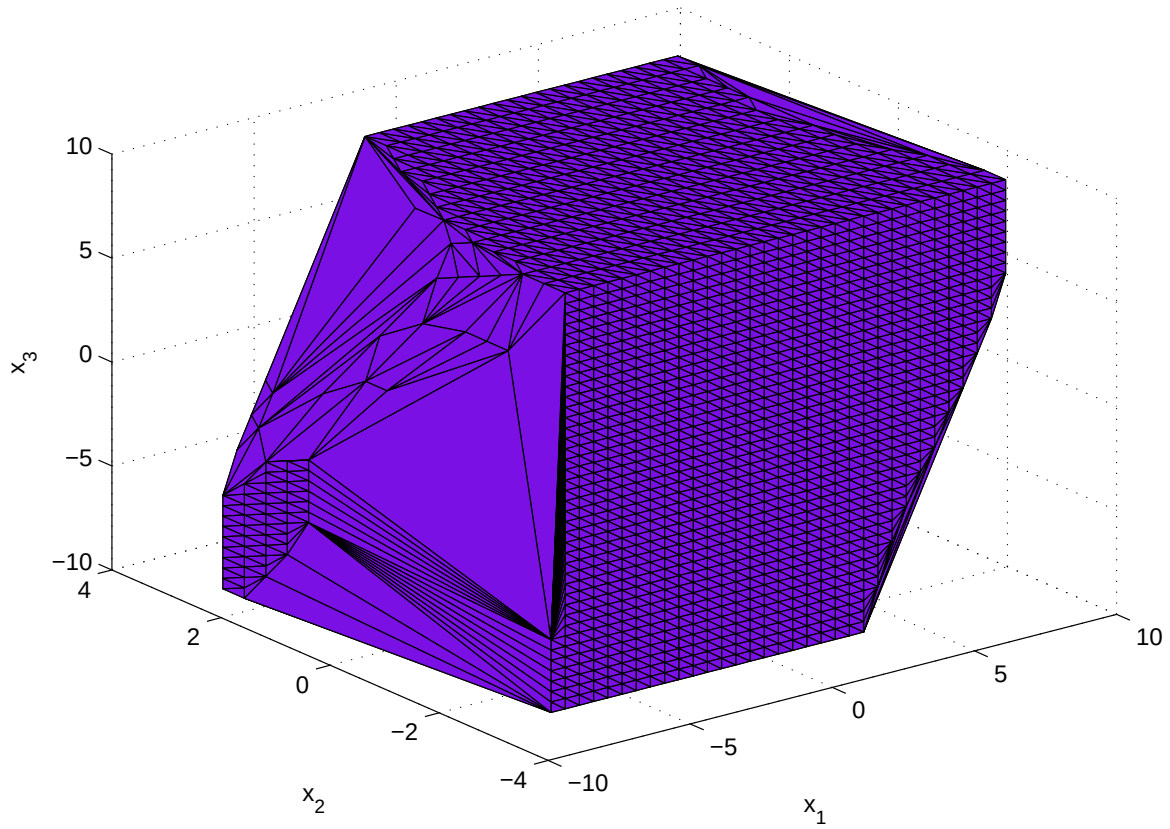

Figure 4.5: Region of Attraction for the given WMR

to the desired path $y = 0$ for the given initial conditions. However, note that since the running cost is not convex, one only has the guarantee that the WMR will follow the straight line and that for a set of initial conditions it will follow the line in the direction corresponding to $\psi = 0$. Moreover, for $\psi \in (-\pi, \pi]$, the direction corresponding to $\psi = \pi$ also makes $sin(\psi) = 0$ leading to a minimum of the running cost. The estimated region of attraction (ROA) for the trajectories of WMR is also shown in *Fig*.4.5. This ROA has been found numerically using simulation with a set of different initial conditions. It shows that for which sets of initial conditions the trajectories of the WMR converge to the origin. Simulation results of the estimated ROA also show that the states x_1 and x_2 are bounded, as seen in *Fig*.4.5, but the state x_3 can be extended to $(-\infty, +\infty)$. The time response of the running cost, the Lyapunov function, the states, and the control input for the system subject to the initial condition $x_0 = \begin{bmatrix} 2 & \pi/2 & 0 \end{bmatrix}^T$ are also shown in *Fig*.4.6.

As seen, both functions $L(x,u)$ and $V(x)$ are positive and converge to zero as $t \to \infty$.

For comparison, an LQR controller is also designed for a linearized model of the given nonlinear system with the weighting matrices in the running cost function corresponding to a second order Taylor series approximation of the running cost. Linearizing the resulting cost function (4.47), we again consider two weighting matrices with and without having cross terms given by, respectively,

$$Q_{LQR_{diag}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad Q_{LQR_{cross}} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 3 \end{bmatrix}$$

and $R = 1$. The $LQR(A,B,Q,R)$ controllers are then obtained as

$$K_{LQR_{diag}} = [1.0000 \quad 3.0550 \quad 2.6665]$$

$$K_{LQR_{cross}} = [1 \quad 3 \quad 3]\,.$$

The trajectories of the path following system of the WMR, subject to the initial condition $x_o = [7 \quad 7\pi/8 \quad 1]^T$, is shown in *Fig*.4.7 for both LQR and HJB-based controllers. It indicates that using both controllers stabilizes the given nonlinear system. However, compared to LQR controllers, the proposed HJB-based controller uses a bit less control effort, although the difference is not significant.

4.5 Experimental Results

In this section a practical application is presented to verify the effectiveness of the proposed methodology experimentally. The experimental setup has been explained in more detail in Section 2.2.1. The kinematics equations of the WMR on the $x-y$ plane are

$$\begin{aligned} \dot{y}(t) &= V\,sin(\psi) \\ \dot{\psi}(t) &= w \end{aligned} \tag{4.48}$$

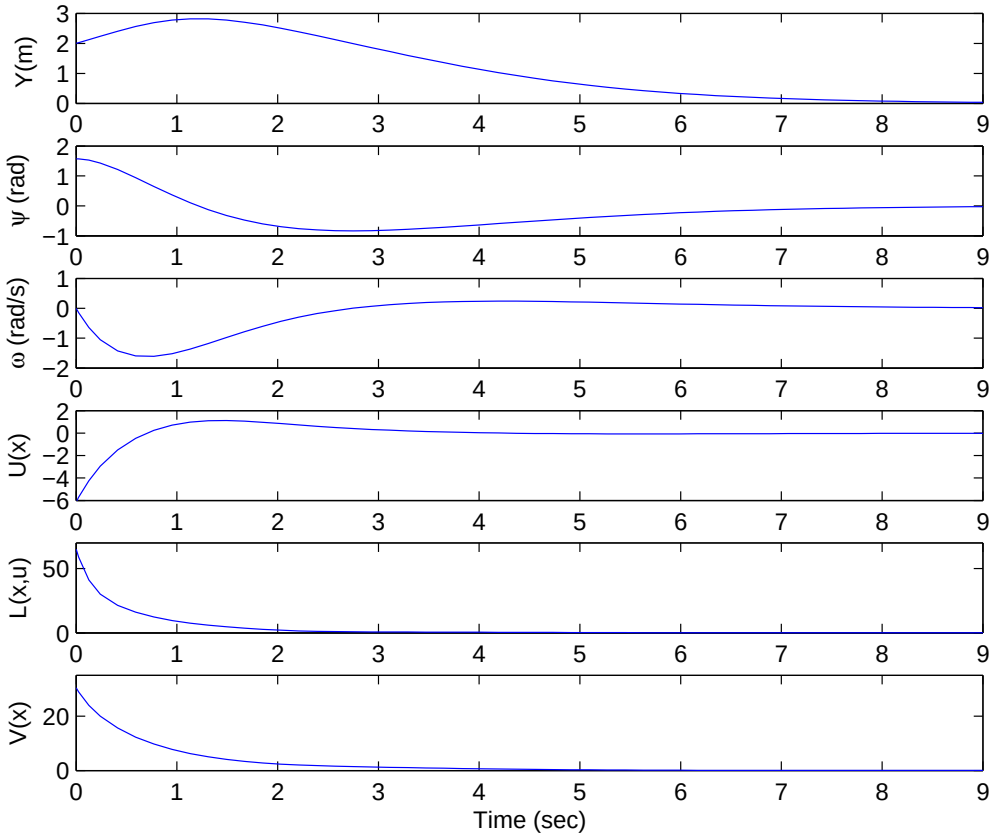

Figure 4.6: Time response of the states, the input, the running cost $L(x,u)$, and the Lyapunov function $V(x)$ for the path following model of the WMR subject to $x_o = [2 \quad \pi/2 \quad 0]^T$

and the dynamic equation of the WMR is given by

$$\dot{w}(t) = cu \tag{4.49}$$

where $c = 0.0066$, and $V = 0.083(m/s)$ is the constant velocity of the WMR. The objective is to find optimal control gains (4.9) for a nonnegative running cost (4.11), which force the WMR to follow the desired path $y = 0$. According to $Fig.2.4$ the states contain the position y, the heading angle ψ, and the angular velocity ω. Since the constants V and b in the WMR model are small values, it experimentally makes sense to select large controller gains K_i compared to example 4.4.3. Choosing

$$q_1 = 300^2\,,\, q_2 = 20^2\,,\, q_3 = 36\,,\, r = 1\,, \tag{4.50}$$

and using (4.4)-(4.5) and (4.28)-(4.29) yields the following optimal control input

$$u = -300y - 20\psi - 909.1\omega - 4.2\sin(\psi)\,. \tag{4.51}$$

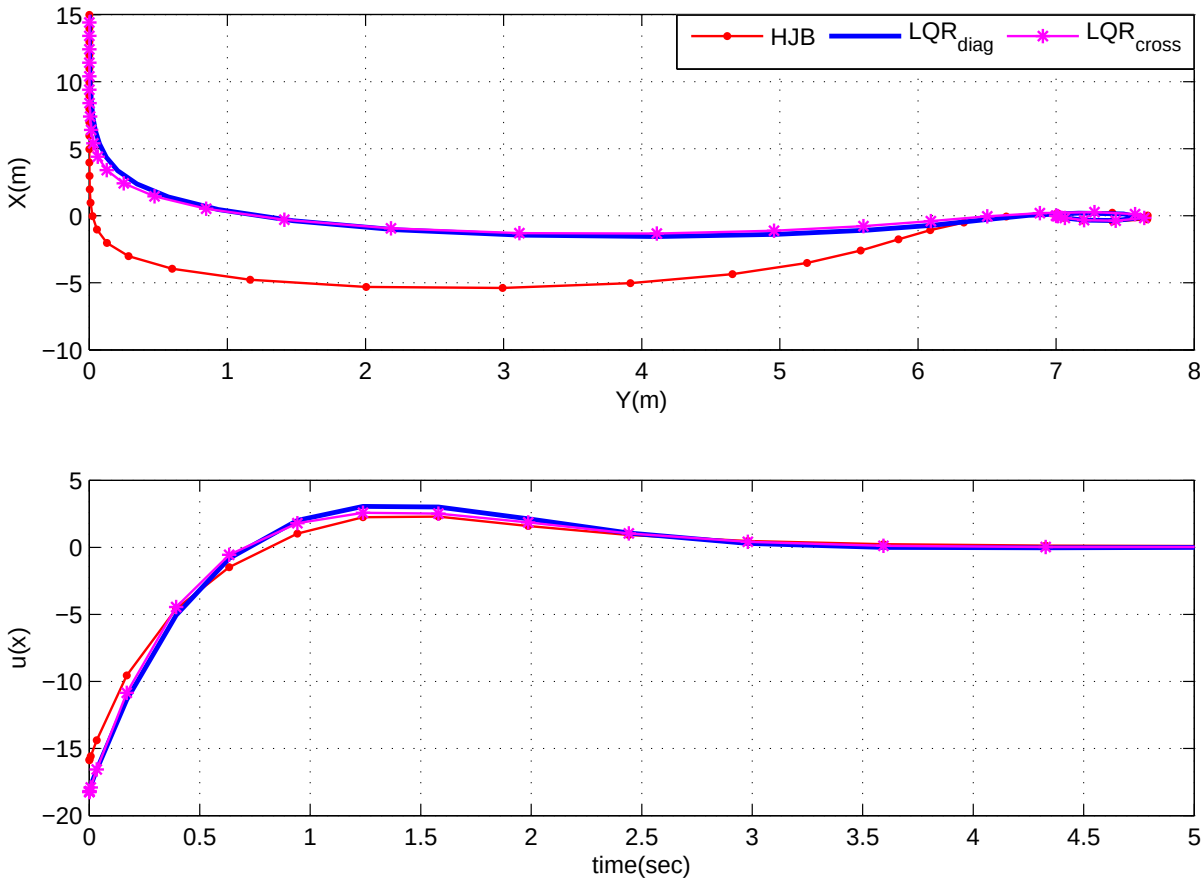

Figure 4.7: Control input and trajectory of the path following system of the WMR for LQR and HJB-based controllers subject to the initial condition $x_o = [7 \quad 7\pi/8 \quad 1]^T$

Fig.4.8 shows the experimental trajectories of the WMR following the line $y = 0$ subject to the following different initial conditions (y_0, ψ_0, w_0)

$$(a) = (0.85, -\pi/3, 0) \quad , \quad (b) = (-0.7, -\pi/2, 0)$$

$$(c) = (0.80, \pi/6, 0) \quad , \quad (d) = (-0.65, \pi/4, 0) \tag{4.52}$$

$$(e) = (0.5, -\pi/2, 0)\,. \tag{4.53}$$

As shown in *Fig*.4.8, the trajectories converge to the desired path $y = 0$. Note however that since the running cost is not convex, one only has the guarantee that the WMR will follow the straight line and that for a set of initial conditions it will follow

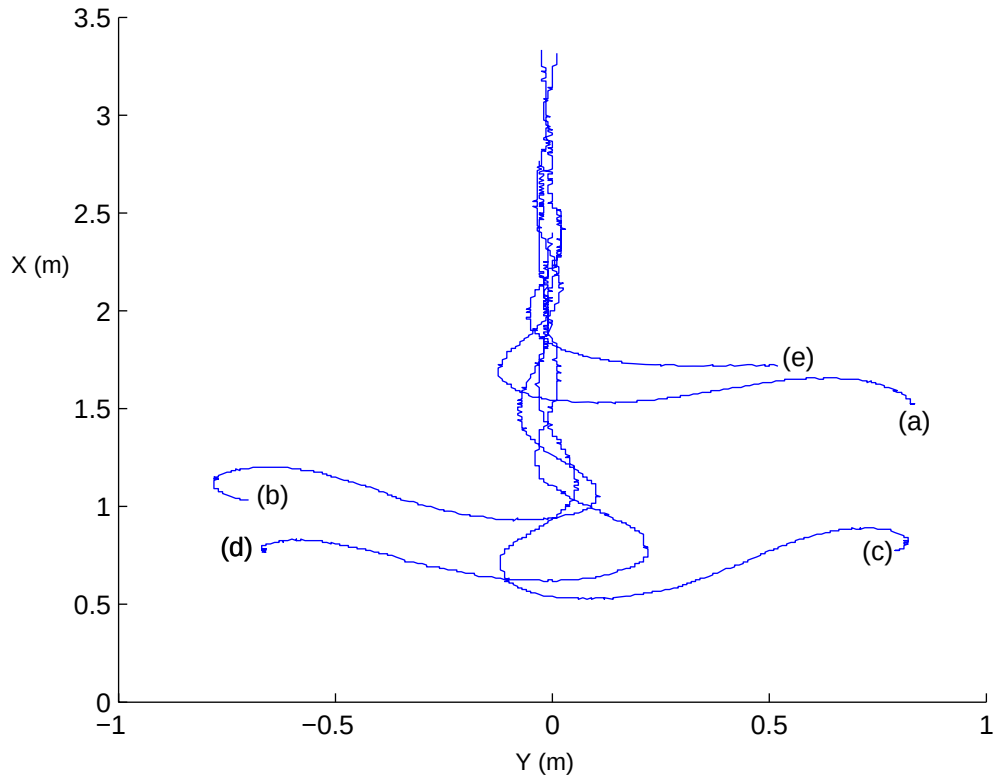

Figure 4.8: Experimental WMR Trajectories

the line in the direction corresponding to $\psi = 0$. The direction corresponding to $\psi = \pi$ also makes $sin(\psi) = 0$ leading to a minimum of the running cost. The results of this paper cannot exclude the possibility of the trajectories converging to this solution. The time response of the experimental control input is shown in *Fig*.4.9. As seen from this figure, the higher the distance to the line and the higher the heading angle of the WMR in its initial condition, the larger is the control input. Also notice that the control input is always bounded, and does not saturate. As discussed in Section (2.2.1), due to hardware and wireless communication limitations, the maximum frequency that the total WMR system can handle is $50Hz$. However, The experimental results in *Fig*.4.8 and *Fig*.4.9 indicate that using this sampling rate of data is quite satisfying to implement the proposed controllers on WMR setup. It is also worthwhile to note that the time trajectories of the WMR and the experimental control inputs are not smooth because of having noise in the experimental setup.

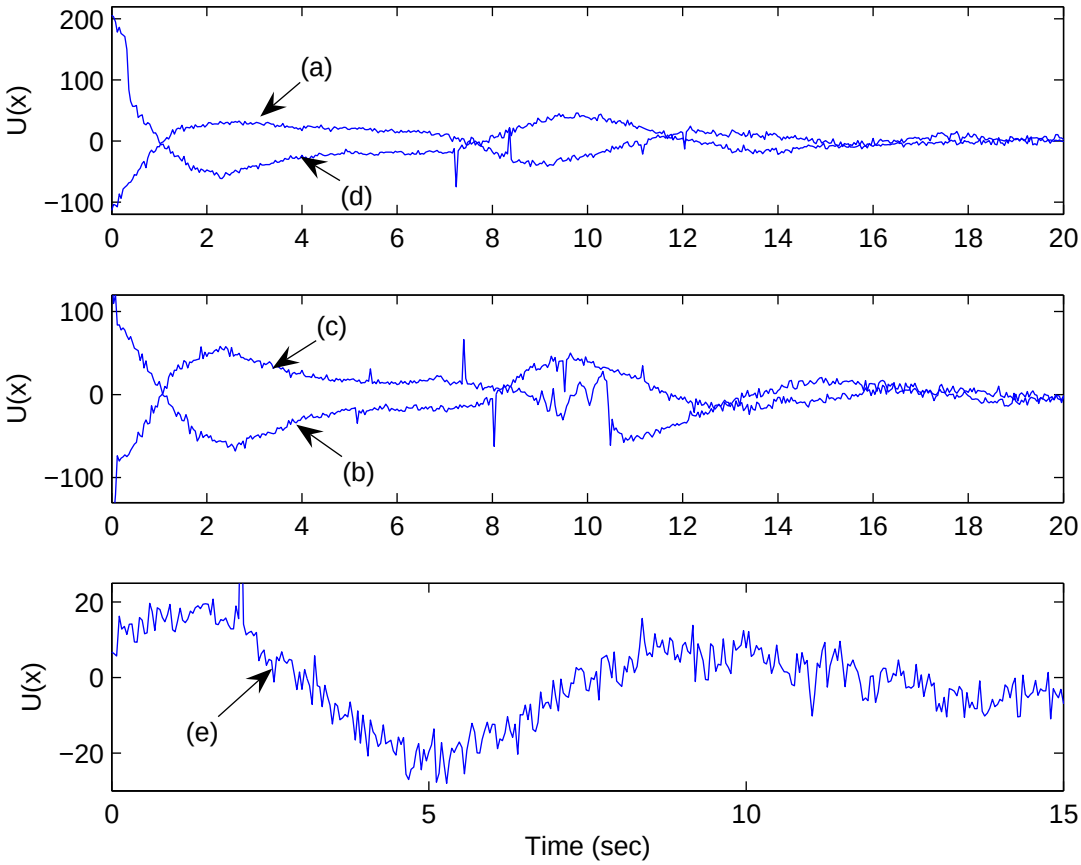

Figure 4.9: Experimental control inputs for the WMR

4.6 Summary

The solution to a class of third order nonlinear optimal control problems has been presented in this chapter using the concept of inverse optimality. The optimal controller and part of the running cost are computed to satisfy the HJB equation. Once the running cost is computed, a local Lyapunov function can be constructed from the value function. Compared to the LQR controller associated with the weight matrices in the running cost function corresponding to a second order Taylor series approximation of the running cost, simulation results show that using the proposed HJB-based optimal controller leads to a smoother responses in states and control effort in some cases. A practical application to a WMR path following problem has also been presented to experimentally verify the effectiveness of the proposed methodology. However, it has been shown that the proposed

method does not always guarantee that the trajectories will converge to a given equilibrium point for all initial conditions. Although the proposed method is restricted to a class of third order optimal control problems, it can potentially be extended to higher order systems assuming that the dynamics are affine and the cost is quadratic in the input.

Chapter 5

Conclusions

In this chapter the main conclusions of this work and the potential future work are stated. In Chapter 2, the kinematics equation of motion for a rigid body has been described using different common representations such as quaternions, Modified Rodrigues Parameter (MRP) and Euler angles. Here we were interested in quaternions rather than MRPs because not only the latter has a geometric singularity, but also the polynomial matrix entries using quaternions are linear while they are nonlinear for the MRP representation. Thus, a quaternion-based attitude model will pose fewer computational challenges. A system identification and an experimental setup of a Wheeled Mobile Robot (WMR) as well as the setup the Quanser helicopter have also been presented in Chapter 2.

In Chapter 3, the main objective was to develop nonsingular rigid-body attitude control laws using a convex formulation, and to implement them in an experimental set up. The attitude recovery problem was first parameterized in terms of quaternions, and then two polynomial controllers using an SoS Lyapunov function and an SoS density function were developed. A quaternion-based polynomial controller using backstepping has also been designed. The simulation results show that the proposed nonlinear controllers guarantee the asymptotic stability of states subject to any initial condition. Moreover,

the numerical simulation as well as experimental results implemented in a Quanser Helicopter verify that the quaternion-based controller stabilizes the closed-loop system in less settling time and with smaller overshoot than the MRP-based controller. A few interesting extensions to the research work in Chapter 3 would be the following:

- Decrease the number of feedback states, specifically in the case of sensor failures,while maintaining the stability
- Adding external disturbances to the system.

In Chapter 4, the solution to a class of third order nonlinear optimal control problems was presented using the concept of inverse optimality. The main contribution of this chapter was to analytically solve the Hamilton-Jacobi-Bellman equation for a class of third order nonlinear optimal control problems for which the dynamics are affine and the cost is quadratic in the input. One special advantage of this work is that the solution is directly obtained for the control input without the computation of a value function first. The optimal controller and part of the running cost are computed to satisfy the HJB equation. Once the running cost is computed, a local Lyapunov function can be constructed from the value function, yielding a proof certificate for stability. Moreover, simulation results show that using the proposed HJB-based optimal controller leads to smooth responses in states and control effort. A practical application to a WMR path following problem was also presented to experimentally verify the effectiveness of the proposed methodology. However, it has been shown that the proposed method does not always guarantee that the trajectories will converge to a given equilibrium point for all initial conditions. Although the proposed method is restricted to a class of third order optimal control problems, it can potentially be extended to higher order systems assuming that the dynamics are affine and the cost is quadratic in the input. Also adding noise and uncertainty parameters can be an interesting extension to this work.

Bibliography

[1] L. Fortuna, G. Muscato, and M.G. Xibilia, ”A Comparison Between HMLP and HRBF for Attitude Control”, *IEEE Transactions on Neural Network*, vol. 12, no. 2, 2001

[2] ”Spacecraft Robust Attitude Tracking Design: PID Control Approach”, *Proceedings of the American Control Conference*, Anchorage, AK, May 8-10, 2002

[3] H. Bang, M. Tahk, and H. Choi, ”Large Angle Attitude Control of Spacecraft with Actuator Saturation”’, *Control Engineering Practice*, Volume 11, Issue 9,pp. 989-997, 2003

[4] B.T Costic, D.W. Dawson, M.S. De Queiroz, and et al, ”Quaternion-based attitude tracking controller without velocity measurement”, *Journal of Guidance, Control, and Dynamics*, vol.24, pp. 1214-1222, 2001

[5] H. Wong, M.S. De Queiroz, and V. Kapila, ”Adaptive tracking control using synthesized velocity form attitude measurements”, *automatica*, vol.37, pp. 947-953, 2001

[6] R. Sharma, and A. Tewari, ”Optimal nonlinear tracking of spacecraft attitude maneuvers”, *IEEE Transactions on Control Systems Technology*, vol.12, pp. 677-682, 2004

[7] H. Leeghim, Y. Choi, and H. Bang, ”Adaptive Attitude Control of Spacecraft Using Neural Networks”, *Acta Astronautica* vol. 64, Issues 7-8, pp. 778-786, 2009

[8] L. Fortuna, G. Muscato, and M.G.Exibilia, ”A comparison between HMLP and HRBF for attitude control”, *IEEE Transaction Neural Networks*, vol.12, pp. 318-328, 2001

[9] A. Satyadas, and K. KrishnaKumar, ”EFM-based controllers for space station attitude control: application and analysis”, *Genetic Algorithm and Soft Computing*, vol.8, pp. 152-171, 1996

[10] P. Guan,X.J. Liu, F. Lara-Rosano, and J.B. Chen, ”Adaptive fuzzy attitude control of satellite based on linearization”, *Proceedings of the American Control Conference*, vol.2, pp. 1091 - 1096, 2004

[11] L. Yingying, and Z. Jun, ”Fuzzy attitude control for flexible satellite during orbit maneuver”, *International Conference on Mechatronics and Automation*, pp. 1239-1243, ICMA 2009

[12] G. Meyer, ”Design and global Analysis of Spacecraft Attitude Control”, tech. rep., NASA Ames Research Center, NASA TR R-361, 1971

[13] ”On the useo fE ulers theorem on rotations for the syntheiis of attitude control systems”, Ames Res. Cen., Moffet Field, CA, NASA Tech. Note NASA TN. D-3643, 1966.

[14] P. Crouch, ”Spacecraft Attitude Control and Stabilization: Applications of Geometric Control Theory to Rigid Body Models”, *IEEE Transactions on Automatic Control*, vol. 29, Issue 4, pp321-331, 1984

[15] J. Wen, and K. Kreutz-Delgado, ”The Attitude Control Problem”, *IEEE Transaction Automatic Control*, vol. 36, pp. 11481162, 1991

[16] W. Dwyer, ”Exact Nonlinear Control of Large Angle Rotational Maneuvers”, *IEEE Transaction on Automatic Control*, Vol. AC-29, No. 9, September 1984

[17] P. Tsiotras, ”Stabilization and Optimality Results for the Attitude Control Problem”, *AIAA Journal of Guidance, Control, and Dynamics*, vol. 19, no. 4, pp. 772-779 1996

[18] S. Tafazoli, and K. Khorasani ”Attitude Recovery of Flexible Spacecraft Using Nonlinear Control”, *IEEE TENCON*, vol. 4, pp. 585-588, 2004

[19] S. Tafazoli, and K. Khorasani, ”Nonlinear Control and Stability Analysis of Spacecraft Attitude Recovery”, *IEEE Transactions on Aerospace and Electronic Systems*, Volume 42, Issue 3, pp 825-845, 2006

[20] H.K. Khalil, ”Nonlinear Systems”, Prentice-Hall Inc., Upper Saddle River, NJ, second edition, 1996

[21] R.A. Freeman, and P.V. Kokotovic, ”Inverse Optimality in Robust Stabilization”, *SIAM Journal on Control and Optimization*, vol. 34, no. 4, pp. 1365-1391, 1996

[22] S. Bi, H. Ji, and S. Chen, ”Robust Attitude Control of Aircraft Based on Partitioned Backstepping”, *IEEE International Conference on Control and Automation*, Christchurch, New Zealand, December 9-11, 2009

[23] R. Kristiansen, P.J. Nicklasson, and J. Tommy Gravdahl, ”Quaternion-Based Backstepping Control of Relative Attitude in a Spacecraft Formation”, *Proceedings of the 45th IEEE Conference on Decision and Control*, Manchester Grand Hyatt Hotel San Diego, CA, USA, December 13-15, 2006

[24] R. Kristiansen, P.J. Nicklasson, and J.T. Gravdahl, ”Satellite Attitude Control by Quaternion-Based Backstepping”, *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 1, January 2009

[25] R. Kristiansen, and P.J. Nicklasson, ”Satellite Attitude Control by Quaternion-Based Backstepping”, *American Control Conference*,June 8-10, USA, 2005

[26] A.P. Aguiar, D.B. Dacic, J.P. Hespanha, and P. Kokotovic, ”Path-following or reference-tracking? An answer based on limits of performance”, *in Processing 5th IFAC/EURON Symposium Intell. Autonomous Vehicle*, Lisbon, Portugal, July 2004

[27] A. Pedro Aguiar, and Joao P. Hespanha, ”Trajectory-Tracking and Path-Following of Underactuated Autonomous Vehicles With Parametric Modeling Uncertainty”, *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, vol. 52, no. 8, August 2007

[28] B. Wie, H. Weiss, and A. Arapostathis, ”Quaternion feedback regulator for spacecraft eigenaxis rotation”, *Journal of Guidance and Control Dynamics*, vol. 12, no. 3, pp. 375380, 1988

[29] S.M. Joshi, A.G. Kelkar, and Y.Wen, ”Robust attitude stabilization of spacecraft using nonlinear quaternion feedback”, *IEEE Transaction on Automatic Control*, vol. 40, no. 10, pp. 18001803, October 1995

[30] H.B. Jensen, and R. Wisniewski, ”Quaternion feedback control for rigid-body spacecraft”, *the Processing of AIAA Guidance and Navigation Control Conference*, Montreal, QC, Canada, 2001.

[31] S.N Singh, and A.Iyer, ”Nonlinear decoupling sliding mode control and attitude control of spacecraft”, *IEEE Transaction on Aerospace and Electronic Systems*, vol. 25, pp. 621-633, 1989

[32] Y.P. Chen, and S.C. Lo, ”Sliding-mode control design for spacecraft attitude tracking maneuvers”, *IEEE Transaction on Aerospace and Electronic Systems*, vol. 29, pp. 1328-1333, 1993

[33] J.L Crassidis, and F.L. Markley, ”Sliding mode control using Modified Rodrigues Parameter”, *Journal of Guidance, Control, and Dynamics*, vol. 19, pp. 1381-1383, 1996

[34] R.D. Robinet, and G.G. Parker, ”Spacecraft Euler parameter tracking of large-angle maneuver via sliding mode control”, *Journal of Guidance, Control, and Dynamics*, vol. 19, 1996

[35] C. Pukdeboon, and A.S.I. Zinober, ”Optimal Sliding Mode Controllers for Attitude Tracking of Spacecraft”, *18th IEEE International Conference on Control Applications, Part of 2009 IEEE Multi-conference on Systems and Control*, Saint Petersburg, Russia, July 8-10, 2009

[36] S. Bharadwaj, M. Osipchuk, K.D. Mease, F.C. Park, ”Geometry and Inverse Optimality in Global Attitude Stabilization”, *Journal of Guidance, Control and Dynamic*, vol. 21, no. 6, pp. 930939, 1998.

[37] T. Lee, N. H. McClamroch, and M. Leok, ”Optimal Attitude Control for a Rigid Body with Symmetry”, *Proceedings of the American Control Conference*, New York City, USA, July 11-13, 2007

[38] W. Luo, Y. Chu, and K. Ling, ”Inverse Optimal Adaptive Control for Attitude Tracking of Spacecraft”, *IEEE Transactions on Automatic Control*, vol. 50, no. 11, November 2005

[39] J.H. Lee, R.W. Diersing, and C.H. Won, ”Satellite Attitude Control Using Statistical Game Theory”, *The Processing of the American Control Conference*, Seattle, Washington, USA, June 11-13, 2008

[40] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, ”Nonlinear and Adaptive Control Design”, New York: Wiley, 1995.

[41] A. Iyer, Sahjendra, and N. Singh, ”Nonlinear adaptive attitude control of satellite using gyrotorquers”, *Proceeding of the 29th Conference on the Decldon and Control*, Honolulu, Hawaii, December 1990

[42] C. Li, and G. Ma, ”Adaptive Backstepping Control for Attitude Tracking of a Spacecraft”, *Symposium on IEEE International Industrial Electronics*,pp. 83-88, 2007

[43] S.N. Singh, and W. Yim, ”Nonlinear Adaptive Spacecraft Attitude Control using Solar Radiation Pressure”, *IEEE Transactions on aerospace and Electronic Systems*, vol. 41, no. 3, July 2005

[44] W. Su, C. E. Souza, and L. Xie, ”H_∞ Control for Asymptotically Stable Nonlinear Systems”, *IEEE Transactions on Automatic Control*, vol. 44, no. 5, May 1999

[45] M. Elgersma, G. Stein, M.R. Jackson, and J. Yeichner, ”Robust controllers for space station momentum management”, *IEEE Control Systems*, vol. 12, no. 5, pp. 1422, 1992.

[46] W. Kang, ”Nonlinear H_∞ control and its applications to rigid spacecraft”, *IEEE Transactions Automatic Control*, vol. 40, pp. 12811285, 1995

[47] W. Luo, Y.C. Chu, and K.V. Ling, ”H_∞ Tracking Control of a Rigid Spacecraft”,*Proceeding of the 2004 American Control Conference*, Boston, Massachusetts, June 30-July 2, 2004

[48] A. papachristodoulou and S. Prajna, ”A Tutorial on Sum of Squares techniques for Systems Analysis”, *Proceedings of American Control Conference* , pp. 2686-2700, 2005.

[49] P.A. Parrilo, ”Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization”, PhD dissertation, Calif. Inst. Technol., Pasadena, CA, 2000

[50] S. Prajna, A. Papachristodoulouand, P. Seiler, and P.A. Parrilo, ”Sum of Squares Optimization Toolbox for MATLAB, User’s Guide”, 2004.

[51] K. Krishnaswamy, G. Papageorgiou, S. Glavaski, and A. Papachristodoulou, ”Analysis Of Aircraft Pitch Axis Stability Augmentation System Using Sum Of Squares Optimization”, *Proceeding of American Control Conference*, USA, June 8-10, 2005

[52] A. Ataei-Esfahani, ”Robust Control Design for Aircrafts: Case Studies Using Semidefinite Programming Approach”, *A Thesis in Mechanical Engineering*, Pennsylvania State University, 2006.

[53] A. Ataei-Esfahani, and Q, Wang, ”Robust Nonlinear Control Design for a Hypersonic Aircraft Using Sum of Squares method”, *Proceedings of the ASME 2010 Dynamic Systems and Control Conference (DSCC)*, September 12-15, Cambridge, Massachusetts, USA, 2010

[54] N. Gollu and L. Rodrigues, ”Control of Large Angle Attitude Maneuvers for Rigid Bodies Using Sum of Squares”, *IEEE proceedings American Control Conference* , pp. 3156-3161, 2007.

[55] H. Ichiharay, and M. Kawata, ”Attitude Control of Acrobot by Gain Scheduling Control Based on Sum of Squares”, *2010 Proceeding of American Control Conference*, Baltimore, MD, USA, June 30-July 02, 2010

[56] S. Prajna, A. Papachristodoulou and F. Wu, ”Nonlinear Control Synthesis by Sum of Squares Optimization: A Lyapunov-based Approach”, *Proceeding of the ASCC*, pp. 157-165, 2004.

[57] S. Prajna, A. Papachristodoulou and A. Rantzer, ”Nonlinear Control Synthesis by Convex Optimization”, *IEEE Transactions on Automatic Control*, Vol. 49, no.2, pp. 310-314, 2004.

[58] V.A. Chobotov, ”Spacecraft Attitude Dynamics and Control”, Krieger Publishing Company, pp. 1-32, Malabar, Florida, 1991.

[59] C. Grubin, ”Derivation of the Quaternion Scheme via the Euler angle and Axis”, Journal Spacecraft and Rockets, vol. 7, no. 10, October, 1970

[60] B. Wie, ”Space Vehicle Dynamics and Control”, AIAA Education Series, Second Edition, Blacksburg, Virginia, 2008

[61] N. Gollu, ”Development of Novel Satellite Attitude Determination and Control Algorithms based on Telemetry Data from an Earth Satellite”, PHD Thesis, Concordia University, 2008

[62] Pichwick, World Wide Web, `http://fr.wikipedia.org/wiki/Angles_d'Euler`

[63] T. Wen, S. Seereeram, and D.S. Bayard, ”Nonlinear Predictive Control applied to Spacecraft Attitude Control”, *Proceedings of the American Control Conference*, 1997

[64] P.C. Hughes, ”Spacecraft Attitude Dynamics”, John Wiley & Sons, Inc., TL1050.H84, pp. 93-100 1986.

[65] M.J. Sidi, ”Spacecraft Dynamics and Control: A Practical Engineering Approach”, Cambridge University Press, pp. 318-327, 1997.

[66] A.J. TurnerAn, ”Open-Source, Extensible Spacecraft Simulation And Modeling Environment Framework”, MSc Thesis, Aerospace Engineering Department, Virginia Polytechnic Institute and State University, USA, 2003

[67] A. Rantzer, ”A dual to Lyapunovs stability theorem”, System Control Letter, vol. 42, no. 3, pp. 161168, 2001

[68] R. Endress, ”Modeling and Control of Two Degrees of Freedom Quanser Helicopter”, MSc Thesis, Department of Mechanical and Industrial Engineering, Montreal, Quebec, Canada, 2008

[69] T. Fukao, ”Inverse optimal tracking control of a nonholonomic mobile robot”, *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, pp. 14751480, 2004

[70] R. Kalman, ”When is a linear control system optimal”, *ASME Transactions, Journal of Basic Engineering*, vol. 86, pp. 51-60, 1964

[71] F. Thau, ”On the inverse optimum control problem for a class of nonlinear autonomous systems”, *IEEE Transactions on Automatic Control*, vol. 12, no. 6, pp. 674681, 1967

[72] M. Krstic, and P. Tsiotras, ”Inverse optimal stabilization of a rigid spacecraft”, *IEEE Transactions on Automatic Control*, vol. 44, Issue 5, pp. 1042-1049, 1999

[73] J. Guojun, ”Inverse Optimal Stabilization of a Class of Nonlinear Systems”, *Proceedings of the 26th Chinese Control Conference*, pp. 226-230, 2007

[74] M. Kanazawa, S. Nakaura, and M. Sampei, ”Inverse optimal control problem for bilinear systems: Application to the inverted pendulum with horizontal and vertical movement”, *IEEE Conference on Decision and Control*, pp. 2260-2267, 2009

[75] K. Hongkeun, B. Juhoon, S. Hyungbo, and J.H. Seo, ”Locally optimal and globally inverse optimal controller for multi-input nonlinear systems”, *IEEE American Control Conference*,pp. 4486-4491, 2008

[76] X. Cai, and Z. Han, ”Inverse optimal control of nonlinear systems with structural uncertainty”, *IEEE Proceedings of Control Theory and Applications*, vol. 152, pp. 79-83, China, January 2005

[77] T. Fukao, T. Kanzawa, and K. Osuka, ”Inverse optimal tracking control of an aerial blimp robot”, *Proceedings of the 5th International Workshop on Robot Motion and Control*, pp. 193-198, June 23-25, 2005

[78] K. Dupree, P.M. Patre, M. Johnson, and W.E. Dixon, ”Inverse optimal adaptive control of a nonlinear Euler-Lagrange system, part I: Full state feedback”, *Proceedings of the 48th IEEE Conference on*

Decision and Control, held jointly with the 28th Chinese Control Conference, pp. 321-326, 15-18 December, China, 2009

[79] D.E. Kirk, ”Optimal Control Theory: An Introduction”, *Dover Publications Inc.*, pp. 86-96, 1998

[80] L. Rodrigues, ”An Inverse Optimality Method to Solve a Class of Second Order Optimal Control Problems”, submitted to *A*utomatica.

[81] D. Egan, ”The Emergence of ZigBee in Building Automation and Industrial Controls”, *IEE Computing & Control Engineering*, vol. 16, no. 2, pp. 14-19, 2005

[82] D. V. Gadre, ”Programming and Customizing the AVR Microcontroller”, *mcgraw-Hill*, ISBN 0-07-134666-X, 2001

[83] R.C. Gonzalez, and R.E. Woods, ”Digital Image Processing”, *Prentice Hall*, SBN 0201180758. , 2002

[84] O. Egeland, and J. T. Gravdahl, ”Modeling and Simulation for Automatic Control”, Trondheim, Norway: Marine Cybernetics, 2002.