

https://dspace.org/

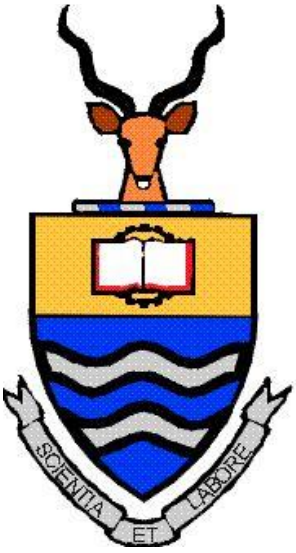
Robust nonlinear trajectory tracking and control of quadrotor UAV

Chingozha, Tinashe

2014-09-03
http://hdl.handle.net/10539/15373

Downloaded from a DSpace repository. https://dspace.org/

ROBUST NONLINEAR TRAJECTORY TRACKING AND CONTROL OF QUADROTOR UAV



Tinashe Chingozha

Faculty of Engineering

University of the Witwatersrand

A Dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg in fulfilment of the requirements of the degree of Master of Science.

Johannesburg, 2014

To my mother, Tafadzwa, Saul and Keilah

Acknowledgements

I would like to extend my heartfelt and sincere gratitude to my supervisor Dr Otis Nyandoro, for his support and assistance during this work. In a special way I would like to thank Dr Nyandoro for going out of his way in sourcing the funding that made this work possible and for being more than a work colleague

Secondly I would like to thank my family for their unwaivering support and encouragement. My mother, my brothers Saul and Tafadzwa to you I am eternally grateful.

Abstract

Unmanned Aerial Vehicles(UAVs) provide a versatile platform for the automation of a wide variety of tasks such as powerline inspection, border interdiction, search and rescue e.t.c. The success of these UAV platforms relies heavily on the development of control algorithms that can cope with the harsh and uncertain environments in which the UAVs will operate in. This dissertation focuses on the development of robust trajectory tracking control algorithms for a quadrotor UAV platform. Robustness in this context refers to the ability of the controller to guarantee system performance in the presence of uncertainties such as unknown system parameters or some other unmodeled effects. By exploiting the strict feedback form of the quadrotor dynamics a backstepping based control strategy for the system which comprises of two sub-controllers namely a translational controller and an attitude controller is developed. For the translational controller of the UAV a novel robust bounded controller is developed. This novel controller is developed by combining A.R Teel's nonlinear saturated controller with sliding mode techniques to achieve bounded error tracking in the presence of disturbances while at the same time ensuring bounded control which captures the limited nature of the UAV's thrust actuators. Additionally conditions on the controller parameters are identified which ensure that the UAV does not overturn during flight. The controller for the vehicle attitude is based on a modified backstepping method. Conventional backstepping control is formulated under the implicit assumption of a perfectly known system, thus in instances where uncertainty exists the performance of conventional backstepping deteriorates. To improve on the robustness of conventional backstepping control, methods of combining it with

adaptive and/or sliding mode techniques are considered. Adaptive backstepping control is robust against parametric uncertainty however its performance deteriorates in the presence of disturbances. An adaptive backstepping controller with nonlinear damping is proposed as a solution to this problem, Lyapunov based analysis shows that this controller achieves bounded error tracking in the presence of parametric and non-parametric uncertainty. A second modification of the backstepping method that is considered involves combining sliding mode control with conventional backstepping control. Sliding backstepping control is a powerful control method in that it is able to achieve asymptotic tracking in the presence of uncertainty. However this is only achieved if the upper bounds of the uncertainty are known *a priori*, this requirement is very difficult to meet in practice. Thus an adaptive sliding backstepping controller is proposed which removes the requirement of *a priori* knowledge of the upper bounds. In conclusion the key features of this work are a novel robust bounded translational controller, an adaptive backstepping attitude controller with nonlinear damping and an adaptive sliding backstepping attitude controller with guaranteed asymptotic tracking. Thus a comprehensive robust trajectory tracking controller for a quadrotor UAV is developed.

Contents

Contents **v**

List of Figures **ix**

1 Introduction **1**
1.1 Background 2
1.2 Research Motivation 3
1.3 Contributions 4
1.4 Dissertation Outline 5
1.5 Conclusion 5

2 Literature Review **6**
2.1 Chapter Overview 6
2.2 UAV Modelling 6
2.3 Quadrotor Trajectory Tracking Methodologies 8
2.3.1 Linear Methods 8
2.3.2 Nonlinear Methods 9
2.4 Conclusion 12

3 Quadrotor UAV Modelling **13**
3.1 Chapter Overview 13
3.2 Preliminaries 13
3.3 Rotation Matrices 14
3.3.1 The $SO(3)$ Group 15
3.3.2 Quaternion Representation of Rotations 17

3.3.3 Euler Angle Representation of Rotations 19
3.4 Quadrotor UAV Kinematics 20
3.4.1 Translational Kinematics 20
3.4.2 Rotational Kinematics 21
3.5 Quadrotor UAV Dynamics 22
3.5.1 Translational Dynamics 22
3.5.2 Rotational Dynamics 23
3.6 Full Quadrotor Model 24
3.7 External Wrench Model 25
3.8 Simplified Model 28
3.9 Conclusion 29

4 Control Strategy Overview **30**
4.1 Chapter Overview 30
4.2 Quadcopter Dynamics 30
4.3 Problem Statement 31
4.4 Control Strategy 31
4.5 Conclusion 32

5 Translational Control **34**
5.1 Chapter Overview 34
5.2 Translational Dynamics 35
5.3 Stabilization of Multiple Integrator System with Bounded Controls 36
5.3.1 Teel's Method 36
5.3.2 Modified Teel's Method 41
5.4 Translational Controller 50
5.5 Conclusion 53

6 Attitude Control 1 **54**
6.1 Chapter Overview 54
6.2 Backstepping Control 54
6.3 Backstepping with Uncertainty 65
6.3.1 Adaptive Backstepping Control 65
6.4 Nonlinear Damping 74

6.5 Attitude Control 75
6.5.1 Orientation Controller 76
6.5.2 Yaw Control 79
6.5.3 Simulation Results 80
6.6 Conclusion 81

7 Attitude Control 2 **87**
7.1 Chapter Overview 87
7.2 Sliding Mode Control 87
7.3 Adaptive Sliding Mode Control 93
7.3.1 Method 1 96
7.3.2 Method 2 98
7.4 Sliding Backstepping Control 99
7.5 Attitude Control 106
7.5.1 Orientation Controller 107
7.5.2 Yaw Controller 108
7.5.3 Simulation Results 109
7.6 Conclusion 111

8 Quadrotor UAV Trajectory Control **116**
8.1 Chapter Overview 116
8.2 Simulation Results 116
8.2.1 Hovering Maneuver 117
8.2.2 Figure 8 Maneuver 124
8.3 Conclusion 130

9 Conclusions and Recommendations **131**
9.1 Conclusions 131
9.2 Recommendations 133

Appendix A **135**
.1 Vehicle Frame 135
.2 Vehicle-1 Frame(F^{v1}) 136
.3 Vehicle-2 Frame(F^{v2}) 136

.4 Body Frame F^b 136

References **138**

List of Figures

1.1 UAV systems 2

3.1 Inertial frame(superscript I) and Body fixed frame(superscript B) 15
3.2 Transformation of $\mathbf{p}$ to $\mathbf{p}'$ by matrix $\mathbf{T}$ 16
3.3 Matrix $\mathbf{T}$ as a coordinate system rotation 17
3.4 Z-Y-X Euler angles 19
3.5 Full quadrotor model[1] 25
3.6 Quadrotor UAV schematic showing propeller rotation directions . 26
3.7 Quadrotor simplified model 28

4.1 Attitude and Translational subsystem interconnection that shows the "partial" feedback form 32
4.2 Backstepping based control strategy 33

5.1 Saturation functions σ_1 and σ_2 37
5.2 System response with Teel controller. Blue is for initial condition ($x_1 = 0.5$, $x_2 = 0.2$, $x_3 = 0.1$) and green ($x_1 = 0.5$, $x_2 = 0.5$, $x_3 = 0.5$) 42
5.3 State trajectory of system with a sinusoidal disturbance input of amplitude 0.1 43
5.4 State trajectory of system without disturbance. Blue is for initial condition ($x_1 = 0.5$, $x_2 = 0.2$, $x_3 = 0.1$) and green ($x_1 = 0.5$, $x_2 = 0.5$, $x_3 = 0.5$) 49
5.5 System response with zero initial conditions and sinusoidal disturbance amplitude 0.1 50

5.6 System response with initial conditions ($x_1 = 0.5$, $x_2 = 0.5$, $x_3 = 0.5$) and sinusoidal disturbance amplitude 0.1 51

6.1 Block diagram of the system 55
6.2 Backstepping example simulation results. Blue = ($\lambda = 0.5$), green = ($\lambda = 1$), red = ($\lambda = 1.5$), black = ($\lambda = 2$) 63
6.3 Integral Backstepping example simulation results. Blue =($\lambda_1 = 0.7, \lambda_2 = 1, \lambda_3 = 0.8$, green = ($\lambda_1 = 0.5, \lambda_2 = 0.6, \lambda_3 = 0.4$), red = ($\lambda_1 = 1.2, \lambda_2 = 1.6, \lambda_3 = 1$). 64
6.4 Adaptive Backstepping example simulation results 73
6.5 Reference signal $\phi_d = 0.2sin(\pi t)$, $\theta_d = 0.2sin(0.5\pi t)$, $\delta_\phi(t) = 0.1sin(2\pi t)$ 82
6.6 Reference signal $\theta_d = 0.2sin(0.5\pi t)$, $\psi_d = 0$, $\delta_\theta(t) = 0.1sin(5\pi t)$, $\delta_\psi(t) = 0.1sin(7\pi t)$ 83
6.7 $\hat{I}_\theta$, $\hat{I}_\psi$ and τ_ϕ plots 84
6.8 τ_θ and τ_ϕ plots 85
6.9 τ_ψ plot 86

7.1 Sliding mode controlled system phase portrait[45] 91
7.2 Chattering in practical sliding mode control[45] 91
7.3 Signum function approximation, blue = $sgn(.)$, green = $tanh(.)$, red = $\frac{\pi}{2}arctan(.)$ 92
7.4 Signum function approximation: Green = $tanh(10x)$ and blue = $tanh(500x)$ 94
7.5 x plot: Green = $tanh(10x)$ and blue = $tanh(500x)$ 94
7.6 $\dot{x}$ plot: Green = $tanh(10x)$ and blue = $tanh(500x)$ 95
7.7 u plot: Green = $tanh(10x)$ and blue = $tanh(500x)$ 95
7.8 x plot: Blue = method 1 adaptation, green = method 2 adaptation 100
7.9 $\dot{x}$ plot: Blue = method 1 adaptation, green = method 2 adaptation 100
7.10 $\hat{K}$ plot: Blue = method 1 adaptation, green = method 2 adaptation 101
7.11 u plot: Blue = method 1 adaptation, green = method 2 adaptation 101
7.12 $x_1(t)$ plot: Blue = u_1, green = u_2 105
7.13 $x_2(t)$ plot: Blue = u_1, green = u_2 106
7.14 Control input plot: Blue = u_1, green = u_2 107

7.15 ϕ plot: Green = reference signal, blue = without disturbance, red = with disturbance 110
7.16 θ plot: Green = reference signal, blue = without disturbance, red = with disturbance 111
7.17 ψ plot: Green = reference signal, blue = without disturbance, red = with disturbance 112
7.18 $\hat{K}_\phi$ plot: Blue = without disturbance, green = with disturbance . 113
7.19 $\hat{K}_\theta$ plot: Blue = without disturbance, green = with disturbance . 113
7.20 $\hat{K}_\psi$ plot: Blue = without disturbance, green = with disturbance . 114
7.21 τ_ϕ plot: Blue = without disturbance, green = with disturbance . . 114
7.22 τ_θ plot: Blue = without disturbance, green = with disturbance . . 115
7.23 τ_ψ plot: Blue = without disturbance, green = with disturbance . . 115

8.1 Hover maneuver x plot: Green = without disturbance, red = with disturbance 118
8.2 Hover maneuver y plot: Green = without disturbance, red = with disturbance 118
8.3 Hover maneuver z plot: Blue = reference trajectory, green = without disturbance, red = with disturbance 119
8.4 Hover maneuver ϕ plot: Blue = without disturbance, green = with disturbance 119
8.5 Hover maneuver θ plot: Blue = without disturbance, green = with disturbance 120
8.6 Hover maneuver ψ plot: Blue = without disturbance, green = with disturbance 120
8.7 Hover maneuver τ_ϕ plot: Blue = without disturbance, green = with disturbance 121
8.8 Hover maneuver τ_θ plot: Blue = without disturbance, green = with disturbance 121
8.9 Hover maneuver τ_ψ plot: Blue = without disturbance, green = with disturbance 122
8.10 Hover maneuver Thrust plot: Blue = without disturbance, green = with disturbance 122

8.11 Hover maneuver 3D plot: Green = without disturbance, red = with disturbance 123
8.12 Figure 8 maneuver x plot: Green = reference trajectory, blue = without disturbance, red = with disturbance 124
8.13 Figure 8 maneuver y plot: Green = reference trajectory, blue = without disturbance, red = with disturbance 125
8.14 Figure 8 maneuver z plot: Green = reference trajectory, blue = without disturbance, red = with disturbance 125
8.15 Figure 8 maneuver ϕ plot: Blue = without disturbance, green = with disturbance 126
8.16 Figure 8 maneuver θ plot: Blue = without disturbance, green = with disturbance 126
8.17 Figure 8 maneuver ψ plot: Blue = without disturbance, green = with disturbance 127
8.18 Figure 8 maneuver τ_ϕ plot: Blue = without disturbance, green = with disturbance 127
8.19 Figure 8 maneuver τ_θ plot: Blue = without disturbance, green = with disturbance 128
8.20 Figure 8 maneuver τ_ψ plot: Blue = without disturbance, green = with disturbance 128
8.21 Figure 8 maneuver Thrust plot: Blue = without disturbance, green = with disturbance 129
8.22 Figure 8 maneuver 3D plot: Green = reference trajectory, blue = without disturbance, red = with disturbance 129

1 Vehicle and body fixed frame 135

Chapter 1

Introduction

Man's interest in unmanned flying vehicles is as old as manned flight. Unmanned aerial vehicles(UAVs) can be traced back to 1916 when American inventor Elmer Sperry of Sperry Gyroscope Company successfully implemented a stabilising control system for the Curtiss Flying boat [2],[3]. During the First World War much effort was put into development of "flying bombs" however the efforts of the time were met with limited success. Success in this front was finally achieved by the Germans with the deployment of the V-1 "Buzz-Bomb" during the course of the second World War. The V-1 bomb was the first successful cruise missile and was a precursor to the modern UAV.[3]

The age of the modern UAV began in the late 1940s with the development of the BQM-34A "Firebee" drone which was mainly used as a target drone for missile testing[4]. The shooting down of the American U2 spy plane over Russia in 1960 spurred military strategists to consider using UAVs for surveillance missions. However it was not until the Vietnam conflict that UAVs were used enmasse in combat situations. During the Vietnam war UAVs flew over 3 400 combat missions in which they were mostly used for intelligence gathering[3]. After the Vietnam war UAV technology continued to advance evidenced by the successful development of the Israeli Pioneer UAV. By the mid 1990s UAV technology had matured with the development of Northrop Grumman's Global Hawk (see Figure 1.1a) and General Atomics' Predator drone(Figure 1.1b), two UAVs that have become synonymous with the term "drone".

From this brief history it is evident that the development of UAV technology



(a) Northrop Grumman Global Hawk[5]



(b) General Atomics Predator[6]

Figure 1.1: UAV systems

has been largely driven by military needs. According to a comprehensive UAV application survey [7] civilian applications accounted for only 3% of the total UAV revenue in 2000. However developments in micro-electro mechanical systems(MEMS) and IC miniaturisation has driven UAV development costs down making UAVs economical for civilian use. UAVs possess a lot of potential in the civilian market with possible uses ranging from border interdiction, search and rescue missions, wild fire suppression, industrial plant inspection e.t.c. Most of these applications are performed in cluttered and constrained enviroments which differ from the open terrain of the battlefield. This difference in operating enviroments means that the fixed wing UAVs that are very successful in the battlefield enviroment cannot be used in most of these civilian applications. As such rotary wing UAVs have become the mainstay of civilian UAV applications as they are highly manueverable and their high thrust to weight ratio means that smaller UAVs can be used.

1.1 Background

According to the survey paper by Chen et al[8] an autonomous UAV should be capable of not just automated operation but should also be able to detect and respond to unanticipated changes in the environment. Such a system will comprise numerous subsystems that perform tasks of trajectory planning, fault detection

and toleration, trajectory tracking and learning. The scope of this research is restricted to only the trajectory tracking system of the autonomous UAV with the investigation being of a theoretical nature. Experimental verification of the developed algorithm was not undertaken as the design and construction of a UAV testbed covers extensive knowledge areas and would have made the scope of the work too big to be finished within the time frame of the research.

Over the past decade a lot of research has been done in the control engineering community with regards to quadrotor trajectory tracking control. Trajectory tracking control of quadrotors presents a challenge due to the nature of the dynamics of the quadrotor. Quadrotor dynamics are highly coupled, nonlinear and the quadrotor is underactuated making the controller design a significant challenge. Despite these complexities a number of successes have been achieved, these include Stanford University's STARMAC project [9], ETH Zurich's OS4 [10] project and University of Pennsylvania's GRASP UAV testbed[11]. It is the aim of this research to provide a starting point for the possible development of a UAV testbed at the University of the Witwatersrand

1.2 Research Motivation

Quadrotor UAVs are perfectly suited to operations in cluttered indoor environments because of their light weight and high manoeuverability. Successful operation in such environments requires near perfect trajectory tracking as any deviations from the planned trajectory might result in collisions with obstacles. To achieve these requirements the trajectory tracking controller must tackle the complexities that arise due to nonlinearities and uncertainties.

In designing control algorithms the algorithm is only as good as the system model on which it is based. Thus in order to design controllers for the quadrotor UAV one has to consider nonlinear models which can fully describe the UAV's dynamics in all the flight regimes. This presents a significant challenge in the controller design as the nonlinear model is described by highly nonlinear and highly coupled differential equations. Another difficulty that arises is due to the underactuated nature of the quadrotor dynamics, underactuation refers to the fact that the quadrotor has more degrees of freedom than it has control inputs.

The behavior of a system is greatly influenced by the values of the parameters that appear in the differential equations that describe it. Ideally the values of these parameters should be known before hand. Unfortunately for the quadrotor this is not the case, the irregular shape of the quadrotor body means that the values of the vehicle's inertia cannot be exactly determined thus introducing uncertainties in the vehicle model. Another source of uncertainty is due to wind disturbances which act on the quadrotor in flight. Thus the controller that will be designed needs to guarantee the system's performance against both of these unknowns, such ability of the controller is referred to as controller robustness.

1.3 Contributions

The nature of the work covered in this dissertation is of a theoretical nature and as such the contribution of the work has a theoretical emphasis. The main focus of the contributions of this research is in the development of novel control methods for the quadrotor trajectory tracking problem. Additionally the major contributions of this work are of a general nature and as such their applicability is not limited to the quadrotor UAV system. The key findings of this work are summarised as follows:

1. Development of a novel robust bounded control method for multiple integrator systems with matched uncertainty which is presented in section 5.3.2. This bounded controller is used to control the translational dynamics of the quadrotor UAV.

2. Conditions for the selection of controller parameters for the translational controller so as to ensure that the UAV does not overturn. This is necessary as the Euler angles rotation parameterisation breaks down in such a manoeuvre.

3. In chapter 6 an adaptive sliding backstepping control algorithm is developed for strict feedback systems with matched uncertainty. To combine sliding mode control and backstepping control two methods are investigated, the first one uses the conventional Lyapunov based approach while the second

approach relies on the selection of an appropriate sliding manifold. The second approach is shown to result in a simpler controller than the conventional Lyapunov based approach

4. To achieve adaptation of the sliding gain in the adaptive sliding backstepping controller in section 7.3 a unique adaptation law is used. This differs from the conventional adaptation rules in that it gives the minimum gain estimate that will sufficiently counteract the effect of the uncertainties.

5. A novel adaptive sliding backstepping control scheme is developed using the results highlighted in points 4 and 5. This controller is used to develop an attitude controller for the quadrotor UAV

1.4 Dissertation Outline

This dissertation is organised as follows. Chapter 2 gives the literature review in which the problem is contextualised within the framework of existing works. The mathematical model for the quadrotor UAV is developed in chapter 3. A high level description of the proposed control solution is given in chapter 4, the following three chapters give a detailed account of the proposed solution. Chapter 5 focusses on the translational control while chapters 6 and 7 focus on the attitude control problem. Simulation results of the proposed controller are presented in chapter 8 with concluding remarks and reccomendations being given in chapter 9.

1.5 Conclusion

UAVs possess great potential to drastically change how our modern world operates. Realisation of this potential has driven the interest of numerous researchers all over the world into the vast field of UAVs. Huge strides have been thus far made in the area of UAVs but however a lot still remains to be done in order to achieve fully autonomous UAVs. The work that is presented in this dissertation is focused on studying the trajectory tracking problem for the quadrotor UAV. Special emphasis being placed on robustness of the control algorithms.

Chapter 2

Literature Review

2.1 Chapter Overview

This chapter provides a broad summary of some of the methods that have been employed in the literature for the Modelling and control of quadrotor UAVs. The literature review also serves to motivate for the research areas that this work focuses on. Some of the issues discussed in this chapter are treated in depth in the relevant chapters of the dissertation.

2.2 UAV Modelling

A quadrotor UAV can be viewed as a rigid body in 3 dimensional space and thus possesses 6 degrees of freedom, three translational and three rotational degrees of freedom. Description of the translational position of the UAV is a trivial task however the description of the vehicle orientation is fairly complicated and has implications on the derived model. A number of methods exist for describing the orientation of a general rigid body in space such as quaternions, Euler angles, axis-angle, Cayley-Klein parameters and Euler-Rodrigues parameters[12],[13], [14], [15]. Quaternions and Euler angles are the most used in aeronautical applications and thus shall be the focus of this discussion. Euler angles comprise three angles yaw, pitch and roll which are used to describe the orientation of a rigid body. One of the advantages of Euler angles is that they are intuitive and it is easy to visualise

rotations described in this way. On the other hand Euler representation has a disadvantage in that it exhibits singularities, this phenomenon is called "gimbal lock" which restricts the trajectories that the quadrotor can track[16]. Thus if Euler angles are used to model the UAV the control algorithms that are designed using that model are not capable of executing aggressive aerobatic manoeuvres. Quaternions represent a rotation by a four element vector, this method does not suffer from the singularity issues of Euler angles and thus provides a globally valid way of representing UAV orientation. Additionally in comparison to Euler angles quaternions are computationally efficient as they use a 4 element vector to describe rotations compared to a 3×3 matrix in the case of Euler angles. Despite these advantages quaternions are used less in modelling quadrotors because they are conceptually challenging to understand and are not very intuitive. In this work Euler angles are used for representing the quadrotor UAV's orientation.Due to the limitations of Euler angles the controller is designed in such a way that gimbal lock is avoided. More detailed discussion of quaternions and Euler angles is contained in chapter 3 of this dissertation.

In deriving the equations of motion of the quadrotor it is common to assume that the vehicle is a rigid body. The motion of a rigid body in 3-D space is governed by the Newton-Euler equations[17]. Using these equations the dynamic model of the quadrotor can be derived as shown in the books on helicopter flight theory [18],[19]. The models developed in [18] and [19] are quite comprehensive as they take into consideration the complicated aerodynamic phenomena at play during flight such as blade flapping. Such a model however is very complicated which results in complicated control algorithms which are computationally intensive and difficult to implement on DSPs. A common simplification is to assume that the flight of the quadrotor will be in the low velocity regime in which the aerodynamic effects are negligible[20]. Another simplification is to lump all the aerodynamics forces and torques and consider them as disturbance inputs this is the approach that is taken in this dissertation. The nonlinear Newton-Euler equations can be further simplified by assuming that the time derivatives of the Euler angles and the body frame angular velocities are equal[21]. Taking this reasoning a step further it can be assumed that the Coriolis terms that appear in the nonlinear model are negligible[22],[23] which results in the orientation dynamics

being decoupled into three forced double integrator systems. This simplification is very popular in the literature but however it results in a very limited operation area in the flight envelope as the assumptions are only satisfied if the quadrotor flies at low speeds and small angles of attack. As such these assumptions are not employed in this work.

2.3 Quadrotor Trajectory Tracking Methodologies

2.3.1 Linear Methods

Control methods for UAV trajectory tracking can be grouped into two general categories, linear and nonlinear methods. In this brief survey linear control methods refers to control methods that make use of linear system models. Linearisation of the nonlinear quadrotor model is achieved using the Jacobian method, in this method an operating point is first identified and the system is then linearised about that point. For the quadrotor three major operating regions exist, these are take-off/landing, hover and forward flight. Michael et al[11] implemented a PID controller for the quadrotor system linearised about hover point, this controller was successfully tested on an experimental platform. In their work Michael et al[11] showed that even though the model they use is linearised about hover the PID control algorithm is even capable of executing waypoint trajectory tracking. Similar results are shown by the work presented in [24] and [25]. The work presented in [25] and [24] goes a step further as they showed that the PID controller is superior to the LQR based controller. Despite all these advantages because the PID controller is based on a linear model which neglects important nonlinear components of the dynamics it shows poor disturbance rejection qualities as is shown in [26]. To try and improve robustness of the LQR controller it is proposed in [27] to add feedforward terms and a robust filter to the controller, this attempt showed marked improvement in the controller. A linear method that has been shown to have good robustness qualities is the H_∞ loop shaping method. H_∞ loop shaping methods are used in [28] and [29] to successfully control a 3-DOF

quadrotor bench model. In [30] and [31] full model controllers are designed using H_∞ techniques with satisfactory results. The superior performance of H_∞ methods above other linear methods[28] is primarily due to fact that with the H_∞ framework it is possible to incorporate the nonlinearity as disturbances rather than totally ignoring them as in other linear control methods. From this brief survey of linear UAV control methods it might seem that linear methods are sufficient for the task of UAV trajectory tracking. Using models derived by making near hover assumptions full authority trajectory tracking controllers have been developed using such simple algorithms as PID control[11]. However it should be noted that in order to achieve waypoint tracking using a near hover model based controller it is necessary to restrict the UAV to low velocities and small angles of attack so as to meet the near hover assumption. The result is that such controllers do not fully take advantage of the strengths of quadrotor platforms such as their agility and high thrust to weight ration. It is this fundamental limitation of linear methods which makes nonlinear methods preferable in this application.

2.3.2 Nonlinear Methods

Numerous types of nonlinear control methods have been developed and applied to the UAV trajectory tracking problem. As such the review done in this section does not claim to be exhaustive. For a more comprehensive survey of the field the interested reader is directed to the survey paper [32].

One of the areas that has been extensively studied within UAV control deals with the development of constrained controls for the UAV trajectory tracking problems. Constrained controls are advantageous as they capture the limited nature of the UAV actuators. Constrained optimal control is one method that has been investigated in this regard. In [33] an optimal time controller is developed in which a constraint is placed on the rotational speed of the rotors. Alexis et al[33] do show via simulations the feasibility of this approach however optimal control is known to be very computationally intensive which is a huge drawback. Another approach to constrained control makes use of the theory of nested saturated control algorithms for multiple integrator systems. A major result in this field is the nonlinear feedback controller developed by A.R. Teel[34]. This result was

exploited in [1] to design a translational controller for a conventional helicopter, in [35] and [36] this method is further extended to the quadrotor helicopter. Teel's control is elegant and simple but however has two major drawbacks, firstly it is inflexible as there is no way to shape the transient response secondly the method has poor disturbance rejection. A number of extensions to Teel's control have been put forward to try and improve its response [37],[38],[39]. The improvements in [39] were incorporated in the quadrotor trajectory controller developed by Hably et al[40]. In chapter 5 a novel extension to Teel's control is developed with the aim of improving the robustness characteristics of the controller.

In developing trajectory tracking control algorithms for UAVs robustness considerations are very important because of the uncertainty introduced in the system by unknown parameters such as inertia and unmodeled dynamics such as air drag, wind loading e.t.c. Sliding mode control is a powerful control design technique which is capable of handling such uncertainties as are present in the UAV system. Bouabdallah et al[41] developed a sliding mode controller for the quadrotor's orientation, in [42] and [43] the sliding mode technique is applied to the whole UAV system. The work presented in [42] develops the sliding mode controller so as to achieve fault tolerant performance of the UAV. Xu et al[44] develop a new design method for sliding mode control for underactuated systems which is applied successfully to the quadrotor UAV system. It is a well known fact that controls synthesized using the sliding mode method exhibit high frequency chattering[45], in [41] this chattering is reported as causing a deterioration of the system performance. Another drawback of sliding mode control is the fact that the designer is required to know the upper bounds of the uncertainties *a priori*, this requirement is very difficult to meet in practice if for example the uncertainty is wind gust disturbances. The chattering in the sliding mode control can be eliminated by using the boundary region method however this tends to compromise performance since the sliding manifold is not reached[45]. The relaxation of the a priori upper bound requirement presents an area of active research with promising results coming from the adaptive sliding mode approach. In [46] and [47] fuzzy logic is used to come up with adaptation laws for the sliding gain however this approach fails to guarantee the tracking performance. Huang et al[48] devised a gain adaptation law in which the rate of growth of the gain

estimate is proportional to the sliding manifold error, a major drawback of this work is that it leads to an over estimation of the gain leading to unnecessarily large controls. An alternative method which limits the sliding gain is proposed in [49] however this method requires *a priori* knowledge of the uncertainty upper bounds. Chapter 7 of this dissertation deals with these issues in a more in depth and rigorous manner.

Backstepping control is a Lyapunov based recursive control design method developed in the early 1990s first appearing in [50] and [51]. This method is applicable to the class of systems that are in strict-feedback form of which the quadrotor system is a member. Another characteristic of this method which makes it suited for quadrotor control is the ease with which controls for underactuated systems can be developed within the backstepping scheme. As such numerous researchers have developed backstepping controllers for the quadrotor UAV, in [52] a backstepping controller is developed for the altitude control subsystem. Full model controllers based on the basic backstepping method for UAV trajectory tracking were developed in [53],[41],[54]. To improve the transient performance PID control can be incorporated into the backstepping scheme. Bouabdallah et al[55] implemented an Integral Backstepping based trajectory tracking control to reduce steady state errors in [56], the effectiveness of the full PID backstepping controller is demonstrated. Amidst the numerous successes of backstepping control in the quadrotor trajectory tracking problem the method suffers from one major drawback. In the formulation of the backstepping method it is implicitly assumed that an exact model is available, thus the method performs poorly when uncertainties are present in the model. To improve the robustness of backstepping one avenue that has been investigated involves coupling backstepping with adaptive elements to cater for parametric uncertainties. This approach has been extremely successful with the first adaptive backstepping algorithm being developed for the matched case in [57]. Extension to the extended matching case was done in [58] but the proposed method had the disadvantage of over-parameterisation of the unknown term. A solution to this overparameterisation was presented in Kristic et al[59] with the development of the tuning function method. The theoretical advances in adaptive backstepping control have been successfully implemented for the quadrotor UAV[24],[60]. One weakness of the adaptive backstepping method

is that the performance of the adaptive element deteriorates in the presence of disturbances[61]. Another avenue of achieving robust backstepping involves the amalgamation of backstepping control and sliding mode control[62],[63]. The robustification of backstepping control is further addressed in chapters 6 and 7 of this dissertationOne of the areas that has been

2.4 Conclusion

From the brief review of the literature provided it is evident that nonlinear control methods provide a more comprehensive solution to the quadrotor UAV control problem. As such this approach will be adopted in this work. Special emphasis will be placed on robust control methods, some of which have already been highlighted in this chapter.

Chapter 3

Quadrotor UAV Modelling

3.1 Chapter Overview

It is well known that the effectiveness of any control system is dependent on how accurate the controlled system is modeled. Developing a model that incorporates all the system dynamics is very difficult and even if it were possible the controls synthesized using such a model will likely be too complicated to be implementable. Therefore in developing a model one has to balance between model accuracy and simplicity. The aim of this chapter is the development of a quadrotor UAV model to be used for the controller synthesis. This is achieved by assuming the quadrotor to be a rigid body and derivingthe vehicle dynamics from the Newton-Euler equations. To simplify the model aerodynamic effects such as drag are not explicitly modeled but are treated as disturbances to the system however the small angle approximation which is common in the literature is discarded. The result is the development of a more comprehensive model which is tractable at the same time.

3.2 Preliminaries

In deriving the equations of motion of the quadrotor UAV two assumptions are made:

1. The Earth is assumed to be flat and non-rotating which is valid for the quadcopter as the distances it moves are relatively small in comparison to the Earth's radius.

2. The quadrotor UAV is assumed to be a rigid body.

Assumption 1 allows any Earth fixed frame of reference to be considered as an inertial frame of reference. Due to the second assumption dynamic effects caused by the elastic distortions of the vehicle structure under aerodynamic loading can be neglected.

To model the motion of the quadrotor UAV it is necessary to use different frames of references, specifically an Earth fixed inertial frame and a body fixed frame is needed. This is required because:

- desired trajectories are given in the inertial frame
- sensors measure quantities in the body frame
- actuators exert forces and torques in the body frame

The body fixed frame corresponds to a coordinate system whose origin coincides with the centre of mass of the quadrotor UAV. For both the inertial and body fixed frame the North-East-Down(N.E.D) coordinate system is used, figure 3.1 shows the relationship of the inertial and body fixed frame.

3.3 Rotation Matrices

Now that the two coordinate systems that are needed for deriving the mathematical model of the quadrotor UAV have been identified, a method to relate the two frames needs to be established. It should be clear that the body fixed frame can be viewed as a linear transformation of the inertial frame where this transformation is composed of a translation and a rotation. Representing the translation part of the transformation is trivial as this is just the position vector of the quadrotor UAV in the inertial frame. This section is going to focus on the representation of the rotational part of the transformation.

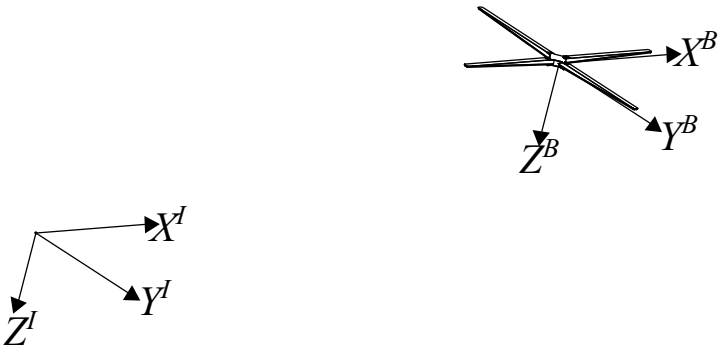


Figure 3.1: Inertial frame(superscript I) and Body fixed frame(superscript B)

Consider the 2D vector $\mathbf{p} = [1\ 1]$, if $\mathbf{p}$ is multiplied by a 2×2 matrix $\mathbf{T}$:

$$\mathbf{T} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

The resultant vector is $\mathbf{p}' = [2\ 0]$ which is as is shown in figure 3.2. The matrix $\mathbf{T}$ can be interpreted as a transformation of the vector $\mathbf{p}$ which comprises of a rotation and a scaling, the scaling factor is equal to the square root of the determinant of matrix $\mathbf{T}$. Thus a pure rotation can be represented by a square matrix with determinant equal to 1. Matrices with determinant of -1 correspond to a rotation and a reflection and as such are not used. The vector $\mathbf{p}'$ can also be regarded as the vector $\mathbf{p}$ observed from the co-ordinate system (O$\mathbf{x}'\mathbf{y}'$) which is the result of transforming the coordinating system (O$\mathbf{xy}$) by the matrix $\mathbf{T}$ as is shown in figure 3.3 below.

3.3.1 The $SO\,(3)$ Group

It has been shown that rotations in n-dimensional space can be represented by $n \times n$ matrices with determinant of 1.This family of matrices is termed the special orthogonal n group $SO(n)$, in fact $SO(n)$ is a Lie group i.e a manifold that

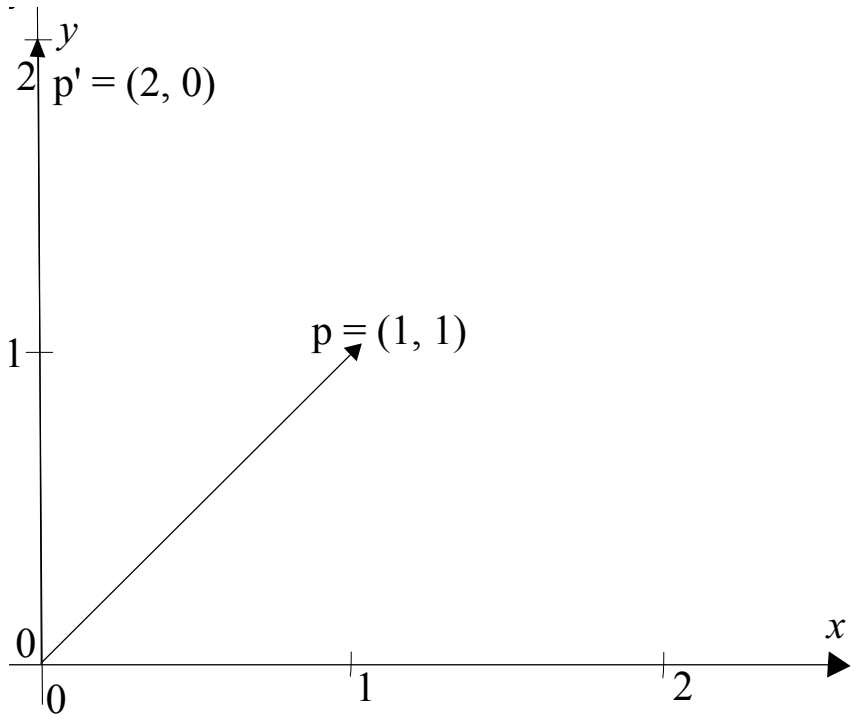


Figure 3.2: Transformation of $\mathbf{p}$ to $\mathbf{p}'$ by matrix $\mathbf{T}$

possesses a group structure[64]. In this work the concern is with rotations in 3-D therefore the interest will be in elements of the $SO(3)$ group. Elements of $SO(3)$ have the following properties[1] :

- special i.e have a determinant of 1
- orthogonal i.e if $R \in SO\left(3\right), RR^T = 1$ where T denotes matrix transpose
- each column(row) of R is a unit vector

As stated earlier $SO(3)$ forms a manifold. To "visualize" the structure of this manifold requires first Euler's Rotation Theorem[65].

Euler Rotation Theorem. *Any finite rotation can be achieved by a single rotation about some axis*

Consider a body that is rotated about the origin, according to Euler's theorem such a rotation can be described by a unit vector for the rotation axis and a scalar

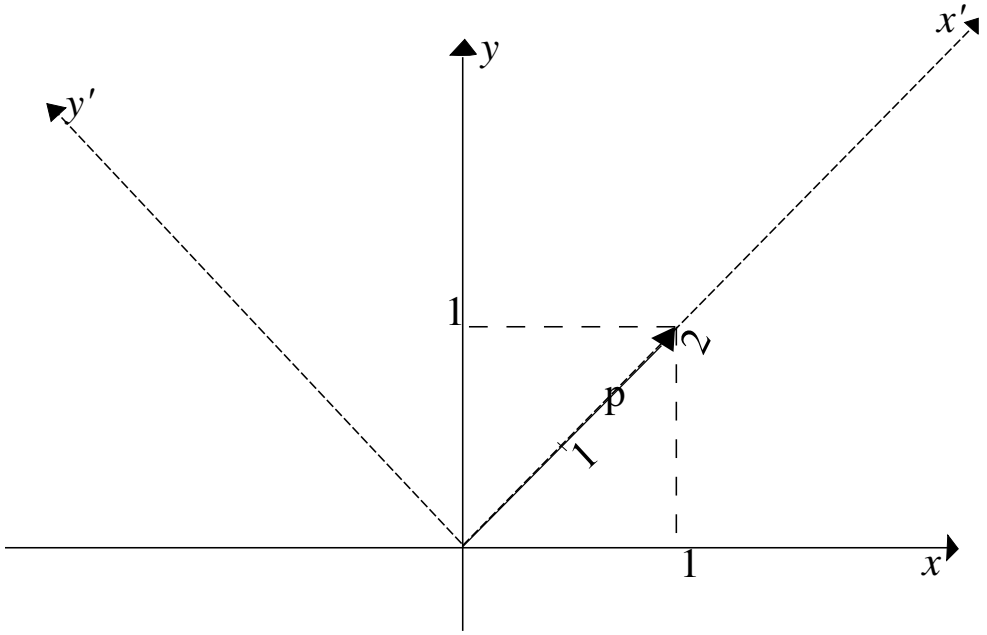


Figure 3.3: Matrix **T** as a coordinate system rotation

quantity for the rotation angle. Thus each such rotation can be associated to a line through the origin where each line corresponds to an axis of rotation and the length of each line is equal to the angle of rotation. The set of lines through the origin of $\mathbb{R}^3$ form a manifold structure called the real projective plane $\mathbb{R}P^2$, this manifold is topologically similar(i.e homeomorphic) to the 4 dimensional sphere (S^3) with antipodal points identified[66]. Identification of antipodal points means that points on S^3 that are opposite each other will correspond to the same rotation. Thus $SO(3)$ can be viewed as being topologically similar to the 4 dimensional sphere S^3.

3.3.2 Quaternion Representation of Rotations

Quaternions were devised by the 19th century Irish mathematician William Rowan Hamilton. The subject of quaternions is very vast so this section will present only a brief summary. A quaternion $\mathbf{q} \in \mathbb{H}$ is a vector with a scalar component(q_0)

and a vector component ($\mathbf{q}_{1:3}$) :

$$\mathbf{q} = [q_0 \; q_1 \; q_2 \; q_3]^T = \begin{bmatrix} q_0 \\ \mathbf{q}_{1:3} \end{bmatrix} \tag{3.1}$$

Consider the unit quaternion $\mathbf{q}'$, the components of $\mathbf{q}'$ will satisfy the equations

$$q_0'^2 + q_1'^2 + q_2'^2 + q_3'^2 = 1 \tag{3.2}$$

This is the equation of a 4 dimensional sphere. From the previous section it was established that $SO(3)$ is homeomorphic to the 4 dimensional with antipodal points identified. As such the quaternions can be used to represent rotations in 3 dimensions. However since the unit quaternions describe the unit sphere they provide a double cover for the $SO(3)$ manifold which means that two quaternions($\mathbf{q}$ and -$\mathbf{q}$) will represent the same rotation. Quaternions have a number of advantages for attitude representations:

1. Quaternions provide a globally valid way of representing rotations
2. Quaternions are computationally efficient since only 4 parameters are used rather than the 9 in the rotation matrix representation

Despite these advantages quaternions do present some challenges in implementation, some of them are:

1. since quaternions are a double cover of the $SO(3)$ manifold they do not provide a one-to-one relationship with rotations
2. quaternions are not intuitive as one cannot easily visualize rotations in quaternion form
3. the requirement of unit magnitude on the quaternions presents computational challenges especially if one is to perform an optimization as the unit norm requirements translates to a quadratic constraint on the optimization

3.3.3 Euler Angle Representation of Rotations

Euler angles are a way of representing the orientation of a reference frame (such as the body frame) relative to another fixed frame(such as the inertial frame) by considering a sequence of rotations about the axes of the frame. There are twelve possible sequences that one can use however in this work the focus is on the Z-Y-X sequence. The three Z-Y-X/Tait-Bryan angles are the yaw, pitch and roll (ψ, θ, ϕ) angles, figure 3.4 shows the angles with respect to the quadrotor UAV body.

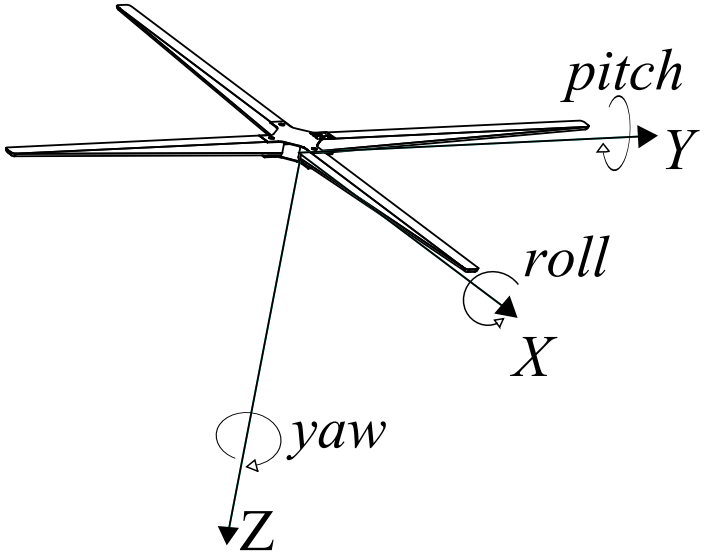


Figure 3.4: Z-Y-X Euler angles

Given the three Z-Y-X Euler angles one can express the rotation matrix that transforms the body frame to the inertial frame as [67] :

$$R = \begin{bmatrix} cos\psi cos\theta & cos\psi sin\theta sin\phi - sin\psi cos\phi & sin\psi sin\phi + cos\psi sin\theta cos\phi \\ sin\psi cos\theta & cos\psi cos\phi + sin\psi sin\theta sin\phi & sin\psi sin\theta cos\phi - sin\phi cos\psi \\ -sin\theta & cos\theta sin\phi & cos\theta cos\phi \end{bmatrix} \tag{3.3}$$

A derivation of this rotation matrix is given in Appendix A. The Euler parameterisation of $SO\,(3)$(which is similar to the 4 dimensional sphere) is very similar

to the latitude-longitude coordinates of the 3-dimensional sphere. The latitude-longitude system uses 2 angles to represent a point on the 3-dimensional sphere while the Euler parameterisation uses 3 angles to represent a point on the 4 dimensional sphere. The latitude-longitude system breaks down at the poles where the longitude becomes degenerate and there is no unique value for the poles, a similar thing also happens with the Euler angles. If the pitch $= 90^0$ one degree of freedom will be lost this phenomenon is commonly known as gimbal lock. To clearly see the cause of gimbal lock consider the inverse map of the Z-Y-X Euler parameterisation.

$$E_{ZYX} : \mathbb{R}^3 \rightarrow SO\,(3) \quad E_{XYZ} = [r_{ij}]\, i, j = 1, 2, 3 \tag{3.4}$$

$$E_{ZYX}^{-1} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} atan2\,(r_{23}, r_{33}) \\ -arcsin\,(r_{13}) \\ atan2\,(r_{12}, r_{11}) \end{bmatrix} \tag{3.5}$$

where $atan2(y, x)$ is the four quadrant inverse tangent function. From equation 3.5 it can be seen that the inverse mapping has a singularity at $\phi = 90^0$ which is the source of the gimbal lock. In this research the Euler angle representation will be used despite the singularity problems since Euler angles are more intuitive and they are commonly used in the aeronautical community.

3.4 Quadrotor UAV Kinematics

3.4.1 Translational Kinematics

The vehicle position and velocity in the inertial frame are represented by the vectors:

$$\mathbf{p}^I = [p_x\ p_y\ p_z]^T \tag{3.6}$$

$$\mathbf{v}^I = [v_x\ v_y\ v_z]^T \tag{3.7}$$

It should be clear then that the translational kinematics are governed by the trivial equation:

$$\dot{\mathbf{p}^I} = \mathbf{v}^I \tag{3.8}$$

3.4.2 Rotational Kinematics

The vehicle attitude and its angular rates are represented by the vectors $\mathbf{\Theta}$ and ω^B such that:

$$\mathbf{\Theta} = [\phi\ \theta\ \psi]^T \tag{3.9}$$

$$\omega^B = [p\ q\ r]^T \tag{3.10}$$

It should be noted that the angular rate vector ω is measured in the body fixed frame. To see how the Euler angles and the angular rates relate consider the infinitesimal rotations $d\phi$, $d\theta$ and $d\psi$, this rotation can be represented by a vector:

$$\mathbf{n} = d\phi\mathbf{i}^B + d\theta\mathbf{j}^{v2} + d\psi\mathbf{k}^{v1} \tag{3.11}$$

Note that the components of the vector $\hat{\mathbf{n}}$ are measured from different frames see Appendix A for more detail. The angular velocity can then be expressed as [1]:

$$\omega = \frac{d\mathbf{n}}{dt} = \dot{\phi}\mathbf{i}^B + \dot{\theta}\mathbf{j}^{v2} + \dot{\psi}\mathbf{k}^{v1} \tag{3.12}$$

$$\omega = p\mathbf{i}^B + q\mathbf{j}^B + r\mathbf{k}^B \tag{3.13}$$

where the superscripts B, $v1$ and $v2$ denote the vehicle frame, vehicle 1 and 2 frames respectively. For more details on this notation see Appendix A. Using

equations 1-3 from Appendix A gives:

$$\begin{bmatrix} \text{p} \\ \text{q} \\ \text{r} \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_{v2}^{B}(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{v2}^{B}(\phi) R_{v1}^{v_2}(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{3.14}$$

$$\begin{bmatrix} \text{p} \\ \text{q} \\ \text{r} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3.15}$$

This equation can be expressed in a more convenient form as :

$$\dot{\mathbf{\Theta}} = \mathbf{\Psi}(\mathbf{\Theta})\,\omega^{B} \tag{3.16}$$

where:

$$\Psi(\Theta) = \begin{bmatrix} 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & cos\phi & -sin\phi \\ 0 & \frac{sin\phi}{cos\theta} & \frac{cos\phi}{cos\theta} \end{bmatrix} \tag{3.17}$$

In general the rotation matrix is time varying, the derivative of the rotation matrix is given by[68]:

$$\dot{R} = R\hat{\omega}^{B} \tag{3.18}$$

where $\hat{\omega}^{B}$ is the skew symmetric matrix of the vector ω^{B}. For $\omega^{B} = [p\ q\ r]^{T}$ the skew symmetric matrix is defined as:

$$\hat{\omega}^{B} = \begin{bmatrix} 0 & \text{-r} & \text{q} \\ \text{r} & 0 & \text{-p} \\ \text{-q} & \text{p} & 0 \end{bmatrix} \tag{3.19}$$

3.5 Quadrotor UAV Dynamics

3.5.1 Translational Dynamics

According to Newton's 2^{nd} law the acceleration of a body is proportional to the applied force. For the quadrotor the forces are applied in the body frame while the vehicle's position and velocity are measured in the inertial frame. Thus the

forces need to be transformed via the rotation matrix $R_B^I(\phi, \theta, \psi)$ to the inertial frame. Thus the translational dynamics become:

$$\dot{\mathbf{v}}^I = \frac{1}{m} R_B^I(\phi, \theta, \psi)\, \mathbf{f}^B + g\mathbf{e}_3 \tag{3.20}$$

where $\mathbf{f}^B = \begin{bmatrix} f_x^B & f_y^B & f_z^B \end{bmatrix}^T$ is the force vector exerted on the quadrotor in the body frame, $\mathbf{e}_3 = [0\ 0\ 1]$ and g is gravity. Note that gravity is positive because in the North-East-Down(N.E.D) frame vertical downwards is positive.

3.5.2 Rotational Dynamics

For rotational motion Newton's 2^{nd} law of motion states that the rate of change of angular momentum is equal to the net torque acting on the body. This can be expressed as :

$$\frac{d\mathbf{H}^B}{dt_I} = \tau \tag{3.21}$$

The angular momentum $\mathbf{H}^B = \mathbf{I}\omega^B$ with $\mathbf{I}$ being the 3×3 inertia matrix given by :

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{3.22}$$

The inertial matrix elements are given by :

$$\begin{aligned} I_{xx} &= \int (y^2 + z^2) dm \\ I_{yy} &= \int (x^2 + z^2) dm \\ I_{zz} &= \int (x^2 + y^2) dm \end{aligned} \tag{3.23}$$

$$\begin{aligned} I_{xy} = I_{yx} &= \int xy dm \\ I_{xz} = I_{zx} &= \int xz dm \\ I_{yz} = I_{zy} &= \int yz dm \end{aligned} \tag{3.24}$$

Assuming the quadrotor to be perfectly symmetrical about all of its three axis the inertia cross terms become $I_{xy} = I_{xz} = I_{yz} = 0$ and the inertia matrix

becomes.

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.25}$$

In equation 3.21 a body frame vector is differentiated in the inertia frame. Using the equation of Coriolis[17]:

$$\frac{d\mathbf{H}^B}{dt^I} = \frac{d\mathbf{H}^B}{dt^B} + \omega_{b/i}^B \times \mathbf{H}^B \tag{3.26}$$

Applying this to equation 3.21 the rotational dynamics become:

$$\mathbf{I}\dot{\omega}^B = -\omega^B \times \left(\mathbf{I}\omega^B\right) + \tau^B \tag{3.27}$$

where $\tau^B = \begin{bmatrix} \tau_\phi^B & \tau_\theta^B & \tau_\psi^B \end{bmatrix}^T$ is the torque acting on the quadrotor expressed in the vehicle frame. In expanded form the rotational dynamics are given by the equations:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{J_{yy}-J_{zz}}{J_{xx}} qr \\ \frac{J_{zz}-J_{xx}}{J_{yy}} pr \\ \frac{J_{xx}-J_{yy}}{J_{zz}} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{J_{xx}}\tau_\phi \\ \frac{1}{J_{yy}}\tau_\theta \\ \frac{1}{J_{zz}}\tau_\psi \end{bmatrix} \tag{3.28}$$

3.6 Full Quadrotor Model

Combining the quadrotor equations for the kinematics and dynamics, the complete quadrotor model is given by:

$$\begin{aligned} \dot{\mathbf{p}}^I &= \mathbf{v}^I && (3.29) \\ \dot{\mathbf{v}}^I &= \frac{1}{m} R_B^I(\phi,\theta,\psi)\,\mathbf{f}^B + g\mathbf{e}_3 && (3.30) \\ \dot{R} &= R\hat{\omega}^B && (3.31) \\ \mathbf{I}\dot{\omega}^B &= -\omega^B \times \left(\mathbf{I}\omega^B\right) + \tau^B && (3.32) \end{aligned}$$

The pose of the quadrotor is defined by the pair $\left(\mathbf{p}^I, R\right) \in SE(3)$ where $SE(3)$ is the Special Euclidean group. The orientation dynamics of the quadrotor are

equally described by both equations 3.16 and 3.18 however the rotation matrix equation will be used more as this has advantages when it comes to control synthesis. Figure 3.5 shows how these equations connect in the model.

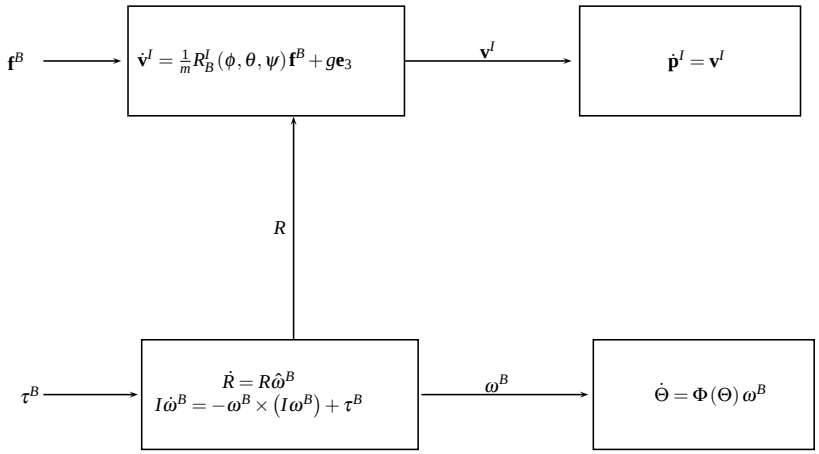


Figure 3.5: Full quadrotor model[1]

3.7 External Wrench Model

In the previous section the full nonlinear model of the quadrotor UAV was derived. In this section expressions for the torques and forces that are acting on the quadrotor are derived, these forces are referred to as the external wrench acting on the quadrotor body. The modelling approach presented here is adapted from that presented in [1], [69].

Consider the force vector $\mathbf{f}^B = \begin{bmatrix} f_x^B & f_y^B & f_z^B \end{bmatrix}^T$ acting on the quadrotor frame in the body frame, the quadrotor's rotors do not flap and thus the thrust from the rotors will always be along the z-axis of the body frame. Now if the aerodynamical forces(e.g air drag, rotor hub forces) are assumed to be negligible then the force vector will only consist of the k^B component equal to the thrust. Based on these assumptions the quadrotor translational dynamics can be approximated by the equations:

$$\dot{\mathbf{v}} = -\frac{1}{m} R_B^I(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ T_T \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} g \tag{3.33}$$

where T_T is the total thrust generated by the four rotors. The total thrust being the sum of the thrust generated by the four rotors.

$$T_T = \sum_{i=1}^{i=4} T_i \tag{3.34}$$

where T_i is the thrust generated by the i^{th} rotor.

The torques that are exerted on the quadrotor body are due to the thrust from the rotors. Consider the schematic of the quadrotor shown in figure 3.6 with the given rotation directions for the propellers. If the quadrotor's arms are

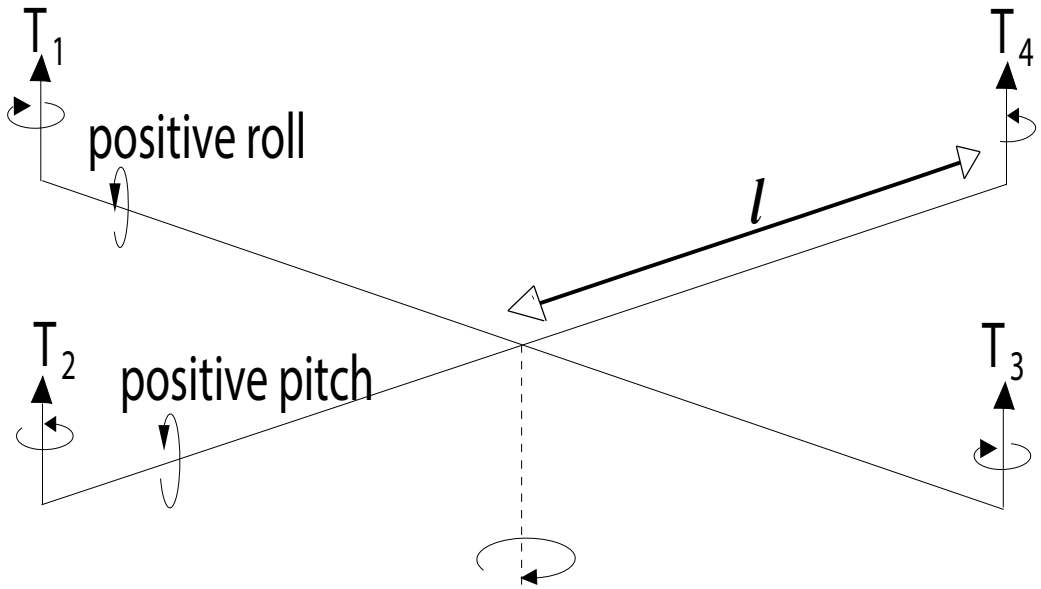


Figure 3.6: Quadrotor UAV schematic showing propeller rotation directions

of length lm each then the torques can be calculated. The rolling torque is due

to the forces of the second and fourth propellers and is given by:

$$\tau_\phi = l\left(T_2 - T_4\right) \tag{3.35}$$

The pitching torque is produced by the action of the third and first propellers and this torque is :

$$\tau_\theta = l\left(T_3 - T_1\right) \tag{3.36}$$

The yawing torque is produced due to the counter torque that is produced by the rotating propellers. Thus the yawing torque will be:

$$\tau_\psi = \tau_{c_1} - \tau_{c_2} + \tau_{c_3} - \tau_{c_4} \tag{3.37}$$

where τ_{c_i} is the counter-torque produced by the i^{th} rotor. The counter torque produced by a rotor rotating with angular velocity Ω is proportional to Ω^2:

$$\tau_{c_i} = c_Q \Omega^2 \tag{3.38}$$

The constant of proportionality c_Q can be experimentally determined using static thrust tests[70].

The thrust generated by a propeller rotating with speed Ω in steady state(i.e the rotor is not moving vertically or horizontally) is given by[70] :

$$T_i = C_T \rho A_{r_i} r_i^2 \Omega_i^2 \tag{3.39}$$

where for the i^{th} rotor, A_{r_i} is the rotor disk area, r_i is the radius, Ω_i is the angular velocity, C_T is the thrust coefficient of the propeller and ρ is the density of the air. In general the constants in equation 3.39 can be lumped up into one thrust constant(c_{T_i}) which can then be experimentally determined. The relationship between the torque, total thrust and the propeller angular velocity can be

compactly represented using matrices by:

$$\begin{bmatrix} T_T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & lc_T & 0 & -lc_T \\ -lc_T & 0 & lc_T & 0 \\ c_Q & \text{-}c_Q & c_Q & \text{-}c_Q \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{3.40}$$

3.8 Simplified Model

In the previous section using a number of simplifying assumptions expressions for the torques and forces that are exerted on the quadrotor frame are developed. Applying these to the quadrotor model of section 3.6 the resulting simplified model is given by:

$$\dot{\mathbf{p}}^I = \mathbf{v}^I \tag{3.41}$$

$$\dot{\mathbf{v}}^I = -\frac{1}{m} Re_3 T_T + ge_3 \tag{3.42}$$

$$\dot{R}e_3 = R\hat{\omega}^B e_3 \tag{3.43}$$

$$\mathbf{I}\dot{\omega}^B = -\omega^B \times \left(\mathbf{I}\omega^B\right) + \tau^B \tag{3.44}$$

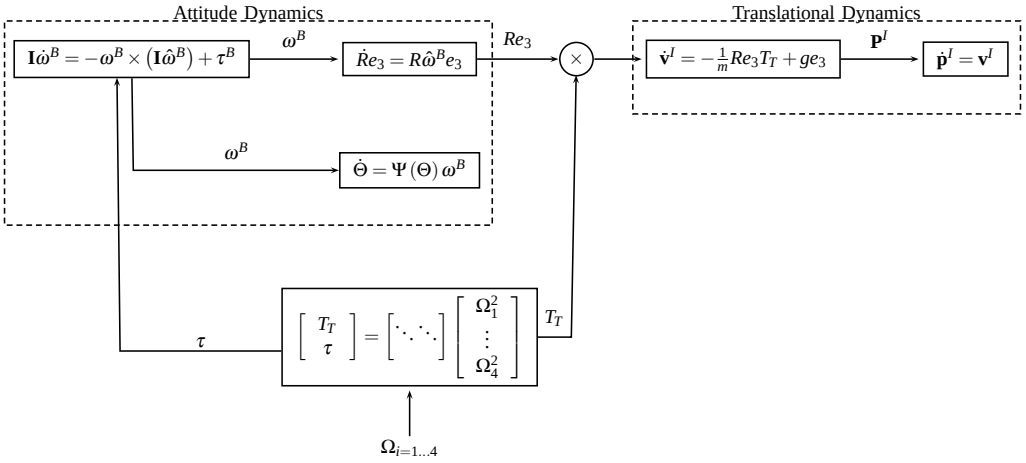


Figure 3.7: Quadrotor simplified model

Figure 3.7 shows a diagram of the simplified model and how the system can be

divided into attitude and translational subsystems. From figure 3.7 it can be seen that the attitude subsystem is totally decoupled from the translational subsystem as such the attitude subsystem can be viewed as a kind of actuator system for the translational subsystem. In most quadrotor models that are used in the literature the angular velocity(ω) is assumed to be equal to the Euler angular rates($\dot{\mathbf{\Theta}}$). In the model that is presented here such a simplification is not employed as it is strictly only valid for hover conditions.

3.9 Conclusion

A derivation of the mathematical model of the quadrotor UAV has been developed in this chapter under the assumptions of a rigid quadrotor UAV body and negligible aerodynamic forces. Euler angles are used for attitude representation and it is also discussed at length why Euler angles only provide a locally valid coordinate chart for the $SO\,(3)$ manifold. In as much as the derived model follows the approach used in most literature the model that has been derived is more comprehensive in that it does not assume the angular velocity(ω) to be equal to the Euler angle rates($\dot{\mathbf{\Theta}}$).

Chapter 4

Control Strategy Overview

4.1 Chapter Overview

This section provides a high level description of the approach that is taken in the design of the quadrotor controller. By exploiting the strict feedback interconnection of the translational and rotational subsystems a backstepping inspired control strategy is developed. The mathematical details of the control procedure are left for the next chapters, however in this chapter the focus is on how the different subsystem controllers connect and are related to each other.

4.2 Quadcopter Dynamics

For convenience the quadrotor model equations derived in the previous chapter are restated here.

$$\dot{\mathbf{p}}^I = \mathbf{v}^I \tag{4.1}$$

$$\dot{\mathbf{v}}^I = -\frac{1}{m} R\mathbf{e}_3 T_T + g\mathbf{e}_3 \tag{4.2}$$

$$\dot{R}\mathbf{e}_3 = R\hat{\omega}^B\mathbf{e}_3 \tag{4.3}$$

$$\mathbf{I}\dot{\omega}^B = -\omega^B \times \left(\mathbf{I}\omega^B\right) + \tau^B \tag{4.4}$$

$$\dot{\mathbf{\Theta}} = \mathbf{\Psi}\left(\mathbf{\Theta}\right)\omega^B \tag{4.5}$$

The configuration space of the quadrotor UAV is the 6 dimensional Special

Euclidean group $SE(3) = \mathbb{R}^3 \times SO(3)$, however looking at the system equations above the quadrotor has only 4 actuator inputs($T_T,\ \tau^B = [\tau_\phi,\ \tau_\theta,\ \tau_\psi]$). This means that the quadrotor is an underactuated system as it has more degrees of freedom than actuators. Now consider the subsystem divisions of the quadrotor model that were introduced in 3.8. The translational subsystem is described by equations (4.1) and (4.2) while the attitude subsystem is described by equations (4.3)-(4.5). Now from this it can be seen that the attitude subsystem is fully actuated having three actuators($\tau_\phi,\ \tau_\theta\ \tau_\psi$) and three degrees of freedom($\phi,\ \theta,\ \psi$). On the other hand the translational subsystem is underactuated having just one actuator(T_T) while having three degrees of freedom($x^I,\ y^I,\ z^I$).

Looking at the quadrotor dynamics as depicted in figure 3.7 the attitude subsystem can be viewed as actuating the translational subsystem with the actuating inputs being the total thrust T_T and the vector Re_3. From the properties of the rotation matrix, Re_3 is a unit vector and thus the vector Re_3T_T is just the thrust vector where the magnitude is given by T_T and the direction by the unit vector Re_3.

4.3 Problem Statement

Here a formal statement of the control problem which is investigated in this work is presented.

Problem Statement. *Given a trajectory in 3 dimensional space defined by the pair* $\left(\boldsymbol{p}_d^I(t), \psi_d(t)\right)$ *where* $\boldsymbol{p}_d^I(t)$ *is the desired position and* $\psi_d(t)$ *is the desired yaw, find a set of admissible controls* $\left(T_T, \tau^B\right)$ *such that the tracking error pair* (e_p, e_ψ) *defined by* $e_{\boldsymbol{p}} = \boldsymbol{p}^I - \boldsymbol{p}_d^I$ *and* $e_\psi = \psi - \psi_d$ *tends asymptotically to zero*

4.4 Control Strategy

The translational and the attitude dynamics can be viewed as subsystems that are interconnected in a strict feedback form as depicted in figure 4.1 and so a backstepping type of control can be used. The vector Re_3T_T can be viewed as a form of pseudo-control for the translational subsystem. This has the advantage of

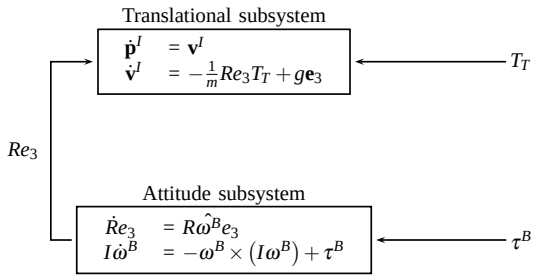


Figure 4.1: Attitude and Translational subsystem interconnection that shows the "partial" feedback form

solving the underactuation problem of the translational subsystem as it provides 2 extra "controls" thus making the subsystem "fully" actuated. Noting that Re_3 is a unit vector this means that only 2 of the 3 terms in Re_3 are required in order to uniquely identify this vector thus providing 2 not 3 extra controls for the translational subsystem. Once the pseudo-control for the translational subsystem is designed a controller for the attitude subsystem can be designed such that the unit vector Re_3 tracks a reference value defined by the translational pseudo-control. The general structure of the control strategy that will be adopted is shown in figure 4.2 where Re_{31d} and Re_{32d} are the first and second elements of the desired unit vector Re_{3d} generated by the translational controller.

4.5 Conclusion

It has been shown in this chapter that the dynamics of the quadrotor UAV are underactuated. Exploiting the strict feedback form of the interconnection of the attitude and translational subsystems a backstepping based control strategy is devised for the whole system. This approach solves the problem of the system being underactuated by introducing pseudo-controls into the system. The following chapters shall delve into the detailed design of the translational and attitude controllers.

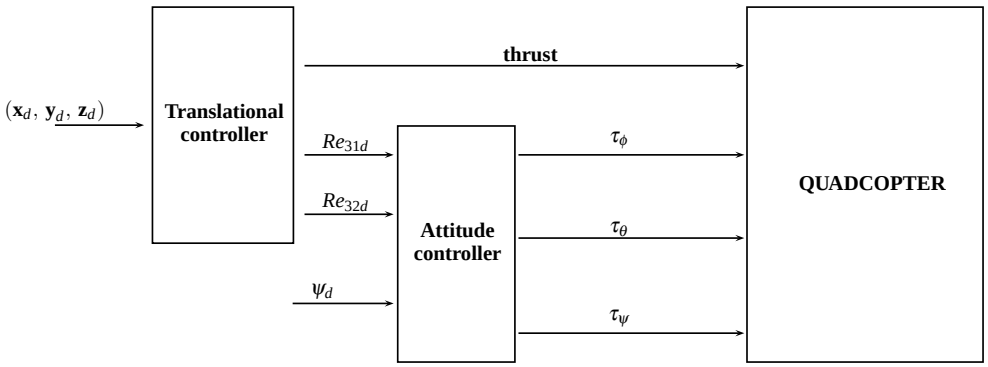


Figure 4.2: Backstepping based control strategy

Chapter 5

Translational Control

5.1 Chapter Overview

The translational controller's task is to ensure at the very least bounded error trajectory tracking. Given that such environmental phenomena like wind are unavoidable the controller should be robust enough to withstand such disturbances. Other disturbances that are present in the system are introduced by the high order dynamics such as air drag that have been neglected in the modelling stage. It is imperative that the designed controller be robust if it is to be of any practical use. In this chapter we present two important results in this regard, firstly we present a novel robust bounded controller based on the result of A.R Teel[34]. By adding sliding mode like terms to Teel's nonlinear saturated control a more robust controller with improved disturbance rejection characteristics is developed. The second result is the development of conditions on the controller gains that ensure that the quadrotor UAV's attitude is always non-singular which translates physically to requiring that the UAV does not overturn in flight.

5.2 Translational Dynamics

For convenience the dynamic equations of the translational subsystem are restated here.

$$\dot{\mathbf{p}}^I = \mathbf{v}^I \tag{5.1}$$

$$\dot{\mathbf{v}}^I = -\frac{1}{m} R e_3 T_T + g e_3 + \Delta(t) \tag{5.2}$$

where $\Delta(t)$ is an unknown bounded function of time that models all the uncertainties. The control problem that is to be solved for the translational subsystem can be stated as:

Translational Control Problem Formulation. *Given the time parameterised vector of the desired trajectory $\boldsymbol{p}_d(t)$ find $R\boldsymbol{e}_3$ and T_T such that for the tracking error $\boldsymbol{e}_p = \boldsymbol{p}_d^I(t) - \boldsymbol{p}^I(t)$ there exists $\epsilon > 0$ such that $\forall\, t > 0, \parallel \boldsymbol{e}_p(t) \parallel < \epsilon$.*

Thus from the formulation of the control problem the requirement is that at least bounded error tracking of the desired trajectory should be achieved. Considering the translational subsystem dynamics given by equations (5.1) and (5.2) the error dynamics can be formulated as:

$$\dot{\mathbf{e}}_p = \mathbf{e}_v \tag{5.3}$$

$$\dot{\mathbf{e}}_v = -\ddot{\mathbf{p}}_d - \frac{1}{m} R \mathbf{e}_3 T_T + g \mathbf{e}_3 + \Delta(t) \tag{5.4}$$

Applying the input transformation $R\mathbf{e}_3 T_T = m(-\nu + g\mathbf{e}_3 - \ddot{\mathbf{p}}_d)$ where ν becomes our new control, the translational dynamics are then transformed to the double integrator system given by :

$$\dot{\mathbf{e}}_p = \mathbf{e}_v \tag{5.5}$$

$$\dot{\mathbf{e}}_v = \nu + \Delta(t) \tag{5.6}$$

5.3 Stabilization of Multiple Integrator System with Bounded Controls

Practical control systems have actuators that can produce limited control effort as such it is desirable that designed controls be bounded also. The control of systems with bounded controllers is not a new concept but has been used for a long time. One of the popular methods that uses bounded controls is the bang-bang optimal controller. It was shown in [71] that for a general linear system of order $n \geq 3$ a bounded linear feedback controller cannot achieve global stabilization and thus one needs to consider nonlinear feedback control laws. Teel [34] proposed a bounded nonlinear feedback control scheme that achieves global stabilization of integrator chains. The construction of Teel's control law is fairly simple and as such is going to be the basis for the translational controller which will be developed in this chapter.

5.3.1 Teel's Method

This section presents the result and proof of Teel's bounded controller[34]. The proof will follow closely the method presented in [34]. Before stating the result consider the saturation function definition:

Definition. *Given two constants L and M such that $L, M > 0$ and $L < M$, a function $\sigma : \mathbb{R} \to \mathbb{R}$ is a saturation function if it is continuous, non-decreasing and satisfies the following conditions:*

1. $s\sigma(s) > 0 \; \forall s \neq 0$
2. $\sigma(s) = s$ *when* $|s| < L$
3. $|\sigma(s)| \leq M \; \forall s \in \mathbb{R}$

From condition 3 in the definition M defines the saturation level of the function σ, the constant L from condition 2 defines the linear region of the saturation function. Condition 1 constrains the saturation function to the first and the third quadrants. To illustrate the kind of functions that satisfy the definition consider

two functions σ_1 and σ_2 for M = 2 and L = 1 which are plotted in figure 5.1.

$$\sigma_1(s) = sign(s)\, min(|s|, M) \tag{5.7}$$

$$\sigma_2(s) = \begin{cases} s & \text{if } |s| \leq L \\ L + (M-L)\left(1 - e^{-\frac{(s-L)}{\tau}}\right) & \text{if } s > L \\ -L - (M-L)\left(1 - e^{-\frac{(-s-L)}{\tau}}\right) & \text{if } s < -L \end{cases} \tag{5.8}$$

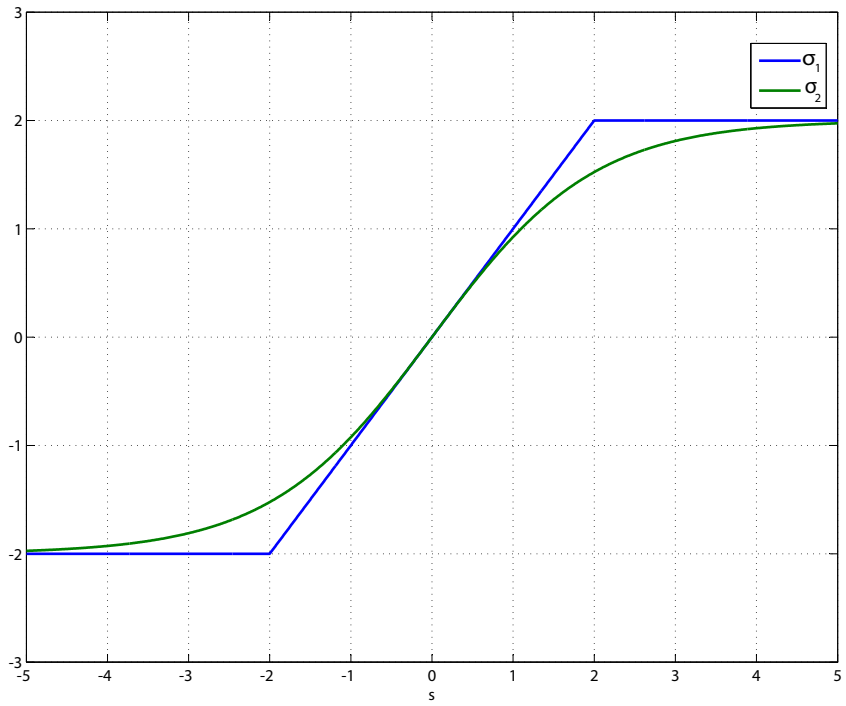


Figure 5.1: Saturation functions σ_1 and σ_2

Now let us consider the n^{th} order system given by :

$$\dot{x_1} = x_2, \quad \ldots \quad , \dot{x}_n = u. \tag{5.9}$$

According to [34] the control law given by the following theorem will globally asymptotically stabilize the system given in (5.9).

Theorem. *There exists a function $h_i : \mathbb{R}^N \to \mathbb{R}$ such that for a set of saturation functions σ_i for $i = 1, \ldots, n$ defined by positive constants (L_i, M_i) for which $L_i < M_i$ and $M_j < \frac{1}{2} L_{j+1}$ for $j = 1, \ldots, (n-1)$. The bounded control*

$$u = -\sigma_n \left(h_n (x) + \sigma_{n-1} \left(h_{n-1} (x) + \cdots + \sigma_1 \left(h_1 (x) \right) \right) \ldots \right)$$

makes the system in (5.9) *globally asymptotically stable.*

Proof. Assume that there exists a linear coordinate transformation $y = Tx$ that transforms the system given by (5.9) into the form $\dot{y} = Ay + Bu$ where A and B are given by:

$$A = \begin{bmatrix} 0 & 1 & \ldots & 1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & 1 \\ 0 & \ldots & \ldots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

The control from the stated theorem is then:

$$u = -\sigma_n \left(y_n + \sigma_{n-1} \left(y_{n-1} + \ldots + \sigma_1 \left(y_1 \right) \right) \ldots \right)$$

The closed loop dynamics of the new transformed system become:

$$\begin{aligned} \dot{y}_1 &= y_2 + \cdots + y_n - \sigma_n \left(y_n + \sigma_{n-1} \left(y_{n-1} + \ldots + \sigma_1 \left(y_1 \right) \right) \ldots \right) \\ &\vdots \\ \dot{y}_{n-1} &= y_n - \sigma_n \left(y_n + \sigma_{n-1} \left(y_{n-1} + \ldots + \sigma_1 \left(y_1 \right) \right) \ldots \right) \\ \dot{y}_n &= -\sigma_n \left(y_n + \sigma_{n-1} \left(y_{n-1} + \ldots + \sigma_1 \left(y_1 \right) \right) \ldots \right) \end{aligned}$$

Now consider the dynamics of the state y_n which is described by the equation:

$$\dot{y}_n = -\sigma_n \left(y_n + \sigma_{n-1} \left(y_{n-1} + \ldots + \sigma_1 \left(y_1 \right) \right) \ldots \right)$$

Let $V_n = \frac{1}{2} y_n^2$ be a candidate Lyapunov function, the time derivative of which is:

$$\dot{V}_n = -y_n \left[\sigma_n \left(y_n + \sigma_{n-1} \left(y_{n-1} + \ldots + \sigma_1 \left(y_1 \right) \right) \ldots \right) \right]$$

From the definition of the saturation function $y_n\sigma_n(y_n) > 0$ will be positive if:

- when y_n is positive $y_n + \sigma_{n-1}(\dots)$ is also positive
- when y_n is negative $y_n + \sigma_{n-1}(\dots)$ is also negative

This condition is satisfied in the region $|y_n| > M_{n-1}$, from the theorem this region can be expressed as $|y_n| > \frac{1}{2}L_n$. Thus $\dot{V}_n$ will be negative $\forall\ y_n \notin Q_n = \{y_n : |y_n| \leq \frac{1}{2}L_n\}$. This implies that y_n will enter the set Q_n after some finite time and will remain therein, we can also safely assume that during this time the other states$(y_1, \dots, y_{n-1})$ will remain bounded.

Now consider the dynamics of y_{n-1}. Note that it has been established that after some finite time $y_n \leq \frac{1}{2}L_n$, the argument of the function σ_n becomes also bounded :

$$|y_n + \sigma_{n-1}(y_{n-1} + \ldots + \sigma_1(y_1))| \leq \frac{1}{2}L_n + M_{n-1} \leq L_n \left(\text{recall } M_i \leq \frac{1}{2}L_{i+1}\right)$$

From the definition of the saturation function $\sigma(s) = s$ when $|s| < L$ thus once y_n enters Q_n, σ_n enters it linear region according to the definition. Thus after some finite time the dynamics of y_{n-1} become:

$$\dot{y}_{n-1} = -\sigma_{n-1}(y_{n-1} + \ldots + \sigma_1(y_1))$$

Defining another candidate Lyapunov function $V_{n-1} = \frac{1}{2}y_{n-1}^2$ the time derivative of which is given by

$$\dot{V}_{n-1} = -y_{n-1}\sigma_{n-1}(y_{n-1} + \sigma_{n-2}(y_{n-2} + \ldots + \sigma_1(y_1))\ldots)$$

Applying the same reasoning as in the analysis of the dynamics of y_n after some finite time y_{n-1} will be trapped in the set $Q_{n-1} = \{y_{n-1} : |y_{n-1}| \leq \frac{1}{2}L_{n-1}\}$. This reasoning can be applied to all the states y_i and it will be seen that after some finite time all the saturation functions σ_i will be linear and the complete system

dynamics will become

$$\begin{aligned}
\dot{y}_1 &= -y_1 \\
\dot{y}_2 &= -y_1 - y_2 \\
&\vdots \\
\dot{y}_{n-1} &= -y_1 - y_2 - \ldots - y_{n-1} \\
\dot{y}_n &= -y_1 - y_2 - \ldots - y_{n-1} - y_n
\end{aligned}$$

Thus after some finite time the system dynamics become exponentially stable.

□

It is interesting to note that the control law that has been laid out in the theorem above is such that all the eigenvalues of the closed loop system when it enters the linear region are equal to -1[38].

Example. *To illustrate Teel's bounded control consider the third order integrator system given by:*

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = x_3, \quad \dot{x}_3 = u \tag{5.10}$$

We want to find a u that makes the system globally asymptotically stable.

First we must find a linear transformation $y = Tx$ as stated in the proof. According to [72] such a transformation is given by:

$$y_{n-i} = \sum_{j=0}^{i} \frac{i!}{j!\,(i-j)!} x_{n-j} \tag{5.11}$$

Applying this formula yields the following transformation in matrix form

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{5.12}$$

Thus the control will be given by :

$$u = -\sigma_3 \left(y_3 + \sigma_2 \left(y_2 + \sigma_1 \left(y_1 \right) \right) \right) \tag{5.13}$$

The following combination of constants for the saturation functions are chosen. These values where arrived at by trial and error.

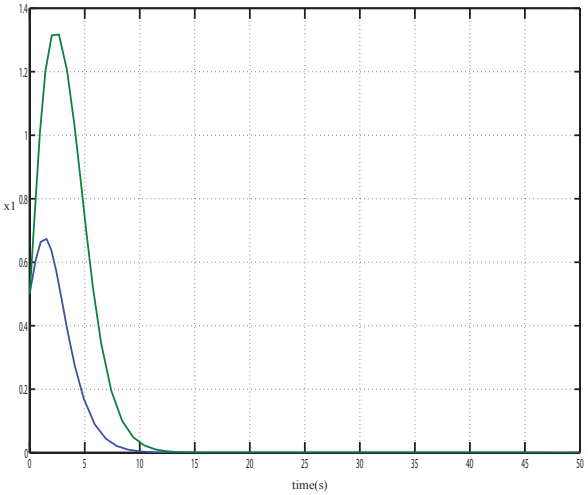
$$\begin{array}{cc} M_3 = 4 & L_3 = 3 \\ M_2 = 1.4 & L_2 = 1.2 \\ M_1 = 0.5 & L_1 = 0.4 \end{array}$$

Simulations of the controller described by (5.13) were done using MATLAB, the results of the simulation are shown is figure 5.2. The simulations were done for different initial conditions, the blue plot corresponds to the initial conditions ($x_1 = 0.5$, $x_2 = 0.2$, $x_3 = 0.1$)and the green plot to ($x_1 = 0.5$, $x_2 = 0.5$, $x_3 = 0.5$). For the blue plot the initial conditions are such that y_1, y_2 and y_3 start within the sets Q_i for $i = 1, 2, 3$ respectively while for the blue plot y_1,y_2 and y_3 start outside these sets. From the simulations it can be seen that in both cases the controller does ensure asymptotic stability however the transient response of the states is greatly affected by how far out of the Q_i sets the states are initially. This deterioration of transient response in Teel's controller is highlighted in the work of Marchand[72] where a modification is proposed in which the saturation levels for the saturation are no longer static but vary depending on the states. Consider the performance of Teel's control in the presence of a bounded matched disturbance input in the form of a sine wave. With initial conditions of ($x_1 = 0$, $x_2 = 0$, $x_3 = 0$) the system was simulated in MATLAB and the results are shown in figure 5.3.

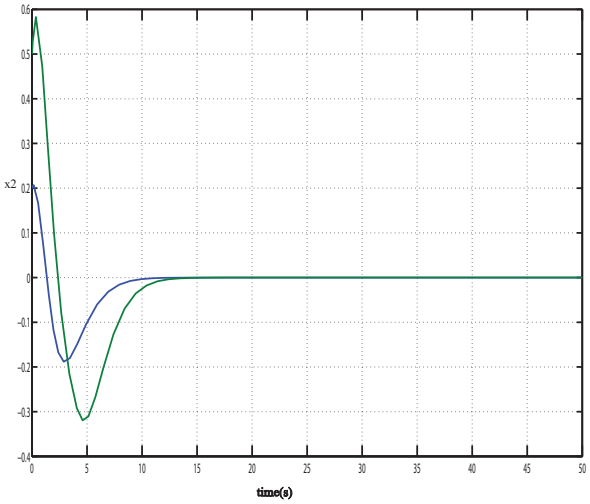
From the simulation results shown in figure 5.3 it can be seen that the effect of the disturbance input is quite marked, for x_1 the input to state gain with regards to the disturbance in nearly equal to 0.5. The poor disturbance rejection performance of Teel's controller is also discussed in the author's work [**?**] in which Teel's controller is used to design a controller for UAV trajectory tracking.

5.3.2 Modified Teel's Method

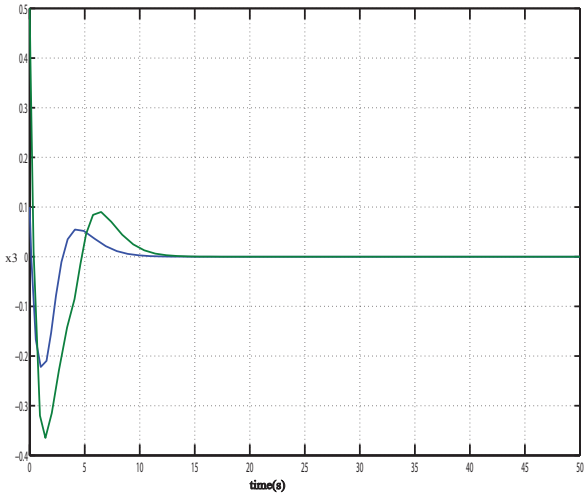
Given the relatively poor disturbance rejection performance of the controller presented in the previous section we propose a modification to the control in order to improve the controller's robustness. For the modified controller, saturation functions as defined in Teel's controller are used.



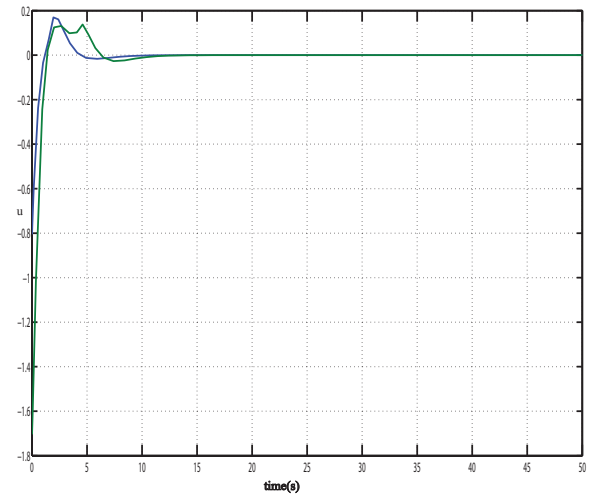
(a) x_1 plot



(b) x_2 plot

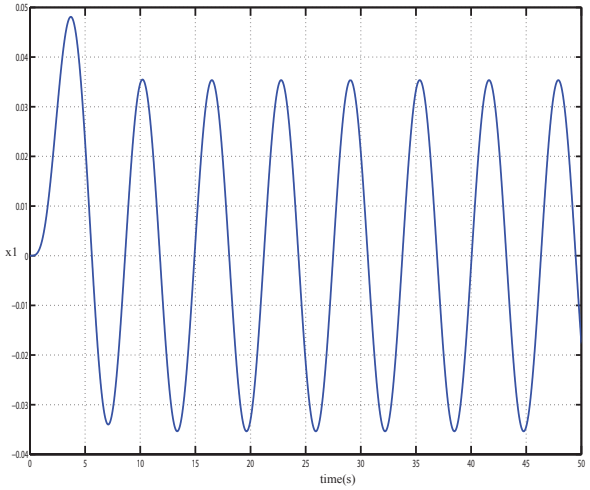


(c) x_3 plot

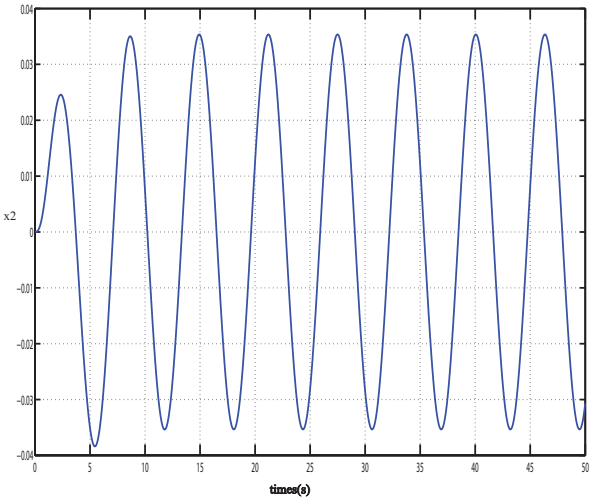


(d) control action

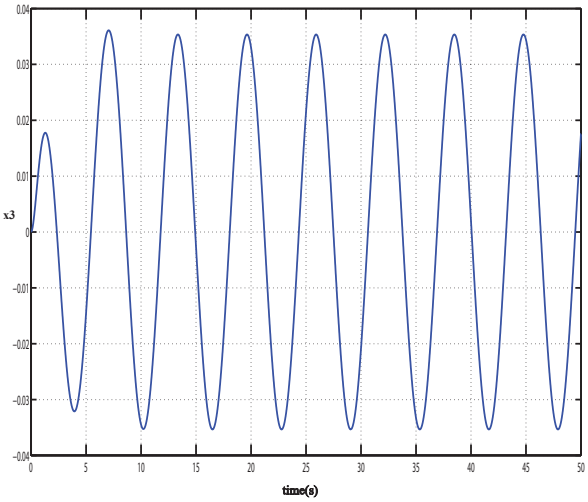
Figure 5.2: System response with Teel controller. Blue is for initial condition (x_1 = 0.5, $x_2 = 0.2$, $x_3 = 0.1$) and green ($x_1 = 0.5$, $x_2 = 0.5$, $x_3 = 0.5$)



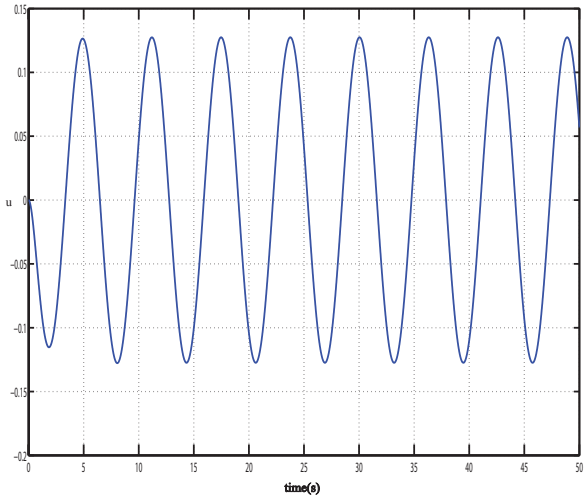
(a) x_1 plot



(b) x_2 plot



(c) x_3 plot



(d) control action

Figure 5.3: State trajectory of system with a sinusoidal disturbance input of amplitude 0.1

Theorem. *Consider the n^{th} order integrator system with a bounded disturbance input $\delta(t)$ such that $|\delta(t)| < \Delta \; \forall \; t$ where Δ is a positive constant.*

$$\dot{x}_1 = x_2, \quad , \dot{x}_n = u + \delta(t) \tag{5.14}$$

Then there exists a set of functions $h_i : \mathbb{R}^N \to \mathbb{R}$ such that for a set of saturation functions σ_i for $i = 1, \ldots, n$ defined by the positive constants (L_i, M_i) the bounded control :

$$u = -\sigma_n\left(h_n(x) - K_{n-1} sign\left(h_{n-1}(x)\right) + \sigma_{n-1}\left(h_{n-1}(x) - K_{n-2} sign\left(h_{n-2}(x)\right) + \ldots\right.\right.$$
$$\left.\left. + \; \sigma_2\left(h_2(x) - K_1 sign\left(h_1(x)\right) + \sigma_1\left(h_1(x)\right)\right)\ldots\right)\right) - K_n sign\left(h_n(x)\right)$$

where $sign(.)$ is the signum function defined as :

$$sign(x) = \begin{cases} 1 & when \; x > 0 \\ -1 & when \; x < 0 \end{cases}$$

ensures that for the system (5.14) $X \in Q \; \forall \; t > t_0$ where $X = (x_1, \ldots)$, Q is a closed set in which the origin is an interior point if the following conditions are met.

$$\begin{array}{cc} K_n = \Delta + \eta_1 & \\ K_{n-j} = -\sum_{i=0}^{j-1} K_{n-1} - \Delta - \eta_{j+1} & \text{for } j = 1, \ldots, n-1 \\ L_i < M_i \text{ and } M_j < \frac{1}{2} L_{j+1} & \text{for } i = 1, \ldots, n \\ 2K_{n-j} + M_{n-j} < \frac{1}{2} L_{n-j+1} & \end{array}$$

where $\eta_i > 0$ fori $= 1, \ldots, n$.

Proof. Assume that there exists a linear coordinate transformation $y = Tx$ that transforms the system given by (5.14) into the form $\dot{y} = Ay + Bu$ where A and B are given by:

$$A = \begin{bmatrix} 0 & 1 & \ldots & 1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & 1 \\ 0 & \ldots & \ldots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

The bounded control from the theorem can be expressed as:

$$u = -\sigma_n \left(y_n - K_{n-1} sign\left(y_{n-1}\right) + \sigma_{n-1}\left(\ldots\right)\ldots \right) - K_n sign\left(y_n\right)$$

The closed loop dynamics of the system then become:

$$\begin{aligned}
\dot{y}_1 &= y_2 + \ldots + y_n + \delta\left(t\right) - \sigma_n \left(y_n - K_{n-1} sign\left(y_{n-1}\right) + \sigma_{n-1}\left(\ldots\right)\ldots \right) - K_n sign\left(y_n\right) \\
&\vdots \\
\dot{y}_{n-1} &= y_n + \delta\left(t\right) - \sigma_n \left(y_n - K_{n-1} sign\left(y_{n-1}\right) + \sigma_{n-1}\left(\ldots\right)\ldots \right) - K_n sign\left(y_n\right) \\
\dot{y}_n &= \delta\left(t\right) - \sigma_n \left(y_n - K_{n-1} sign\left(y_{n-1}\right) + \sigma_{n-1}\left(\ldots\right)\ldots \right) - K_n sign\left(y_n\right)
\end{aligned}$$

Now consider the dynamics of y_n which are given by the equation:

$$\dot{y}_n = \delta\left(t\right) - \sigma_n \left(y_n - K_{n-1} sign\left(y_{n-1}\right) + \sigma_{n-1}\left(\ldots\right) \right) - K_n sign\left(y_n\right)$$

Let $V_n = \frac{1}{2} y_n^2$ be a candidate Lyapunov function the time derivative of which is equal to:

$$\dot{V}_n = -y_n \sigma_n \left(y_n - K_{n-1} sign\left(y_{n-1}\right) + \sigma_{n-1}\left(\ldots\right) \right) + y_n \delta\left(t\right) - K_n y_n sign\left(y_n\right) \quad (5.15)$$

Consider the last two terms in equation (5.15).

$$\begin{aligned}
y_n \delta\left(t\right) - K_n y_n sign\left(y_n\right) &\leq |y_n| \Delta - K_n |y_n| \\
&= |y_n| \Delta - |y_n| \Delta - |y_n| \eta_1 \\
&= -\eta_1 |y_n|
\end{aligned}$$

The whole expression of $\dot{V}_n$ is required to be negative. To achieve this we require $-y_n \sigma_n\left(\ldots\right)$ to be negative also. It should be noted that this requirement is somewhat quite conservative as it should be clear that strictly speaking the condition for negativity of $\dot{V}_n$ is that $-y_n \sigma_n\left(\ldots\right) < \eta_n |y|_n$.

For $-y_n \sigma_n\left(\ldots\right)$ to be negative from the definition of the saturation function this amounts to requiring that the argument of σ_n $\left(y_n - K_{n-1} sign\left(y_{n-1}\right) + \right.$

$\sigma_{n-1}(\ldots)$) should have the same sign as y_n.

$$\begin{aligned}
|y|_n &> |-K_{n-1}sign(y_{n-1})+\sigma_{n-1}(\ldots)| \text{ which is satisfied if} \\
&> |-K_{n-1}sign(y_{n-1})|+|\sigma_{n-1}(\ldots)| \\
&= K_{n-1}+M_{n-1} \\
&= K_{n-1}+\frac{1}{2}L_n
\end{aligned}$$

Thus $\dot{V}_n \leq 0$ for all $y_n \notin Q_n$ where $Q_n = \{y_n : |y|_n < K_{n-1} + \frac{1}{2}L_n\}$. This means then that after some finite time y_n will be trapped inside the set Q_n. Now consider what happens once y_n enters the set Q_n, we have inside Q_n:

$$\begin{aligned}
|y_n - K_{n-1}sign(y_{n-1})+\sigma_{n-1}(\ldots)| &< |y_n|+|K_{n-1}sign(y_{n-1})|+|\sigma_{n-1}(\ldots)| \\
&= K_{n-1}+\frac{1}{2}L_n+K_{n-1}+M_{n-1} \\
&= 2K_{n-1}+\frac{1}{2}L_n+M_{n-1} \\
&< L_n \quad \text{recall } 2K_{n-j}+M_{n-j} < \frac{1}{2}L_{n-j+1}
\end{aligned}$$

Thus once y_n enters the set Q_n, $\sigma_n(\ldots)$ enters also its linear region and the dynamics of y_{n-1} become:

$$\begin{aligned}
\dot{y}_{n-1} &= -\sigma_{n-1}(y_{n-1}-K_{n-2}sign(y_{n-2})+\sigma_{n-2}(\ldots))+K_{n-1}sign(y_{n-1}) \\
&- K_n sign(y_n)+\delta(t) \qquad (5.16)
\end{aligned}$$

Let $V_{n-1} = \frac{y_{n-1}^2}{2}$ be a candidate Lyapunov function whose time derivative is:

$$\begin{aligned}
\dot{V}_{n-1} &= -y_{n-1}\sigma_{n-1}(y_{n-1}-K_{n-2}sign(y_{n-2})+\sigma_{n-2}(\ldots))+K_{n-1}y_{n-1}sign(y_{n-1}) \\
&- K_n y_{n-1} sign(y_n)+y_{n-1}\delta(t) \qquad (5.17)
\end{aligned}$$

Consider the last three terms of equation (5.17):

$$\left.\begin{aligned} K_{n-1}y_{n-1}sign\left(y_{n-1}\right) - K_n y_{n-1}sign\left(y_n\right) \\ + y_{n-1}\delta\left(t\right) \end{aligned}\right\} \leq K_{n-1}|y|_{n-1} + K_n|y|_{n-1} + \Delta|y|_{n-1}$$

$$= -\eta_2|y|_{n-1} \text{from theorem } K_{n-1} = -K_n - \Delta - \eta_2$$

To ensure that $\dot{V}_{n-1} \leq 0$ the requirement is that $-y_{n-1}\sigma_{n-1}\left(\ldots\right) < 0$, as was the case for $\dot{V}_n$. This is satisfied if:

$$\begin{aligned} |y_{n-1}| &> |-K_{n-2}sign\left(y_{n-1}\right) + \sigma_{n-2}\left(\ldots\right)| \text{ which is satisfied if} \\ &> |-K_{n-2}sign\left(y_{n-1}\right)| + |\sigma_{n-2}\left(\ldots\right)| \\ &= K_{n-2} + M_{n-2} \\ &= K_{n-2} + \frac{1}{2}L_{n-1} \end{aligned}$$

Thus $\dot{V}_{n-1} \leq 0$ for all $y_{n-1} \notin Q_{n-1}$ where $Q_{n-1} = \{y_{n-1} : |y_{n-1}| < K_{n-2} + \frac{1}{2}L_{n-1}$. After some finite time y_{n-1} will be trapped in the set Q_{n-1} and once inside Q_{n-1} we have:

$$\begin{aligned} |y_{n-1} - K_{n-2}sign\left(y_{n-2}\right) + \sigma_{n-2}\left(\ldots\right)| &< |y_{n-1}| + |K_{n-2}sign\left(y_{n-2}\right)| + |\sigma_{n-2}\left(\ldots\right)| \\ &< K_{n-2} + \frac{1}{2}L_{n-1} + K_{n-2} + M_{n-2} \\ &= 2K_{n-2} + \frac{1}{2}L_{n-1} + M_{n-2} \\ &< L_{n-1} \text{ from theorem } 2K_{n-j} + M_{n-j} < \frac{1}{2}L_{n-j+1} \end{aligned}$$

Thus once y_{n-1} enters the set Q_{n-1}, $\sigma_{n-1}\left(\ldots\right)$ also enters its linear region. Applying the same analysis to the whole system it will be seen that after some finite time y_j will be trapped in the set $Q_j = \{y_j : |y|_j < K_{j-1} + \frac{1}{2}L_j\}$. Once all the

states have entered the set Q_j the system dynamics become

$$\begin{aligned}
\dot{y}_1 &= -y_1 - K_n sign\,(y_n) + K_{n-1} sign\,(y_{n-1}) + \ldots + K_1 sign\,(y_1) + \delta\,(t) \\
\dot{y}_2 &= -y_1 - y_2 - K_n sign\,(y_n) + K_{n-1} sign\,(y_{n-1}) + \ldots + K_1 sign\,(y_1) + \delta\,(t) \\
&\vdots \qquad\quad \vdots \\
\dot{y}_n &= -y_1 - \ldots - y_n - K_n sign\,(y_n) + K_{n-1} sign\,(y_{n-1}) + \ldots + K_1 sign\,(y_1) + \delta\,(t)
\end{aligned}$$

It still remains to be proved if the above system dynamics are stable in the Lyapunov sense or not. □

Example. *To illustrate our controller consider the third order integrator system given by:*

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = x_3, \quad \dot{x}_3 = u \tag{5.18}$$

Using the transformation stated in the previous example:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{5.19}$$

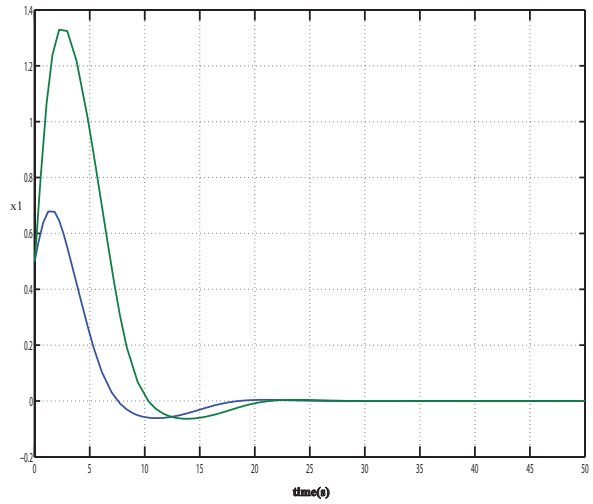
The control u can then be expressed as:

$$u = -\sigma_3\,(y_3 - K_2 sign\,(y_2) + \sigma_2\,(y_2 - K_1 sign\,(y_1) + \sigma_1\,(y_1))) - K_3 sign\,(y_3) \tag{5.20}$$

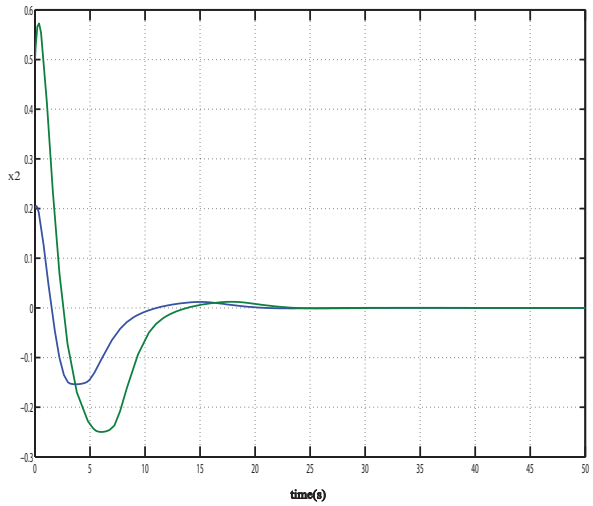
Assuming that the disturbance is a sinusoid of amplitude 0.1 and choosing the following values for the controller parameters by trial error.

$$\begin{array}{lll} K_1 = 0.05 & M_1 = 0.5 & L_1 = 0.4 \\ K_2 = 0.03 & M_2 = 1.4 & L_2 = 1.2 \\ K_3 = 0.15 & M_3 = 4 & L_3 = 3 \end{array}$$

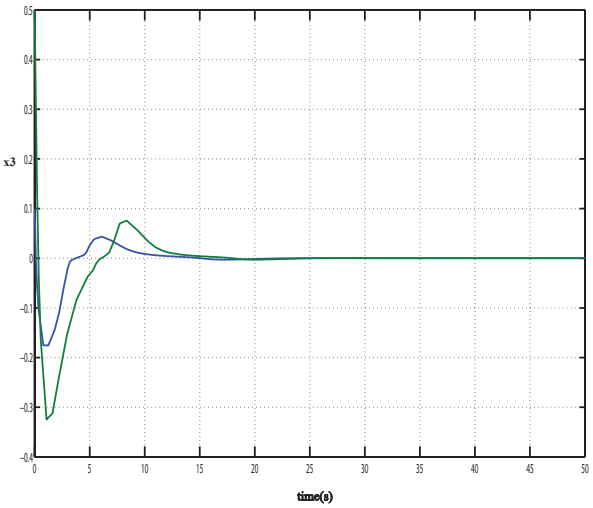
The controller was simulated in the MATLAB/SIMULINK environment for both cases in which disturbances are present and absent. For the case where disturbances are absent the controller was simulated for two sets of initial conditions ($x_1 = 0.5$, $x_2 = 0.2$, $x_3 = 0.1$) and ($x_1 = 0.5$, $x_2 = 0.5$, $x_3 = 0.5$), results of which are shown in figure 5.4.



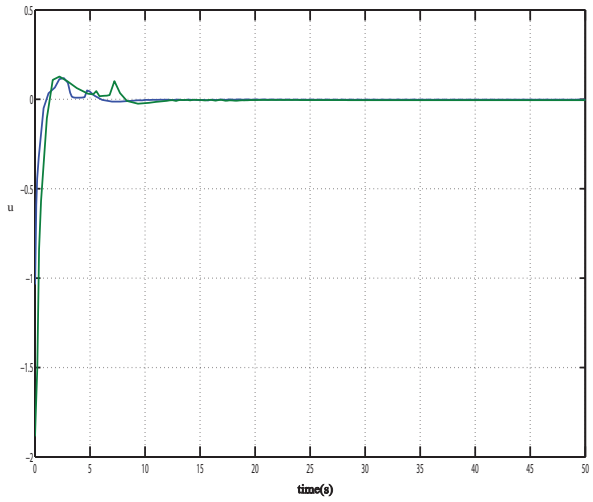
(a) x_1 plot



(b) x_2 plot



(c) x_3 plot

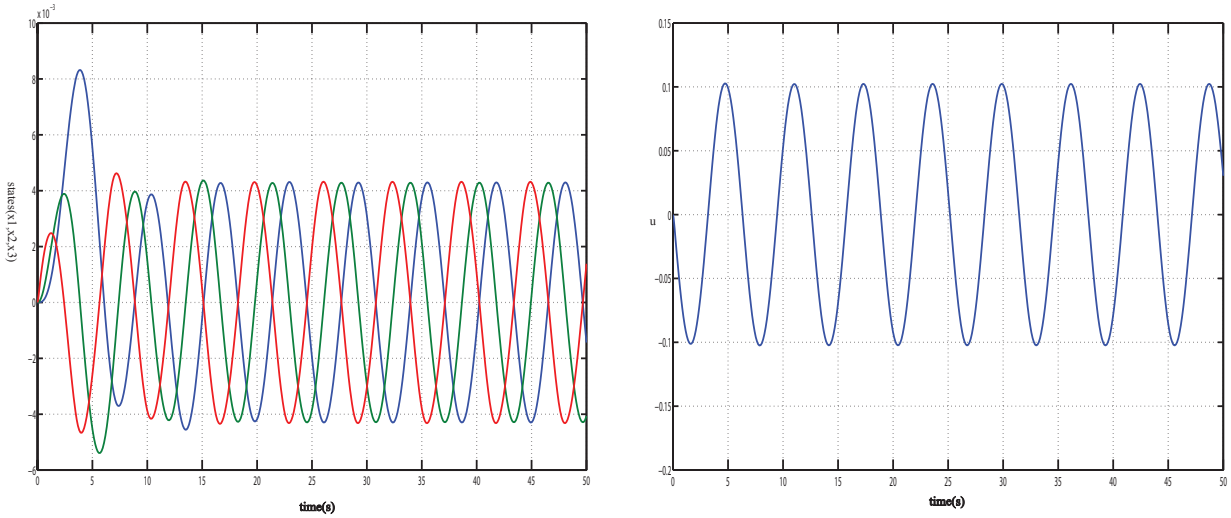


(d) control action

Figure 5.4: State trajectory of system without disturbance. Blue is for initial condition ($x_1 = 0.5$, $x_2 = 0.2$, $x_3 = 0.1$) and green ($x_1 = 0.5$, $x_2 = 0.5$, $x_3 = 0.5$)

It can be seen that the system response does not differ much from the response of the controller based on Teel's method. The proposed controller does have a longer settling convergence period. The controller was also tested for the case where there is a sinusoidal disturbance of amplitude 0.1 and for the initial

conditions ($x_1 = 0.5$, $x_2 = 0.5$, $x_3 = 0.5$) and ($x_1 = 0$, $x_2 = 0$, $x_3 = 0$) results of these simulations are shown in figures 5.5 and 5.6.



(a) x_1 = blue, x_2 = green, x_3 = red

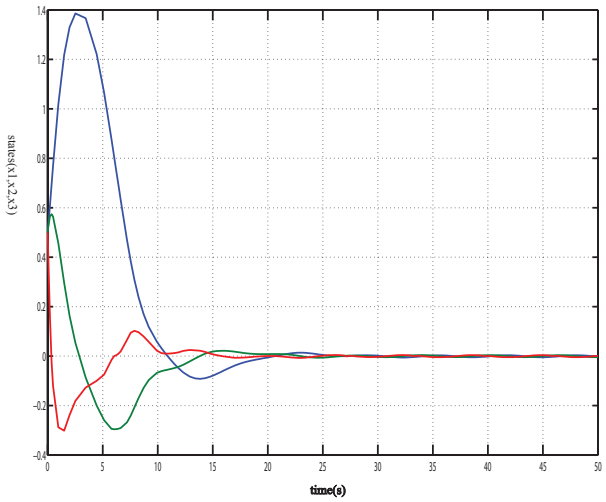
(b) control action

Figure 5.5: System response with zero initial conditions and sinusoidal disturbance amplitude 0.1

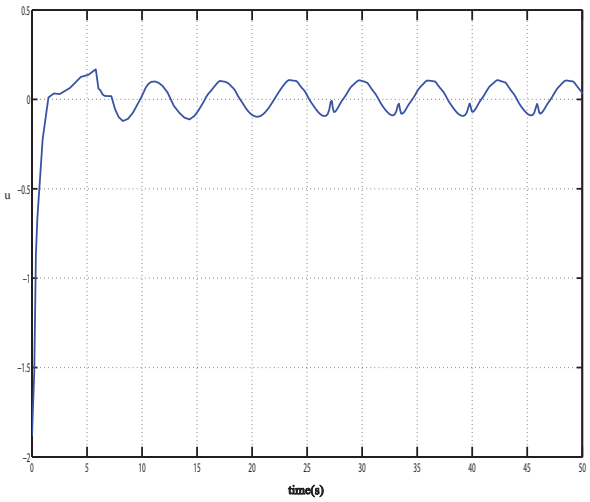
The simulations show that the modified controller out performs the original controller, for the case with zero initial conditions we can see that for the modified controller the states are bounded to within 5×10^{-3} from the origin as opposed to the original controller in which the states are bounded to within 4×10^{-2} from the origin. Also in figure 5.6 it can be seen that the modified controller is able to stabilise the system for non-zero initial conditions with little variation in the performance compared to the case where the disturbance is absent.

5.4 Translational Controller

In section 5.2 it was shown how the translational dynamics could be transformed into a simple double integrator by the following input transformation $Re_3 = m\left(-\nu + ge_3 - \ddot{\mathbf{p}}_d\right)$ where ν is the new control to be designed. Let $\chi = \int e_p dt$ be the integral position error. The transformed translational dynamics will be



(a) x_1 = blue, x_2 = green, x_3 = red



(b) control action

Figure 5.6: System response with initial conditions ($x_1 = 0.5$, $x_2 = 0.5$, $x_3 = 0.5$) and sinusoidal disturbance amplitude 0.1

described by the third order system:

$$\dot{\chi} = \mathbf{e}_p \tag{5.21}$$

$$\dot{\mathbf{e}}_p = \mathbf{e}_v \tag{5.22}$$

$$\dot{\mathbf{e}}_v = \nu + \Delta(t) \tag{5.23}$$

Applying the controller developed in section 5.3.2 results in the following expression

$$Re_3 T_T = m\left[\Sigma_3\left(\mathbf{y}_3 - K_2 sign\left(\mathbf{y}_2\right) + \Sigma_2\left(\mathbf{y}_2 - K_1 sign\left(\mathbf{y}_1\right) + \Sigma_1\left(\mathbf{y}_1\right)\right)\right) - K_3 sign\left(\mathbf{y}_3\right) - \ddot{\mathbf{p}}_d + g\mathbf{e}_3\right] \tag{5.24}$$

where:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \chi \\ \mathbf{e}_p \\ \mathbf{e}_v \end{bmatrix} \tag{5.25}$$

Σ_i is a 3×1 vector of saturation functions $[\sigma_{i1}\ \sigma_{i2}\ \sigma_{i3}]^T$ and K_i is a 3×3 diagonal matrix of gains $\text{diag}(K_{i1}\ K_{i2}\ K_{i3})$ for $i = 1, 2, 3$. Now recall that one of the properties of the rotation matrix is that its columns/rows are unit vectors thus Re_{3d} should be a unit vector for it to be a valid rotation matrix column. Using this knowledge we can extract from (5.24) expressions for Re_3 and T_T.

$$Re_{3d} = \frac{-\ddot{\mathbf{p}}_d + ge_3 + \Sigma_3(\ldots) - K_3 sign(\mathbf{y}_3)}{\|-\ddot{\mathbf{p}}_d + ge_3 + \Sigma_3(\ldots) - K_3 sign(\mathbf{y}_3)\|} \tag{5.26}$$

$$T_T = m\|-\ddot{\mathbf{p}}_d + ge_3 + \Sigma_3(\ldots) - K_3 sign(\mathbf{y}_3)\| \tag{5.27}$$

Since Euler angles are being used to represent the attitude, the commanded attitude(Re_{3d}) must be such that it avoids gimbal lock. Consider the unit vector Re_3 which from the rotation matrix is given by :

$$Re_3 = \begin{bmatrix} sin\phi cos\theta + cos\phi sin\theta cos\psi \\ -cos\psi sin\phi + sin\psi sin\theta cos\phi \\ cos\theta cos\phi \end{bmatrix} \tag{5.28}$$

As was stated earlier the Euler representation of the vehicle attitude breaks down when $\phi, \theta = 90^0$. Consider the third element of the vector $Re_3 = cos\theta cos\phi$ now if $Re_{33} = cos\theta cos\phi > 0$ this is sufficient to guarantee that $\phi, \theta \neq \frac{\pi}{2}$. Thus when tracking the orientation command generated by the translational controller Re_{3d}, the same condition should be met. In the proposition below it is shown that this can be done by appropriately selecting the parameters of the controllers.

Proposition. *Consider the saturation function σ_{33} which is defined by the constants M_{33} and L_{33}. R_{33d} which is the third element of the vector Re_{3d} is strictly positive if:*

$$g - M_{33} - K_{33} > max(\ddot{\boldsymbol{p}}_{dz})(t)$$

which implies that $|\theta(t)|_d, |\phi(t)|_d < \frac{\pi}{2}\ \forall\ t > t_0$

Proof. To verify the above proposition consider the third element of the vector

Re_{3d} which we shall denote as Re_{33d} we have:

$$\begin{aligned} Re_{33d}(t) > 0 &\implies -\ddot{p}_{dz} + g + \sigma_{33}(\ldots) - K_{33} sign(y_{33}) > 0 \\ &\implies -\max(\ddot{p}_{dz}) + g - M_{33} - K_{33} > 0 \\ &\implies g - M_{33} - K_{33} > \max(\ddot{p}_{dz}) \ \forall\, t > t_0 \end{aligned}$$

□

5.5 Conclusion

In this chapter a translational controller whose design is based on Teel's bounded controller was developed. Teel's control strategy has been modified leading to the development of a novel robust bounded controller with superior disturbance rejection qualities. Also in this chapter conditions have been formulated on the controller parameters that ensure that the vehicle rotation matrix is always defined. Physically these conditions ensure that the UAV does not over turn in flight.

Chapter 6

Attitude Control 1

6.1 Chapter Overview

In the previous chapter a translational controller which generates a desired orientation of the vehicle by defining the desired value of the unit vector Re_3 was developed. It is the task of the attitude controller to ensure that asymptotic tracking of the desired attitude as generated by the translational controller is achieved. One of the major difficulties in designing the attitude controller is the fact that the inertia of the quadrotor UAV is difficult to accurately determine given the irregular shape of the vehicle. As such the designed controller is required to be robust against parameter uncertainty as well as disturbances from torques produced by wind and other aerodynamic forces. The strict feedback form of the orientation dynamics makes backstepping control an ideal method to use, however as was highlighted in section 2.3.2 the backstepping method does not cater for the presence of uncertainties. The main result of this chapter is the development of a robust backstepping controller based on the adaptive control method and nonlinear damping.

6.2 Backstepping Control

Backstepping control is a recursive design technique which was developed in the 1990s by the concerted effort of several researchers. The backstepping technique

in its present form first appeared in the work of Saberi,Kokotovic and Sussman[50] with further developments being made by Kanellakopoulos et al[51]. To introduce the idea of backstepping control let us consider a system given by the equations (6.1) and (6.2).

$$\dot{x} = f(x) + g(x)\,\xi \tag{6.1}$$

$$\dot{\xi} = u \tag{6.2}$$

where $x \in \mathbb{R}^N$ and $\xi, u \in \mathbb{R}$ and $f(0) = 0$. This general system can be viewed as being some general nonlinear system defined by (6.1) which has been concatenated with a simple integrator system as is shown in figure 6.1.

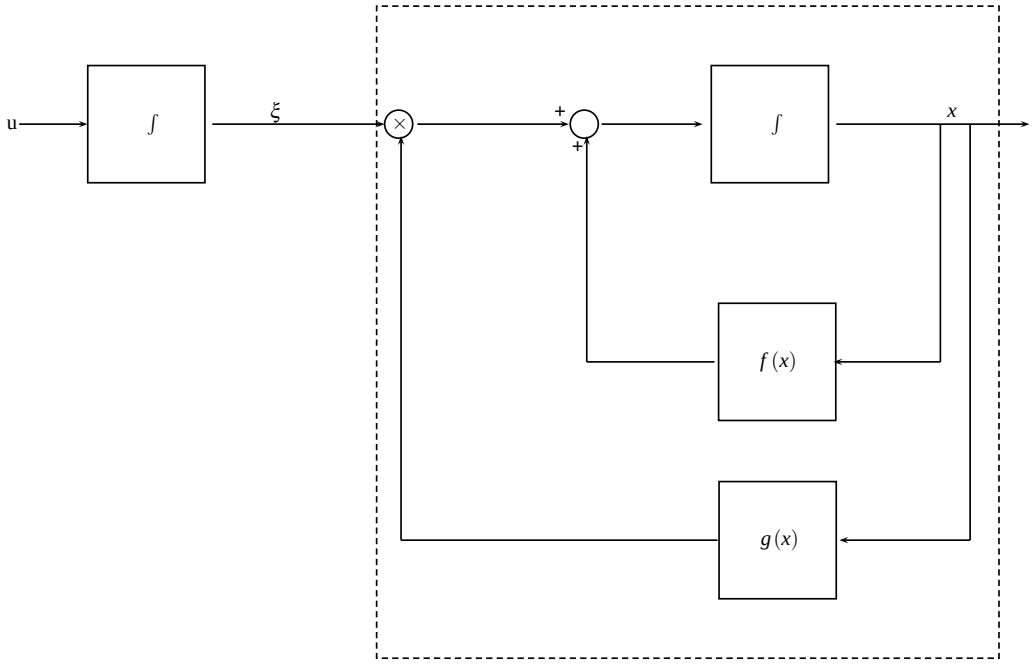


Figure 6.1: Block diagram of the system

The fundamental idea behind the backstepping technique is that if one can consider the system described by (6.1) and (6.2) and assume that the subsystem described by equation (6.1) can be stabilised if ξ is taken as the control. Back-

stepping allows the use of this knowledge to design the overall control(u) of the system by stepping back as it were past the integrator from the "pseudo-control" ξ. The power of this approach is in the fact that it can easily be recursively applied to higher order systems as long as they are in strict feedback form.

Backstepping Lemma. *[73] Consider a system described by equation* (6.1)*. Consider* ξ *as the control for this system then there exists a continuously differentiable function* $\alpha(x)$ *and a smooth, positive definite and radially unbounded function* $V : \mathbb{R}^N \to \mathbb{R}$ *such that*

$$\text{if} \quad \xi = \alpha(x), \quad \alpha(0) = 0 \quad \text{then} \tag{6.3}$$

$$\frac{\partial V}{\partial x}\left[f(x) + g(x)\,\alpha(x)\right] \leq -W(x), \quad \forall x \in \mathbb{R}^N \tag{6.4}$$

where $W : \mathbb{R}^N \to \mathbb{R}$ *is a positive semidefinite function. If the stated assumption is satisfied then for the complete system described by equations* (6.1) *and* (6.2) *the control:*

$$u = -\lambda\left(\xi - \alpha(x)\right) + \frac{\partial \alpha}{\partial x}(x)\left[f(x) + g(x)\,\xi\right] - \frac{\partial V}{\partial x}(x)\,g(x), \quad \lambda > 0 \tag{6.5}$$

will ensure that if $W(x)$ *is positive definite then* $x = 0$, $\xi = 0$ *will be globally asymptotically stable. Otherwise if* $W(x)$ *is positive semidefinite then* $x(t)$ *and* $\xi(t)$ *will converge to the largest invariant set* M_a *contained in the set* $E_a = \{x, \xi | W(x) = 0, \xi = \alpha(x)\}$

Proof. This proof follows from closely the proof in [73]The assumption contained in the first part of the lemma states that if ξ was the actual control then the system given by equation (6.1) could be stabilised by choosing $\alpha(x)$ as the control. However since in actuality ξ is not the control, let $z = \xi - \alpha(x)$ be an "error" variable. Thus the system equations can be transformed from the (x, ξ) space to (x, z) in which case the dynamics become:

$$\dot{x} = f(x) + g(x)\left[z + \alpha(x)\right] \tag{6.6a}$$

$$\dot{z} = u - \frac{\partial \alpha}{\partial x}\left[f(x) + g(x)\left(z + \alpha(x)\right)\right] \tag{6.6b}$$

Consider now the function $V_a(x,z)$:

$$V_a(x,z) = V(x) + \frac{z^2}{2} \tag{6.7}$$

$V_a(x,z)$ is a candidate Lyapunov function which is formed by augmenting the Lyapunov function $V(x)$ with the radially unbounded and positive definite function $\frac{z^2}{2}$. The derivative of this augmented candidate Lyapunov function along the solutions of (6.6) is

$$\begin{aligned}
\dot{V}_a(x,z) &= \frac{\partial V}{\partial x}(x)\left[f(x) + g(x)\alpha(x) + g(x)z\right] + z\left[u - \frac{\partial \alpha}{\partial x}(x)\left(f(x) + g(x)\left(z + \alpha(x)\right)\right)\right] \\
&= \frac{\partial V}{\partial x}(x)\left[f(x) + g(x)\alpha(x)\right] + z\left[u + \frac{\partial V}{\partial x}(x)\, g(x) - \frac{\partial \alpha}{\partial x}(x)\left(f(x) + g(x)\left(z + \alpha(x)\right)\right)\right] \\
&\leq -W(x) + z\left[u + \frac{\partial V}{\partial x}(x)\, g(x) - \frac{\partial \alpha}{\partial x}(x)\left(f(x) + g(x)\left(z + \alpha(x)\right)\right)\right]
\end{aligned} \tag{6.8}$$

Choosing the control u as :

$$u = -\lambda z - \frac{\partial V}{\partial x}(x)\, g(x) + \frac{\partial \alpha}{\partial x}(x)\left[f(x) + g(x)\left(z + \alpha(x)\right)\right], \quad \lambda > 0 \tag{6.9}$$

then the derivative of the augmented candidate Lyapunov function becomes:

$$\dot{V}_a(x,z) \leq -W(x) - \lambda z^2 \tag{6.10}$$

Now if $W(x)$ was positive definite then $V_a(x,z)$ will be negative definite and by Lyapunov's theory this is sufficient to guarantee global asymptotic stability of the origin($x = 0$, $z = 0$). Since $z = \xi - \alpha(x)$ this implies that:

$$\begin{aligned}
\lim_{t\to\infty} z &= \lim_{t\to\infty} \left(\xi - \alpha(x)\right) = 0 \\
&\implies \lim_{t\to\infty} \xi - \alpha\left(\lim_{t\to\infty} x\right) = 0 \\
&= \lim_{t\to\infty} \xi - \alpha(0) = 0 \\
&\implies \lim_{t\to\infty} \xi(t) = 0
\end{aligned} \tag{6.11}$$

Thus showing that the choice of control (6.5) ensures global asymptotic stability of the origin($x(t) = 0,\ \xi(t) = 0$). To analyse the case when $W(x)$ is positive semidefinite requires La Salle's Invariant set theorem[45] which states that

Invariant Set Theorem. *Consider the system* $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ *and let* $V(\boldsymbol{x})$ *be a scalar function. If there exists a region* Ω_l *in which* $V(\boldsymbol{x}) < l,\ l > 0$ *and* $\dot{V}(\boldsymbol{x}) \leq 0\ \forall \boldsymbol{x} \in \Omega_l$ *then every trajectory* $\boldsymbol{x}(t)$ *starting in* Ω_l *will tend to the largest invariant set* $M_a \subset E_a$ *where* $E_a = \{\boldsymbol{x}\,|\dot{V}(\boldsymbol{x}) = 0\}$

For the case when $W(x)$ is positive semi-definite the set $E_a = \{x, \xi\ |\dot{V}_a(x, z) = 0\} = \{x.\xi\ |W(x) = 0, z = 0\} = \{x, \xi\ |W(x) = 0, \xi = \alpha(x)\}$.

□

The backstepping lemma has been presented for a somewhat simple system however this same result can be applied to the larger class of systems that are in strict feedback form[73]. To illustrate this point consider the general strict feedback system:

$$\begin{aligned}
\dot{x} &= f(x) + g(x)\xi_1 \\
\dot{\xi}_1 &= f_1(x, \xi_1) + g_1(x, \xi_1)\xi_2 \\
\dot{\xi}_2 &= f_2(x, \xi_1, x_2) + g_2(x, \xi_1, \xi_2)\xi_3 \\
&\vdots \\
\dot{\xi}_{k-1} &= f_k(x, \xi_1, \ldots, \xi_{k-1}) + g_{k-1}(x, \xi_1, \ldots, \xi_{k-1})\xi_k \\
\dot{\xi}_k &= f_k(x, \xi_1, \ldots, \xi_k) + g_k(x, \xi_1, \ldots, x_k)u
\end{aligned} \tag{6.12}$$

This system is said to be in strict feedback form because the nonlinear functions f_i,g_i for$(i = 1, \ldots k)$ depend only on $x, \xi_1, \ldots, \xi_i$. If the x-subsystem satisfies the assumption of the Backstepping Lemma with ξ_1 as the control the recursive design starts by first defining the error variable $z_1 = \xi_1 - \alpha_1(x)$. The $(x,\ z_1)$ dynamics then become:

$$\begin{aligned}
\dot{x} &= f(x) + g(x)\left(z_1 + \alpha_1(x)\right) \\
\dot{z}_1 &= f_1\left(x_1, \xi_1\right) - \frac{\partial \alpha_1}{\partial x}\left[f(x) + g(x)\left(z_1 + \alpha_1(x)\right)\right] + g_1(x, \xi_1)\xi_2
\end{aligned} \tag{6.13}$$

Constructing the augmented Lyapunov candidate function for (6.13):

$$V_1(x, z_1) = V(x) + \frac{z_1^2}{2} \tag{6.14}$$

Taking ξ_2 as the control for the system (6.13), the task now is to find a stabilising function $\alpha_1(x, z_1)$ for ξ_2 such that the derivative of (6.14) is nonpositive when $\xi_2 = \alpha_1$.

$$\begin{aligned}\dot{V}_1 &= \frac{\partial V}{\partial x}(x)\left(f(x) + g(x)\alpha(x)\right) + z_1\left[\frac{\partial V}{\partial x}(x) + f_1(x_1, \xi_1)\right. \\ &\left. - \frac{\partial \alpha}{\partial x}(x)\left[f(x) + g(x)\left(z_1 + \alpha(x)\right)\right] + g_1(x, \xi_1)\xi_2\right] \\ &\leq -W(x) + z_1\left[\frac{\partial V}{\partial x}(x) + f_1(x_1, \xi_1) - \frac{\partial \alpha}{\partial x}(x)\left[f(x) + g(x)\left(z_1 + \alpha(x)\right)\right] + g_1(x, \xi_1)\xi_2\right]\end{aligned} \tag{6.15}$$

Thus $\alpha_1(x, z_1)$ can be chosen as:

$$\alpha_1(x, z_1) = \frac{1}{g_1(x, z_1)}\left[-\lambda_1 z_1 - \frac{\partial V}{\partial x}(x)g(x) - f_1(x_1, z_1) + \frac{\partial \alpha}{\partial x}(x)\left(f(x) + g(x)\left[z_1 + \alpha(x)\right]\right)\right] \tag{6.16}$$

with $\lambda_1 > 0$ which gives $\dot{V}_1 \leq -W_1(x, z_1)$ where $W_1(x, z_1) = W(x) + \frac{z_1^2}{2}$. The next step is to now augment (6.13) with the dynamics of ξ_2, to do this define another error variable $z_2 = \xi_2 - \alpha_1(x, z_1)$. The $(x,\ z_1,\ z_2)$ can be expressed compactly as:

$$\begin{aligned}\dot{X} &= F_1(X) + G_1(X)z_2 \\ \dot{z}_2 &= f_2(X_1, \xi_2) + g_2(X_1, \xi_2)\xi_3\end{aligned} \tag{6.17}$$

where:

$$X_1 = \begin{bmatrix} x \\ z_1 \end{bmatrix}, \quad G_1(X_1) = \begin{bmatrix} 0 \\ g_1(x, \xi_1) \end{bmatrix}$$

$$F_1(X_1) = \begin{bmatrix} f(x) + g(x)\xi \\ f_1(x, \xi_1) - \frac{\partial \alpha}{\partial x}\left(f(x) + g(x)\xi\right) + g_1(x, \xi_1)\alpha_1(x, z_1) \end{bmatrix}, \quad g_2(X_1, \xi_2) = g_2(x, \xi_1, \xi_2)$$

$$
\begin{aligned}
f_2(X_1,\xi_2) &= f_2(x,\xi_1,\xi_2) - \frac{\alpha_1}{\partial x}\left[f(x)+g(x)\xi\right] \\
&- \frac{\partial\alpha}{\partial z_1}\left[f_1(x,\xi_1) - \frac{\partial\alpha}{\partial x}\left(f(x)+g(x)\xi\right) + g_1(x,\xi_1)\xi_2\right]
\end{aligned}
$$

Following the approach used for the (x,z_1) subsystem the augmented candidate Lyapunov function $V_2(x,z_1,z_2)$ is constructed as:

$$
V_2(x,z_1,z_2) = V_1(X_1) + \frac{z_2^2}{2} = V(x) + \frac{z_1^2}{2} + \frac{z_2^2}{2} = V(x) + \sum_{i=1}^{2}\frac{z_i^2}{2} \tag{6.18}
$$

Having constructed the candidate Lyapunov function $V_2(x,z_1,z_2)$ the stabilising function $\alpha_2(x,z_1,z_2)$ can be chosen such that $\xi_3 = \alpha_2$ ensures that the derivative of V_2 is nonpositive. This procedure can be applied recursively to the system (6.12) and will terminate at the k^{th} step when the actual control u appears in the dynamics. At the k^{th} step the dynamics have the following structure:

$$
\begin{aligned}
\dot{X}_{k-1} &= F_{k-1}(X_{k-1}) + G_{k-1}(X_{k-1})z_k \\
\dot{z}_k &= f_k(X_{k-1},\xi_k) + g_k(X_{k-1},\xi_k)u
\end{aligned} \tag{6.19}
$$

Construct a candidate Lyapunov function for the whole system which is of the form:

$$
V_k(x,z_1,\ldots,z_k) = V(x) + \sum_{i=1}^{k}\frac{z_i^2}{2} \tag{6.20}
$$

Using this candidate Lyapunov function the control u can be chosen such that $\dot{V}_k$ is nonpositive. From this illustration it becomes clear then that the backstepping technique is actually a method of constructing a Lyapunov function for the whole system from a Lyapunov function of some simpler subsystem. The approach that has been outlined for applying backstepping to strict feedback systems can also be applied to wider classes of system such as pure-feedback systems and block-strict-feedback systems[73].

Example. *To illustrate the backstepping technique consider the system given by*

(6.21). *The objective is to regulate* $x(t)$ *for all* $x(0)$ *and* $\xi(0)$.

$$\dot{x} = cosx - x^3 + \xi \tag{6.21a}$$

$$\dot{\xi} = u \tag{6.21b}$$

The first step is to find a Lyapunov function $V(x)$ and a stabilising function $\alpha(x)$ such that if $\xi = \alpha(x)$ then (6.21a) will be rendered globally asymptotically stable about the origin. If $V(x) = \frac{x^2}{2}$ then the derivative along the solutions of (6.21a) is given by:

$$\dot{V} = \frac{\partial V}{\partial x}(x)\dot{x} = x\left[cosx - x^3 + \xi\right] \tag{6.22}$$

To ensure negative definiteness of $\dot{V}(x)$ let $\xi = \alpha(x) = -cosx$ which results in $\dot{V}(x) = -x^4$. Let the error variable be $z = \xi - \alpha(x)$, the system dynamics can thus be expressed in terms of (x,z) as:

$$\dot{x} = -x^3 + z \tag{6.23a}$$

$$\dot{z} = -sinx(z - x^3) + u \tag{6.23b}$$

Constructing a Lyapunov function by augmenting $V(x)$ with a quadratic term of the error variable z:

$$\begin{aligned} V_1(x,z) &= V(x) + \frac{z^2}{2} = \frac{x^2}{2} + \frac{z^2}{2} \\ \dot{V}_1 &= x(-x^3 + z) + z\left[u - sinx\left(z - x^3\right)\right] \\ &= -x^4 + z\left[u + x - sinx\left(z - x^3\right)\right] \end{aligned} \tag{6.24}$$

Let the control u be given by:

$$u = -x - \lambda z - sinx(x^3 - z), \quad \lambda > 0 \tag{6.25}$$

This ensures that $\dot{V}_1 = -x^4 - \lambda z^2$ which implies that $(x = 0, z = 0)$ is globally asymptotically stable. To determine the behavior of ξ with time consider the

limit:

$$\begin{aligned}\lim_{t\to\infty} z &= \lim_{t\to\infty} (\xi + cosx) = 0 \\ &\implies \lim_{t\to\infty} \xi + cos\left(\lim_{t\to\infty} x\right) = 0 \\ &\implies \lim_{t\to\infty} \xi = -1 \end{aligned} \tag{6.26}$$

The choice of $\alpha(x)$ in this example is such that $\alpha(0) \neq 0$, the effect of this choice is that ξ does not settle to zero as shown in the above analysis.The controller was simulated using the MATLAB/SIMULINK environment, results of these simulations are shown in figure 6.2 for different values of λ

Integral control action can be easily incorporated into the backstepping control strategy. For the example system (6.21) this can be accomplished by introducing the integral error variable $\chi = \int x dt$ such that the system dynamics become:

$$\dot{\chi} = x \tag{6.27a}$$

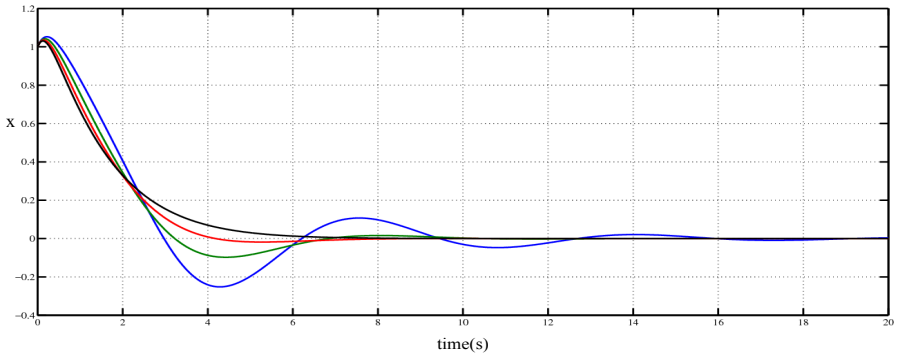
$$\dot{x} = cosx - x^3 + \xi \tag{6.27b}$$

$$\dot{\xi} = u \tag{6.27c}$$

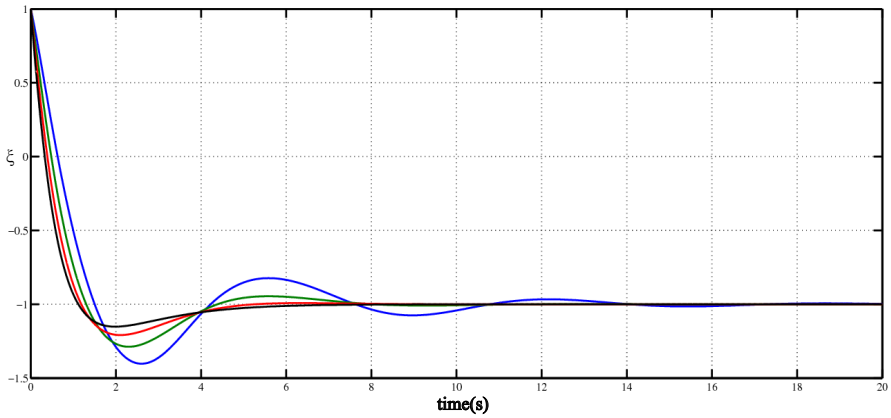
The system (6.27) is still in strict-feedback form and thus the backstepping approach can be used with the requirement being the regulation of $\chi(t)$ and $x(t)$. With the choice of $V(\chi) = \frac{\chi^2}{2}$ the following stabilizing functions and control will guarantee the regulation of $\chi(t)$ and $x(t)$.

$$\begin{aligned} \alpha(\chi) &= -\lambda_1\chi, \quad \lambda_1 > 0, \quad z_1 = x - \alpha(\chi) \\ \alpha_1(\chi, Z_1) &= -cosx + x^3 - \lambda_1 z + \lambda_1^2\chi - \chi - \lambda_2 z_1, \quad \lambda_2 > 0, \quad z_2 = \xi - \alpha_1(\chi, z_1) \\ u &= -z_1 - (\lambda_1^2 - 1)(\lambda_1\chi + z_1) - (\lambda_1 + \lambda_2)(z_2 - \chi - \lambda_2 z_1) - \lambda_3 z_2, \quad \lambda_3 > 0 \end{aligned} \tag{6.28}$$

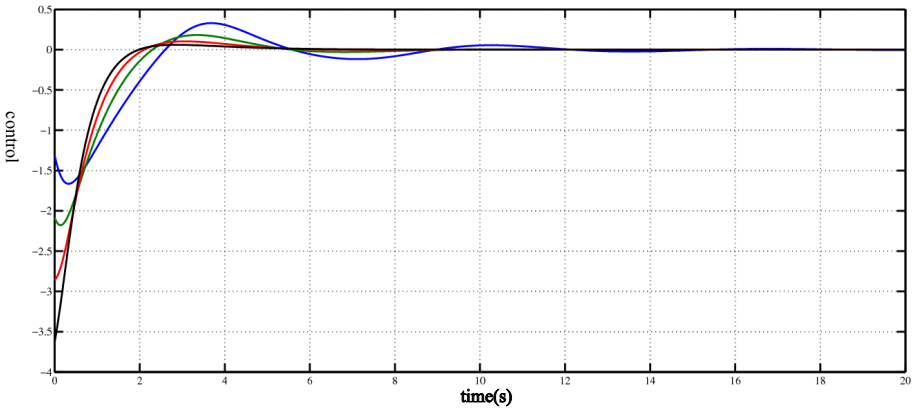
Figure 6.3 shows the response of the system with the integral backstepping controller, from these graphs one can see that the integral backstepping controller can achieve faster settling times though there is an increased overshoot which is characteristic of integral control.



(a) x plot

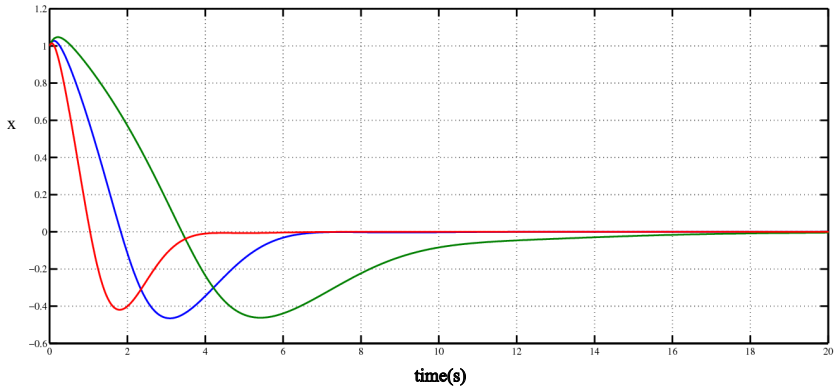


(b) ξ plot

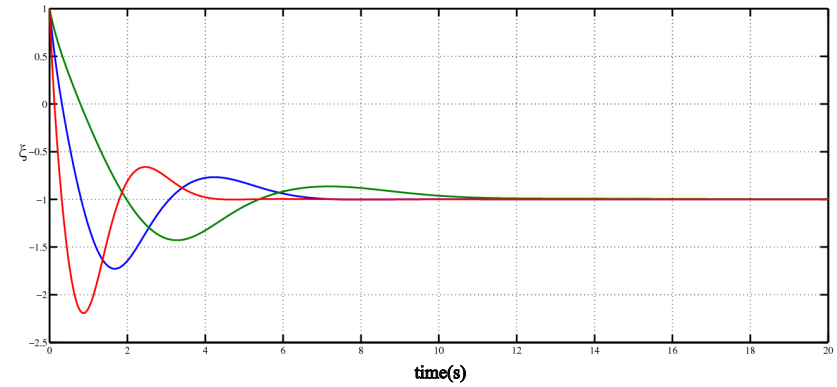


(c) control input

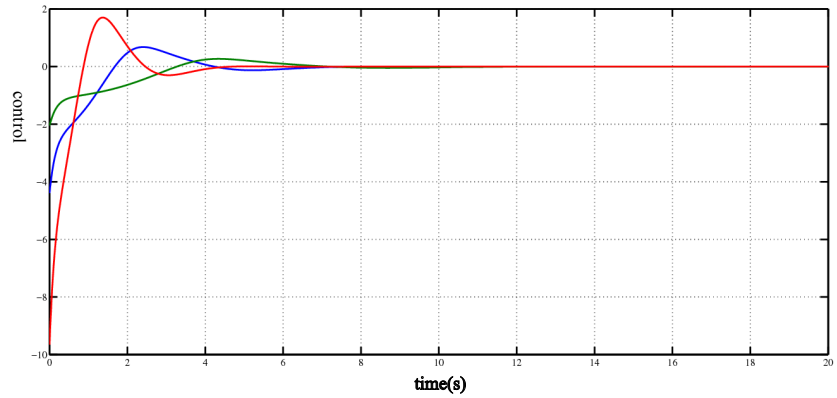
Figure 6.2: Backstepping example simulation results. Blue = ($\lambda = 0.5$), green = ($\lambda = 1$), red = ($\lambda = 1.5$), black = ($\lambda = 2$)



(a) x plot



(b) ξ plot



(c) control input

Figure 6.3: Integral Backstepping example simulation results. Blue $=(\lambda_1 = 0.7, \lambda_2 = 1, \lambda_3 = 0.8$, green $= (\lambda_1 = 0.5, \lambda_2 = 0.6, \lambda_3 = 0.4)$, red $= (\lambda_1 = 1.2, \lambda_2 = 1.6, \lambda_3 = 1)$.

6.3 Backstepping with Uncertainty

So far the focus has been on systems in which the dynamics are fully known, however in real life this is rarely the case as uncertain terms will invariably exist due mainly to simplifying assumptions made in the modelling process. These uncertainties can be grouped into two categories, parametric and non-parametric uncertainties. Parametric uncertainties as the name suggests are uncertainties in the values of the system's parameters such as the inertia of the quadrotor UAV. Non-parametric uncertainties describe those uncertainties that act as disturbance inputs to the system. In the case of the quadrotor UAV aerodynamics forces such as drag can be viewed as non-parametric uncertainties. The backstepping technique can be modified to take into account these uncertainties. The rest of this chapter will be devoted to developing such a backstepping technique. To achieve robustness against parametric uncertainty adaptive techniques are employed while nonlinear damping via Lyapunov redesign will be used to achieve robustness against disturbances.

6.3.1 Adaptive Backstepping Control

Adaptive backstepping control for the matched uncertainty case first appeared in the work of Taylor, Kokotovic, Marino and Kanellakopoulos[57] the extended matching case was later solved in [58] both cases are considered in this chapter. Adaptive backstepping for the matched case is presented first as this is more relevant to the attitude control problem. Consider the system given by (6.29) in which ϑ is an unknown constant scalar quantity.

$$\dot{x} = f(x) + g(x)\xi \tag{6.29a}$$

$$\dot{\xi} = \vartheta h(x, \xi) + u \tag{6.29b}$$

Assume that there exists a continuous function $\alpha(x)$ such that if $\xi = \alpha(x)$ then the subsystem (6.29a) would be globally asymptotically stable with a Lyapunov function $V(x)$ such that the following inequality is satisfied:

$$\dot{V}(x) = \frac{\partial V}{\partial x}\left(f(x) + g(x)\alpha(x)\right) \leq -W(x), \quad W(x) \text{ is positive definite} \tag{6.30}$$

Following the backstepping approach that has been outlined in the previous section let the error variable be defined as $z = \xi - \alpha(x)$, the system dynamics expressed in terms of (x,ξ) become:

$$\dot{x} = f(x) + g(x)\,(z + \alpha(x)) \tag{6.31a}$$

$$\dot{z} = \vartheta h(x) - \frac{\partial \alpha}{\partial x}(x)\,[f(x) + g(x)\,(z + \alpha(x))] + u \tag{6.31b}$$

Constructing the augmented Lyapunov function $V_1(x, z)$:

$$V_1(x, z) = V(x) + \frac{z^2}{2} \tag{6.32}$$

$$\dot{V}_1 = \frac{\partial V}{\partial x}(x)\,[f(x) + g(x)\,(z + \alpha(x))] + z\left[\vartheta h(x) - \frac{\partial \alpha}{\partial x}\,(f(x) + g(x)\,(z + \alpha(x)) + u)\right] \tag{6.33}$$

$$\leq -W(x) + z\left[\vartheta h(x) - \frac{\partial \alpha}{\partial x}(x)\,(f(x) + g(x)\,(z + \alpha(x))) + \frac{\partial V}{\partial x}(x)g(x) + u\right] \tag{6.34}$$

If the parameter ϑ was known then the choice of the control u in (6.35) would ensure that $\dot{V}_1$ is nonpositive.

$$u = -\vartheta h(x) + \frac{\partial \alpha}{\partial x}(x)\,(f(x) + g(x)\,(z + \alpha(x))) - \frac{\partial V}{\partial x}g(x) - \lambda z, \quad \lambda > 0 \tag{6.35}$$

Since ϑ is actually not known the control (6.35) is not feasible making the rendering of the candidate Lyapunov function $V_1(x, z)$ impossible. As an alternative instead of just augmenting $V(x)$ with the quadratic term z^2 consider the estimation error $\tilde{\vartheta} = \vartheta - \hat{\vartheta}$ where $\hat{\vartheta}$ is the estimate for ϑ[73]. Constructing a new candidate Lyapunov function by augmenting a quadratic term of the estimation

error to $V_1(x, z)$.

$$V_{ad}(x, z, \tilde{\vartheta}) = V(x) + \frac{z^2}{2} + \frac{\tilde{\vartheta}^2}{2\gamma}, \gamma > 0 \tag{6.36}$$

$$\begin{aligned}
\dot{V}_{ad} &\leq -W(x) + \frac{\partial V}{\partial x}(x)g(x)z + \frac{\tilde{\vartheta}\dot{\tilde{\vartheta}}}{\gamma} \\
&\quad + z\left[\vartheta h(x) - \frac{\partial \alpha}{\partial x}(x)\left(f(x) + g(x)\left(z + \alpha(x)\right)\right)\right] \qquad (6.37) \\
&\leq -W(x) + z\left[\hat{\vartheta}h(x) - \frac{\partial \alpha}{\partial x}(x)\left(f(x) + g(x)\left(z + \alpha(x)\right)\right) + \frac{\partial V}{\partial x}(x)g(x) + u\right] \\
&\quad + \tilde{\vartheta}zh(x) + \frac{\tilde{\vartheta}\dot{\tilde{\vartheta}}}{\gamma} \qquad (6.38)
\end{aligned}$$

To make $\dot{V}_{ad}$ nonpositive let the control u be given by:

$$u = -\hat{\vartheta}h(x) + \frac{\partial \alpha}{\partial x}(x)\left(f(x) + g(x)\left(z + \alpha(x)\right)\right) - \frac{\partial V}{\partial x}(x)g(x) - \lambda z, \quad \lambda > 0 \tag{6.39}$$

This choice of control leaves $\dot{V}_{ad}$ as :

$$\dot{V}_{ad} \leq -W(x) - \lambda z^2 + \tilde{\vartheta}\left(zh(x) + \frac{\dot{\tilde{\vartheta}}}{\gamma}\right) \tag{6.40}$$

To complete the design of the adaptive backstepping controller an update law is chosen such that the last term of (6.40) is zero.

$$\dot{\tilde{\vartheta}} = -\gamma z h(x) \tag{6.41a}$$

$$\dot{\hat{\vartheta}} = \gamma z h(x) \tag{6.41b}$$

Thus the control and update law for the adaptive backstepping controller are:

$$u = -\hat{\vartheta}h(x) + \frac{\partial \alpha}{\partial x}(x)\left(f(x) + g(x)\left(z + \alpha(x)\right)\right) - \frac{\partial V}{\partial x}(x)g(x) - \lambda z \tag{6.42a}$$

$$\dot{\hat{\vartheta}} = \gamma z h(x) \tag{6.42b}$$

It should easily be seen that the outlined adaptive backstepping technique can

be applied to higher order systems with matched uncertainties by recursive application of the procedure. The effectiveness of adaptive backstepping is however shown in the case where the uncertainties enter the system dynamics one integrator or more before the control, this is the extended matching case. Even though the extended matching case is not encountered in the attitude dynamics it shall be presented for the sake of completeness.

Consider the second order system given by (6.43) where ϑ is an unknown constant scalar.

$$\dot{x} = \vartheta f(x_1) + x_2 \tag{6.43a}$$

$$\dot{x_2} = u \tag{6.43b}$$

Equation (6.43a) contains the unknown parameter ϑ and hence an adaptive controller has to be designed for this subsystem in which x_2 is taken as the pseudo-control. Let $V(x, \tilde{\vartheta}_1)$ be a candidate Lyapunov function where $\tilde{\vartheta}_1$ is the estimation error given by $\tilde{\vartheta}_1 = \vartheta - \hat{\vartheta}_1$.

$$V(x, \tilde{\vartheta}_1) = \frac{x^2}{2} + \frac{\tilde{\vartheta}_1^2}{2\gamma_1} \tag{6.44}$$

Let $x_2 = \alpha(x_1, \hat{\vartheta}_1)$, then the choice of $\alpha(x_1, \hat{\vartheta}_1)$ and the update law in (6.45) will make $\dot{V} = -\lambda_1 x_1^2$ where $\lambda_1 > 0$.

$$\alpha(x_1, \hat{\vartheta}_1) = -\hat{\vartheta}_1 f(x_1) - \lambda_1 x_1 \tag{6.45a}$$

$$\dot{\hat{\vartheta}}_1 = \gamma_1 x_1 f(x_1) \tag{6.45b}$$

Now following the backstepping procedure let the error variable be given as $z = x_2 - \alpha(x_1, \hat{\vartheta}_1)$, the dynamics of the system expressed in terms of (x_1, z) become:

$$\dot{x}_1 = \left(\vartheta - \hat{\vartheta}_1\right) f(x_1) - \lambda_1 x_1 + z_1 \tag{6.46a}$$

$$\dot{z} = u - \frac{\partial \alpha}{\partial x_1} x_2 - \frac{\partial \alpha}{\partial x_1} \vartheta f(x_1) - \frac{\partial \alpha}{\partial \hat{\vartheta}_1} \gamma x_1 f(x_1) \tag{6.46b}$$

Construct a Lyapunov function and design for u to make its derivative nonposi-

tive. Consider the candidate Lyapunov function $V(x_1, \tilde{\vartheta}_1, z)$.

$$V(x_1, \tilde{\vartheta}_1, z) = \frac{x^2}{2} + \frac{\tilde{\vartheta}_1^2}{2} + \frac{z^2}{2} \tag{6.47}$$

$$\begin{aligned} \dot{V} &= x\left[\left(\vartheta - \hat{\vartheta}_1\right) f(x_1) - \lambda_1 x_1 + z\right] + \frac{\tilde{\vartheta}_1 \dot{\tilde{\vartheta}}_1}{\gamma_1} \\ &\quad + z\left[u - \frac{\partial \alpha}{\partial x_1} x_2 - \frac{\partial \alpha}{\partial \hat{\vartheta}_1} \gamma_1 x_1 f(x_1) - \frac{\partial \alpha}{\partial x_1} \vartheta f(x_1)\right] \end{aligned} \tag{6.48}$$

Substituting (6.45b) and also recall that $\tilde{\vartheta}_1 = \vartheta - \hat{\vartheta}_1$ gives:

$$\dot{V} = -\lambda_1 x_1^2 + z\left[u + x_1 - \frac{\partial \alpha}{\partial x_1} x_2 - \frac{\partial \alpha}{\partial \hat{\vartheta}_1} x_1 f(x_1) - \frac{\partial \alpha}{\partial x_1} \vartheta f(x_1)\right] \tag{6.49}$$

From equation (6.49) it can be seen that it is impossible to cancel out the indefinite term $\frac{\partial \alpha}{\partial x_1} \vartheta f(x_1)$. As an alternative consider a different candidate Lyapunov function $V_1(x_1, \tilde{\vartheta}_1, \tilde{\vartheta}_2, z)$ which is constructed by augmenting $V(x_1, \tilde{\vartheta}_1, z)$ with a quadratic term of a second estimation error $\tilde{\vartheta}_2 = \vartheta - \hat{\vartheta}_2$ where $\hat{\vartheta}_2$ is the second estimate of ϑ.

$$\begin{aligned} V_1(x_1, \tilde{\vartheta}_1, \tilde{\vartheta}_2, z) &= V(x_1, \tilde{\vartheta}_1, z) + \frac{\tilde{\vartheta}_2^2}{\gamma_2}, \quad \gamma_1 > 0 \\ &= -\lambda_1 x_1^2 + z\left[u + x_1 - \frac{\partial \alpha}{\partial x_1} x_2 - \frac{\partial \alpha}{\partial \hat{\vartheta}_1} x_1 f(x_1) - \frac{\partial \alpha}{\partial x_1} \vartheta f(x_1)\right] + \frac{\tilde{\vartheta}_2 \dot{\tilde{\vartheta}}_1}{\gamma_2} \\ &= -\lambda_1 x_1^2 + z\left[u + x_1 - \frac{\partial \alpha}{\partial x_1} x_2 - \frac{\partial \alpha}{\partial \hat{\vartheta}_1} x_1 f(x_1) - \left(\tilde{\vartheta}_2 + \hat{\vartheta}_2\right) f(x_1) \frac{\partial \alpha}{\partial x_1}\right] + \frac{\tilde{\vartheta}_2 \dot{\tilde{\vartheta}}_2}{\gamma_2} \\ &= -\lambda_1 x_1^2 + z\left[u + x_1 - \frac{\partial \alpha}{\partial x_1} x_2 - \frac{\partial \alpha}{\partial \hat{\vartheta}_1} x_1 f_1(x_1) - \hat{\vartheta}_2 f(x_1) \frac{\partial \alpha}{\partial x_1}\right] \\ &\quad - \tilde{\vartheta}_2 \left(f(x_1) \frac{\partial \alpha}{\partial x_1} - \frac{\dot{\tilde{\vartheta}}_2}{\gamma_2}\right) \end{aligned} \tag{6.50}$$

Now eliminating the unknown term with the update law:

$$\dot{\hat{\vartheta}}_2 = \gamma_2 f(x_1) \frac{\partial \alpha}{\partial x_1} \tag{6.51}$$

The control law and the update laws for the adaptive backstepping controller for

the extended matching case are given by:

$$u = -x_1 + \frac{\partial \alpha}{\partial x_1} x_2 + \frac{\partial \alpha}{\partial \hat{\vartheta}_1} x_1 f(x_1) + \hat{\vartheta}_2 f(x_1) \frac{\partial \alpha}{\partial x_1} - \lambda_2 z, \quad \lambda_2 > 0 \tag{6.52a}$$

$$\dot{\hat{\vartheta}}_1 = \gamma_1 x_1 f(x_1) \tag{6.52b}$$

$$\dot{\hat{\vartheta}}_2 = \gamma_2 f(x_1) \frac{\partial \alpha}{\partial x_1} \tag{6.52c}$$

This choice of control law and parameters update laws results in $\dot{V}_1 = -\lambda_1 x_1^2 - \lambda_2 z^2$. Applying La Salle's theorem it can be seen that $x_1, z \to 0$ as $t \to \infty$ since $z = x_2 - \alpha(x, \hat{\vartheta}_1)$ this implies that $x_2 \to \alpha(0, \hat{\vartheta}_1)$ as $t \to \infty$. It can also be seen that the presented controller requires two estimates for the single unknown parameter, this over-parameterisation is characteristic of this technique as such the technique is also called the over-parameterised adaptive backstepping scheme[73]. By recursively applying this technique one can stabilise systems that belong to the class of parametric strict-feedback systems of the form:

$$\begin{aligned}
\dot{x}_1 &= x_2 + f_1(x_1)\vartheta_1 \\
\dot{x}_2 &= x_3 + f_2(x_1, x_2)\vartheta_2 \\
&\vdots \\
\dot{x}_{n-1} &= x_{n-2} + f_{n-1}(x_1, \ldots, x_{n-1})\vartheta_{n-1} \\
\dot{x}_n &= g(x_1, \ldots, x_n)u + f_n(x_1, \ldots, x_n)\vartheta_n
\end{aligned} \tag{6.53}$$

For a n^{th} order system with p unknown parameters the over parameterised adaptive backstepping scheme may require as many as pn parameter estimates which becomes problematic for large systems. It is because of this drawback that Kristic, Kanellakopoulos and Kokotovic[59] introduced the tuning function method which eliminates the over-parameterisation. The details of the tuning function method are not covered here however the interested reader can consult [73] for a detailed account.

Example. *Consider the second order system given by* (6.54) *where* ϑ *and* φ *are*

unknown scalar parameters.

$$\dot{x} = \vartheta cosx - x^3 + \xi \tag{6.54a}$$

$$\dot{\xi} = \varphi x^2 + u \tag{6.54b}$$

The task is to regulate the system about $x = 0$.

Firstly an adaptive controller is designed for (6.54a) by choosing the candidate Lyapunov function $V(x, \tilde{\vartheta}_1)$ where $\tilde{\vartheta}_1 = \vartheta - \hat{\vartheta}_1$ with $\hat{\vartheta}_1$ being the first estimate of ϑ.

$$V(x, \tilde{\vartheta}_1) = \frac{x^2}{2} + \frac{\tilde{\vartheta}_1^2}{2\gamma_1}, \quad \gamma_1 > 0 \tag{6.55}$$

The function $\alpha(x, \hat{\vartheta}_1)$ and the parameter update law is chosen such that if $\xi = \alpha(x, \hat{\vartheta}_1)$ the derivative of (6.55) would be nonpositive.

$$\alpha(x, \hat{\vartheta}_1) = -\hat{\vartheta}_1 cosx \tag{6.56a}$$

$$\dot{\hat{\vartheta}}_1 = \gamma_1 x cosx \tag{6.56b}$$

Let $z = \xi - \alpha(x, \hat{\vartheta}_1)$ be the error variable then the system dynamics are expressed as:

$$\dot{x} = \left(\vartheta - \hat{\vartheta}_1\right) cosx - x^3 + z \tag{6.57a}$$

$$\dot{z} = \varphi x^2 - \left(\vartheta cosx + \xi - x^3\right) \frac{\partial \alpha}{\partial x} - \gamma_1 x cosx \frac{\partial \alpha}{\partial \hat{\vartheta}_1} + u \tag{6.57b}$$

Select a u that ensures that the derivative of some Lyapunov function is nonpositive. For the Lyapunov function consider the function:

$$V_1(x, z, \tilde{\vartheta}_1, \tilde{\vartheta}_2, \tilde{\varphi}) = \frac{x^2}{2} + \frac{z^2}{2} + \frac{\tilde{\vartheta}_1^2}{2\gamma_1} + \frac{\tilde{\vartheta}_2^2}{2\gamma_2} + \frac{\tilde{\varphi}^2}{2\gamma_3}, \quad \gamma_2 > 0, \quad \gamma_3 > 0 \tag{6.58}$$

where $\tilde{\vartheta}_2$ is the estimation error ($\tilde{\vartheta}_2 = \vartheta - \hat{\vartheta}_2$) with $\hat{\vartheta}_2$ being the second estimate of ϑ and $\tilde{\varphi}$ is the estimation error for φ given by $\tilde{\varphi} = \varphi - \hat{\varphi}$ with $\hat{\varphi}$ being the

estimate. The derivative of the candidate Lyapunov function is:

$$\begin{aligned}\dot{V}_1 &= x\left[\left(\vartheta-\hat{\vartheta}_1\right)cosx+z-x^3\right]+z\left[\varphi x^2-\left(\vartheta cosxz+\xi-x^3\right)\frac{\partial\alpha}{\partial x}-\gamma_1 xcosx\frac{\partial\alpha}{\partial\hat{\vartheta}_1}+u\right]\\&+\frac{\tilde{\vartheta}_1\dot{\tilde{\vartheta}}_1}{\gamma_1}+\frac{\tilde{\vartheta}_2\dot{\tilde{\vartheta}}_2}{\gamma_2}+\frac{\tilde{\varphi}\dot{\tilde{\varphi}}}{\gamma_3}\\&=-x^4+z\left[x+\hat{\vartheta}_2x^2-\left(\hat{\vartheta}_2cosx+\xi-x^3\right)\frac{\partial\alpha}{\partial x}-\gamma_1 xcosx\frac{\partial\alpha}{\partial\hat{\vartheta}_1}+u\right]\\&-\tilde{\vartheta}_2\left(zcosx\frac{\partial\alpha}{\partial x}-\frac{\dot{\hat{\vartheta}}_2}{\gamma_2}\right)+\tilde{\varphi}\left(zx^2+\frac{\dot{\tilde{\varphi}}}{\gamma_3}\right)\end{aligned}\tag{6.59}$$

Choosing the control law and parameter update laws as:

$$u=-x-\hat{\varphi}_2x^2+\left(\hat{\vartheta}_2cosx+\xi-x^3\right)\frac{\partial\alpha}{\partial x}+\gamma_1 xcosx\frac{\partial\alpha}{\partial\hat{\vartheta}_1}-\lambda z \tag{6.60a}$$

$$\dot{\hat{\vartheta}}_1=\gamma_1 xcosx \tag{6.60b}$$

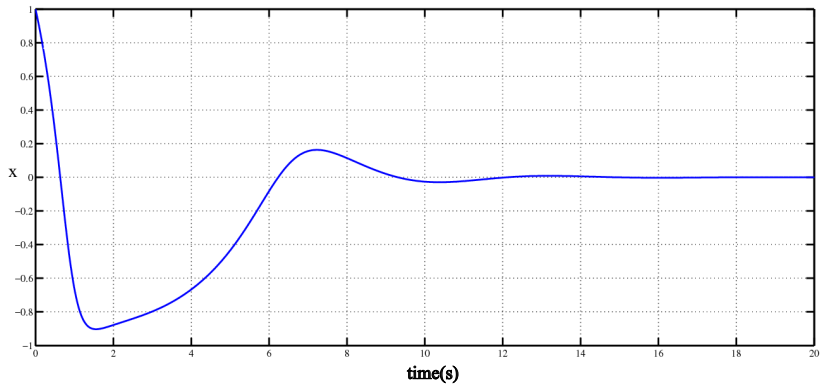
$$\dot{\hat{\vartheta}}_2=-\gamma_2 zcosx\frac{\partial\alpha}{\partial x} \tag{6.60c}$$

$$\dot{\hat{\varphi}}=\gamma_3 zx^2 \tag{6.60d}$$

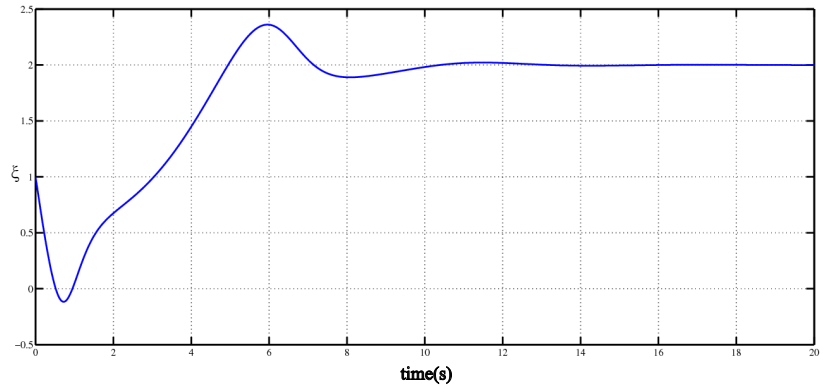
The controller was simulated in the MATLAB/SIMULINK environment in which the true values of the unknown parameters were $\vartheta=-2$ and $\varphi=2$. The following values for the controller parameters where used. The system response shown in

$$\begin{array}{ll}\gamma_1=1 & \gamma_2=2\\ \gamma_3=3 & \lambda=3\end{array}$$

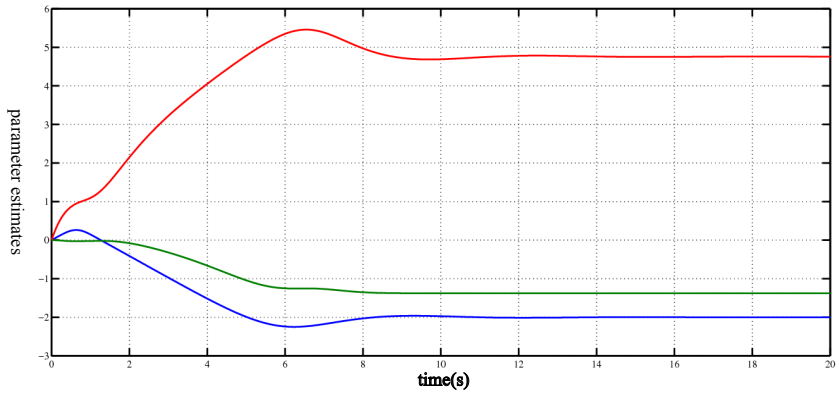
figure 6.4 corresponds to the initial conditions ($x(0)=1$, $\xi(0)=1$).



(a) x plot



(b) ξ plot



(c) Parameter adaptation: blue = $\hat{\vartheta}_1$, green = $\hat{\vartheta}_2$, red = $\hat{\varphi}$

Figure 6.4: Adaptive Backstepping example simulation results

6.4 Nonlinear Damping

Nonlinear damping is a control technique that achieves bounded response when the system is perturbed by a bounded disturbance. The major advantage of this technique is that no knowledge of the upper bounds of the disturbance are required. This technique can be traced back to the work of Feuer and Morse [74] it was then explicitly developed by Barmish et al[75] and Sontag[76].

To illustrate how nonlinear damping works consider the general nonlinear system (6.61).

$$\dot{x} = f(x) + g(x)u + \Gamma(x)\Delta(t) \tag{6.61}$$

where $f(x)$, $g(x)$ and $\Gamma(x)$ are continuous functions also $g(0) \neq 0, f(0) = 0$ and $\Delta(t)$ is a bounded function. Consider the nominal version of (6.61), assume that there exists a function $\alpha(x)$ which stabilises the origin of the nominal system with Lyapunov function $V(x)$ such that the inequality (6.62) is satisfied.

$$\frac{\partial V}{\partial x}(x)\left(f(x) + g(x)\alpha(x)\right) \leq -W(x) \tag{6.62}$$

where $W(x)$ is a positive definite function. Now further consider the derivative of $V(x)$ along the solutions of the actual system (6.61):

$$\dot{V} = \frac{\partial V}{\partial x}(x)\left(f(x) + g(x)u + \Gamma(x)\Delta(t)\right) \tag{6.63}$$

Let $u = \alpha(x) + \nu(x)$ which gives:

$$\begin{aligned}\dot{V} &= \frac{\partial V}{\partial x}(x)\left(f(x) + g(x)\alpha(x)\right) + \frac{\partial V}{\partial x}(x)g(x)\nu(x) + \frac{\partial V}{\partial x}(x)\Gamma(x)\Delta(t) \\ &\leq -W(x) + \frac{\partial V}{\partial x}(x)\left(g(x)\nu(x) + \Gamma(x)\Delta(t)\right)\end{aligned} \tag{6.64}$$

It is required that $\dot{V}$ be negative definite outside some compact region so as to achieve boundedness, choosing $\nu(x) = -\frac{K}{g(x)}\left(\Gamma(x)\right)^2 \frac{\partial V}{\partial x}(x)$ where K is a positive

constant.

$$\begin{aligned}\dot{V} &\leq -W(x) - K\left(\Gamma(x)\right)^2\left(\frac{\partial V}{\partial x}\right)^2 + \Delta(t)\Gamma(x)\frac{\partial V}{\partial x}(x) \\ &= -W(x) - K\left[\left(\Gamma(x)\frac{\partial V}{\partial x}(x) - \frac{\Delta(t)}{2K}\right)^2 - \frac{\Delta^2(t)}{4K^2}\right] \text{ completing the square} \\ &= -W(x) - K\left(\Gamma(x)\frac{\partial V}{\partial x}(x) - \frac{\Delta(x)}{2K}\right)^2 + \frac{\Delta^2(t)}{4K} \end{aligned} \tag{6.65}$$

Thus $\dot{V}$ will be negative definite outside the compact set R defined as:

$$R = \left\{x : \left|W(x) + K\left(\Gamma(x)\frac{\partial V}{\partial x}(x) - \frac{\|\Delta\|_\infty}{2K}\right)^2\right| > \frac{\|\Delta\|_\infty}{4K}\right\} \tag{6.66}$$

Thus $x(t)$ will be bounded to the set R. It is interesting to note that $x(t)$ will be bounded despite of the size of the gain K that is chosen, the set R is finite as long as $\|\Delta\|_\infty$ is finite however the size of the set R can be made arbitrarily small by selecting the right K.

The action of the nonlinear term $\nu(x)$ can be viewed as a dynamic gain which is large when $\Gamma(x)$ is large and correspondingly small for small $\Gamma(x)$. $\Gamma(x)$ multiplies the disturbance $\Delta(t)$ as such it is some kind of a disturbance gain thus the nonlinear damping term acts to reduce the effective disturbance gain.

6.5 Attitude Control

Having developed the tools needed to achieve robust system performance the next step is to design the attitude control. For convenience the attitude kinematic and dynamic equations are restated.

$$\dot{\mathbf{\Theta}} = \mathbf{\Psi}(\mathbf{\Theta})\omega^B \tag{6.67a}$$

$$\dot{\mathbf{R}}\mathbf{e}_3 = \mathbf{R}\hat{\omega}^B\mathbf{e}_3 \tag{6.67b}$$

$$\mathbf{I}\dot{\omega}^B = -\omega^B \times \left(\mathbf{I}\omega^B\right) + \tau^B \tag{6.67c}$$

In chapter 5 a translational controller was developed which generates a desired orientation in the form of the unit vector Re_{3d}. It is the task of the attitude controller to therefore ensure asymptotic tracking of this reference vector. Also the trajectory of the quadrotor UAV is defined as being specified by the pair $(\mathbf{p}_d(x, y, z),\ \psi(t))$ as such a yaw controller is required that ensures asymptotic tracking of the reference yaw. Thus the attitude controller will comprise two sub-controllers: the orientation controller and the yaw controller.

6.5.1 Orientation Controller

Firstly the attitude dynamics in (6.67) are expressed in their expanded form. Let $R = [\rho_{ij}]$ where ρ_{ij} is the element in the i^{th} row and j^{th} column.

$$\begin{bmatrix} \dot{\rho_{13}} \\ \dot{\rho_{23}} \\ \dot{\rho_{33}} \end{bmatrix} = \begin{bmatrix} -\rho_{12} & \rho_{11} \\ -\rho_{22} & \rho_{21} \\ -\rho_{32} & \rho_{31} \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} \tag{6.68a}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_\phi qr \\ I_\theta pr \\ I_\psi pq \end{bmatrix} + \begin{bmatrix} \Delta_\phi(t) + \frac{1}{I_{xx}}\tau_\phi \\ \Delta_\theta(t) + \frac{1}{I_{yy}}\tau_\theta \\ \Delta_\psi(t) + \frac{1}{I_{zz}}\tau_\psi \end{bmatrix} \tag{6.68b}$$

where:

$$\begin{aligned} I_\phi &= \tfrac{I_{yy}-I_{zz}}{I_{xx}} \\ I_\theta &= \tfrac{I_{zz}-I_{xx}}{I_{yy}} \\ I_\phi &= \tfrac{I_{xx}-I_{yy}}{I_{zz}} \end{aligned}$$

Recall that since Re_3 is known to be a unit vector it only requires two of its elements to be fully determined, thus only the equations related to the elements ρ_{13} and ρ_{23} are used. However only a detailed presentation of the design for the controller of the ρ_{13} dynamics is presented because of the symmetry of the problem. Let $\frac{1}{I_{xx}} = 1 + \frac{1-I_{xx}}{I_{xx}}$ then the dynamics of ρ_{13} can be expressed as

$$\dot{\rho}_{13} = q\rho_{11} - p\rho_{21} \tag{6.69a}$$

$$\dot{p} = I_\phi qr + \delta_\phi + \tau_\phi \tag{6.69b}$$

where $\delta_\phi(t) = \Delta_\phi + \frac{1-I_{xx}}{I_{xx}}\tau_\phi$ is an unknown bounded function. The task is to design a control τ_ϕ to ensure that ρ_{13} tracks some reference signal $\rho_{13d}(t)$. To that end let the tracking error variable be $e_{\rho_{13}} = \rho_{13} - \rho_{13d}$ and its integral $\chi_\phi = e_{\rho_{13}}$ such that now the system dynamics are given by:

$$\dot{\chi} = e_{\rho_{13}} \tag{6.70a}$$

$$\dot{e}_{\rho_{13}} = -\dot{\rho}_{31d} + q\rho_{11} - \rho_{21}p \tag{6.70b}$$

$$\dot{p} = I_\phi qr + \delta_\phi(t) + \tau_\phi \tag{6.70c}$$

Taking $e_{\rho_{13}}$ as the pseudo-control for (6.70a) let the candidate Lyapunov function be given by $V_1(\chi) = \frac{\chi^2}{2}$. It should be easy to see that if $e_{\rho_{13}} = \alpha_1(\chi) = -\lambda_1\chi$ where $\lambda_1 > 0$ then the derivative of V_1 will be rendered negative definite. Following the backstepping procedure the error variable is given as $z_1 = e_{\rho_{13}} - \alpha_1(\chi)$ which gives the (χ, z_1) dynamics :

$$\dot{\chi} = -\lambda_1\chi + z_1 \tag{6.71}$$

$$\dot{z_1} = -\dot{\rho_{13d}} + q\rho_{11} + \lambda_1 z_1 - \lambda_1^2\chi - \rho_{21}p \tag{6.72}$$

Taking p as the pseudo-control for the (χ,z_1) system and choosing the candidate Lyapunov function $V_2(\chi, z_1)$ such that:

$$V_2(\chi, z_1) = \frac{\chi^2}{2} + \frac{z_1^2}{2} \tag{6.73}$$

It should be easy to verify that the function α_2 makes the time derivative of $V_2(\chi, z)$ negative definite if $p = \alpha_2$.

$$\alpha_2 = \frac{1}{\rho_{21}}\left[-\dot{\rho}_{13d} + \chi + q\rho_{11} + \lambda 1 z_1 - \lambda_1^2\chi + \lambda_2 z_1\right], \quad \lambda_2 > 0 \tag{6.74}$$

This choice of pseudo-control makes $\dot{V}_2 = -\lambda_1\chi - \lambda_2 z_1^2$. Defining the new error

variable $z_2 = p - \alpha_2$ the dynamics of the (χ, z_1, z_2) system becomes:

$$\dot{\chi} = -\lambda_1 \chi + z_1 \tag{6.75}$$

$$\dot{z}_1 = -\chi - \lambda_2 z_1 - \rho_{21} z_2 \tag{6.76}$$

$$\dot{z}_2 = I_\phi qr - (z_1 - \lambda_1 \chi) \frac{\partial \alpha_2}{\partial \chi} - (-\chi - \lambda_1 z_1 - \rho_{21} z_2) \frac{\partial \alpha_2}{\partial z_1} + \delta_\phi(t) + \tau_\phi \tag{6.77}$$

The dynamics of the (χ, z_1, z_2) system contains uncertain parameters(I_ϕ) and a disturbance input($\delta(t)$). As such the techniques that have been developed in this chapter(i.e. adaptive backstepping and nonlinear damping) shall be made use of to stabilize the system. Consider the candidate Lyapunov function $V_3(\chi, z_1, z_2, \tilde{I}_\phi)$ with $\tilde{I}_\phi = I_\phi - \hat{I}_\phi$ where $\hat{I}_\phi$ is the estimate and $\tilde{I}_\phi$ is the estimation error.

$$V_3(\chi, z_1, z_2) = \frac{\chi^2}{2} + \frac{z_1^2}{2} + \frac{z_2^2}{2} + \frac{\tilde{I}_\phi}{2\gamma_1} \tag{6.78}$$

$$\begin{aligned} \dot{V}_3 = &-\lambda_1 \chi^2 - \lambda_2 z_2 + z_2 qr \tilde{I}_\phi + \frac{\tilde{I}_\phi \dot{\tilde{I}}_\phi}{\gamma_1} \\ &+ z_2 \left[-\rho_{21} z_1 + \hat{I}_\phi qr - (z_1 - \lambda_1 \chi) \frac{\partial \alpha_2}{\partial \chi} + (\chi + \lambda_1 z_1 + \rho_{21} z_2) \frac{\partial \alpha_2}{\partial z_1} + \delta_\phi(t) \right] \end{aligned} \tag{6.79}$$

Consider the nominal control($\tau_{\phi nom}$) and update law which would have made $\dot{V}_3$ negative definite for the nominal case :

$$\tau_{\phi nom} = \rho_{21} z_1 - \hat{I} qr + (z_1 - \lambda_1 \chi) \frac{\partial \alpha_2}{\partial \chi} - (\chi + \lambda_1 z_1 + \rho_{21} z_2) \frac{\partial \alpha_2}{\partial z_1} - \lambda_3 z_2, \quad \lambda_3 > 0 \tag{6.80a}$$

$$\dot{\hat{I}}_\phi = \gamma z_2 qr \tag{6.80b}$$

Let the actual control be $\tau_\phi = \tau_{\phi nom} + \tau_{ND}$ where $\tau_{ND} = -K z_2$ substituting this

and the update law into (6.79) gives:

$$\begin{aligned}
\dot{V}_3 &= -\lambda_1\chi^2 - \lambda_2 z_1^2 - \lambda_3 z_2^2 + z_2\left[\tau_{ND} + \delta_\phi(t)\right] \\
&= -\lambda_1\chi^2 - \lambda_2 z_1^2 - \lambda_3 z_2^2 - K z_2^2 + z_2\delta_\phi(t), \quad K > 0 \\
&= -\lambda_1\chi^2 - \lambda_2 z_1^2 - \lambda_3 z_2^2 - K\left(z_2^2 - \frac{z_2\delta_\phi(t)}{K}\right) \\
&= -\lambda_1\chi^2 - \lambda_2 z_1^2 - \lambda_2 z_2^2 - K\left(z_2 - \frac{\delta_\phi(t)}{2K}\right)^2 + \frac{\delta_\phi^2(t)}{4K} \\
&\leq -\lambda_1\chi^2 - \lambda_2 z_1^2 - \lambda_2 z_2^2 + \frac{\delta_\phi^2(t)}{4K}
\end{aligned} \tag{6.81}$$

From (6.81) $\dot{V}_3$ will be negative outside the ellipsoidal ball R in the (χ, z_1, z_2) space where R is given by:

$$R = \left\{\chi, z_1, z_2 | \lambda_1\chi + \lambda_2 z_1^2 + \lambda_3 z_2^2 \leq \frac{\|\delta_\phi(t)\|_\infty^2}{4K}\right\} \tag{6.82}$$

Thus from (6.82) it can be seen that the size of the ellipsoid R can be decreased by the right selection of the controller gains. One simple case is to make $\lambda_1 = \lambda_2 = \lambda_3$ and then just increase the value K. A similar approach can be applied to the dynamics of ρ_{23} so as to design for the control torque τ_θ.

6.5.2 Yaw Control

The equations of the yaw dynamics are obtained from equations (6.67a) and (6.68b). Let $\psi_d(t)$ be the desired yaw, the yaw tracking error is given by $e_\psi = \psi - \psi_d$. Also let the integral of the tracking error be $\chi_\psi = \int e_\psi$. Thus the yaw dynamics are given as:

$$\dot{\chi}_\psi = e_\psi \tag{6.83a}$$

$$\dot{e}_\psi = -\dot{\psi}_d + \frac{sin\phi}{cos\theta}q + \frac{cos\phi}{cos\theta}r \tag{6.83b}$$

$$\dot{r} = I_\psi pq + \delta_\psi(t) + \tau_\psi \tag{6.83c}$$

The design of the τ_ψ control for the yaw dynamics does not differ much from that of τ_ϕ which has already been outlined, as such a summary of the major points of the design procedure is given. Constructing the candidate Lyapunov function for (6.83a) $V_1(\chi_\psi) = \frac{\chi_\psi^2}{2}$ and choosing e_ψ and r as the pseudo-controls applying the backstepping procedure gives the stabilising functions:

$$\begin{aligned}
\alpha_1(\chi_\psi) &= -\lambda_1 \chi_\psi, \qquad z_1 = e_\psi - \alpha_1(\chi_\psi) \\
\alpha_2(\chi, z_1) &= \frac{cos\theta}{cos\phi}\left[\dot{\psi}_d - \frac{sin\phi}{cos\theta}q - \lambda_1 z - 1 + \lambda_1^2\chi_\psi - \chi_\psi - \lambda_2 z_2\right] \\
z_2 &= r - \alpha_2(\chi_\psi, z_1)
\end{aligned}$$

From applying the adaptive backstepping and nonlinear damping procedure the following control and parameter update laws are derived:

$$\dot{\hat{I}}_\psi = \gamma_1 z_2 pq \tag{6.84a}$$

$$\tau_\psi = -\frac{cos\phi}{cos\theta}z_1 - \hat{I}_\psi pq + (-\lambda_1\chi_\psi + z_1)\frac{\partial\alpha_2}{\partial\chi_\psi} + \left(-\chi_\psi - \lambda_2 z_1 + \frac{cos\phi}{cos\theta}z_2\right)\frac{\partial\alpha_2}{\partial z_1} - \lambda_3 z_2 - Kz_2 \tag{6.84b}$$

As was the case for the ρ_{31} controller the yaw controller will ensure that the states (χ_ψ, z_1, z_2) will be bounded to the ellipsoidal ball R_ψ defined as:

$$R = \left\{\chi_\psi, z_1, z_2 \;\middle|\; \lambda_1\chi_\psi + \lambda_2 z_1^2 + \lambda_3 z_2^2 \leq \frac{\|\delta_\psi(t)\|_\infty^2}{4K}\right\} \tag{6.85}$$

6.5.3 Simulation Results

The attitude controller that has been developed in this chapter was simulated in the MATLAB/SIMULINK environment to verify its performance. Two scenarios were simulated in which the attitude system was required to:

1. track a time varying reference attitude ($\phi_d(t) = 0.2sin(\pi t)$, $\theta_d(t) = 0.2sin(0.5\pi t)$, $\psi_d = 0$) in the absence of disturbances

2. track a time varying reference attitude ($\phi_d(t) = 0.2sin(\pi t)$, $\theta_d(t) = 0.2sin(0.5\pi t)$, $\psi_d = 0$) in the presence of disturbances ($\delta_\phi(t) = 0.1sin(2\pi t)$, $\delta_\theta(t) =$

$0.1sin(5\pi t)$, $\delta_\psi(t) = 0.1sin(7\pi t)$)

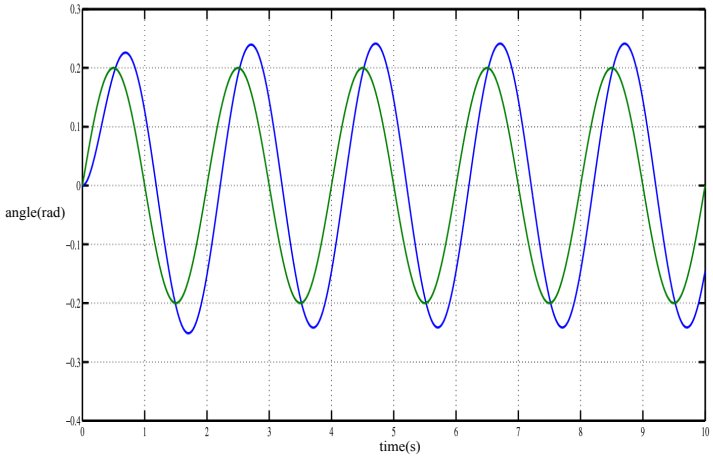
The gains that were chosen for the controller are:

ϕ controller	θ controller	ψ controller
$\lambda_1 = 5$	$\lambda_1 = 10$	$\lambda_1 = 4$
$\lambda_2 = 10$	$\lambda_2 = 9$	$\lambda_2 = 8$
$\lambda_3 = 4$	$\lambda_3 = 8$	$\lambda_3 = 4$
K = 5	K = 5	K = 5

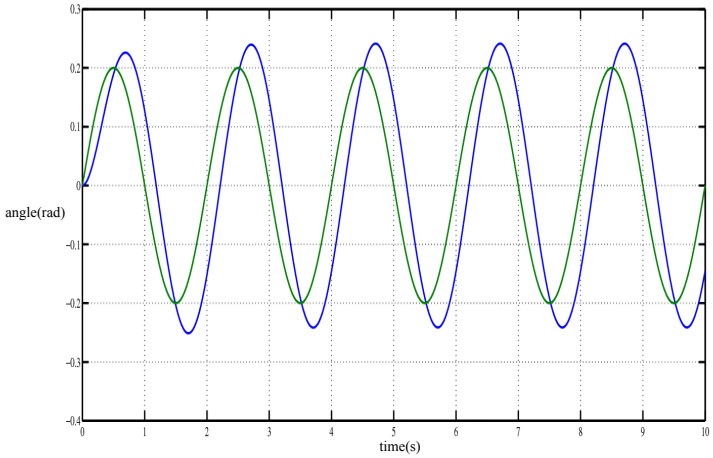
It should be noted that these gains were chosen by trial and error.The results of these simulations are shown in figures 6.5 - 6.9 where it should be clear that the controller does achieve bounded error trajectory tracking for the attitude subsystem.

6.6 Conclusion

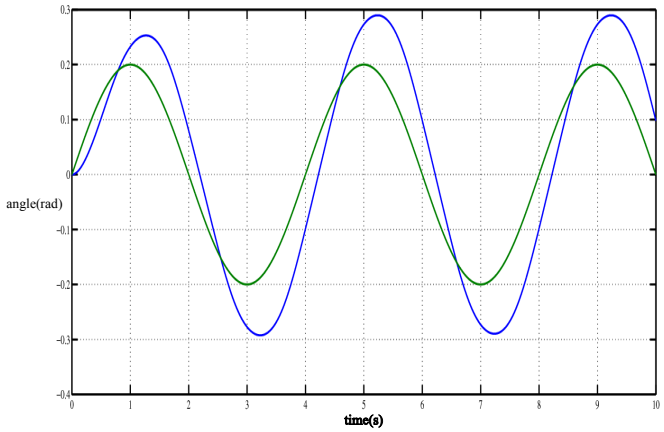
The backstepping technique for the control of nonlinear systems in strict feedback systems has been presented. The backstepping technique was developed as a tool for constructing Lyapunov functions for systems in which the designer has knowledge of a Lyapunov function for a simpler subsystem and uses it to construct a Lyapunov function for the whole system. Using adaptive techniques a modified backstepping technique is presented which can be employed in parametric-strict feedback systems with the parameters being unknown. Further robustness is achieved by adding a nonlinear damping term to the control which guarantees bounded error tracking in the presence of a bounded disturbance without the need for knowledge of the disturbance's upper bounds. Finally simulation results for the attitude control system are presented to demonstrate the feasibility of the controller developed in this chapter.



(a) ϕ plot without disturbance, green = ϕ_d, blue = actual trajectory

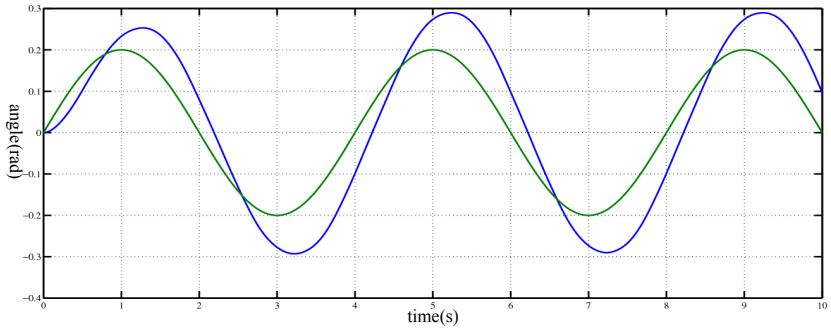


(b) ϕ plot with disturbance, green = ϕ_d, blue = actual trajectory

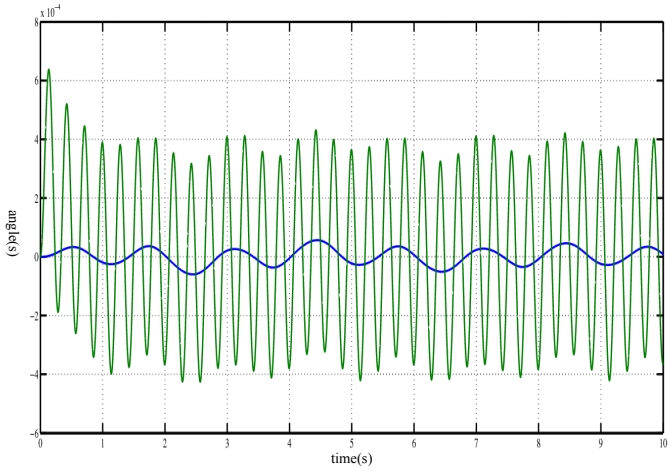


(c) θ plot without disturbance

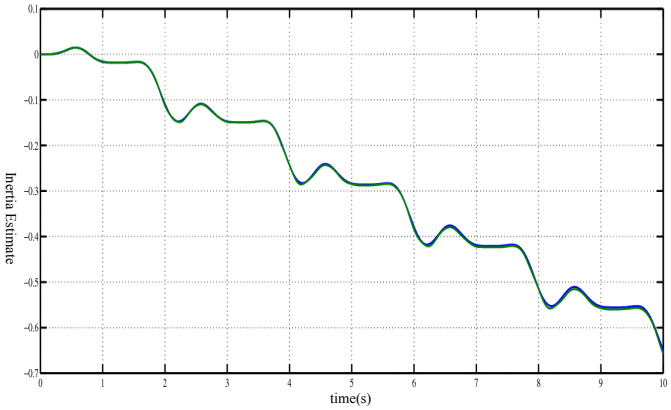
Figure 6.5: Reference signal $\phi_d = 0.2sin(\pi t)$, $\theta_d = 0.2sin(0.5\pi t)$, $\delta_\phi(t) = 0.1sin(2\pi t)$



(a) θ plot with disturbance, green = θ_d, blue = actual trajectory

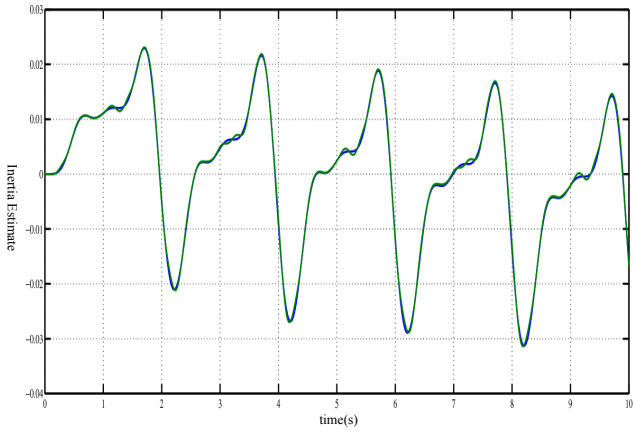


(b) ψ plot, with disturbance = green, without disturbance = blue

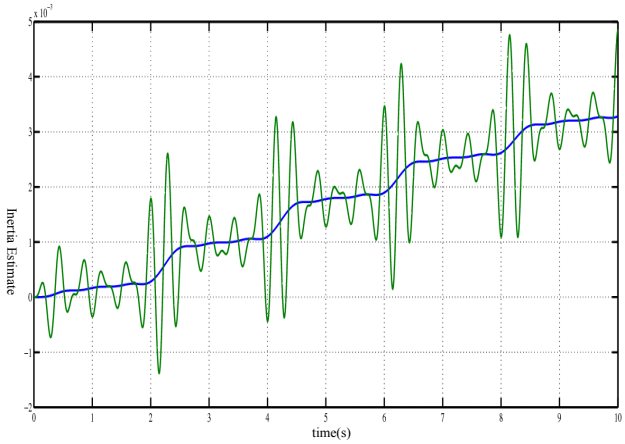


(c) $\hat{I}_\phi$ plot, with disturbance = green, without disturbance = blue

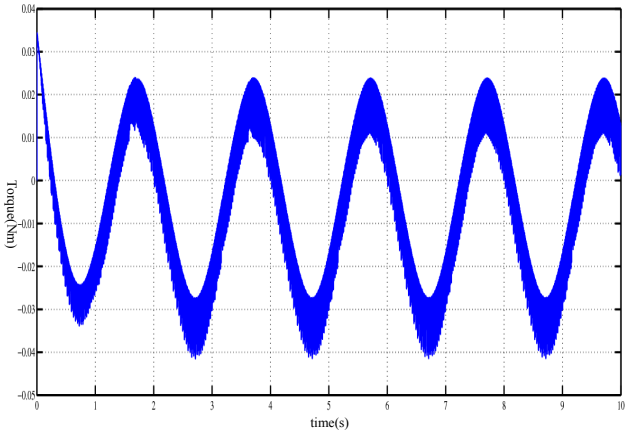
Figure 6.6: Reference signal $\theta_d = 0.2sin(0.5\pi t)$, $\psi_d = 0$, $\delta_\theta(t) = 0.1sin(5\pi t)$, $\delta_\psi(t) = 0.1sin(7\pi t)$



(a) $\hat{I}_\theta$ plot, with disturbance = green, without disturbance = blue

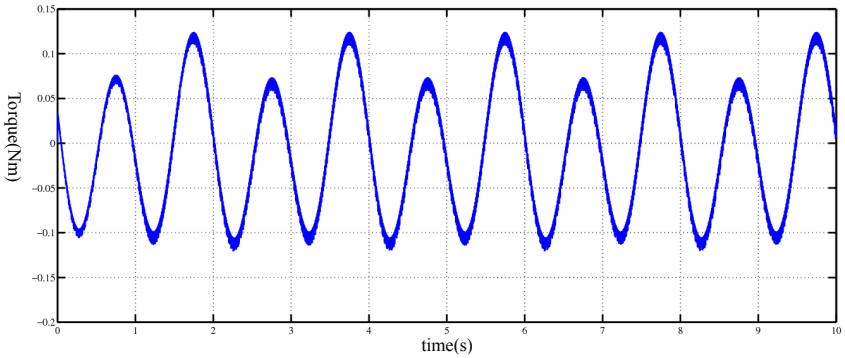


(b) $\hat{I}_\psi$ plot, with disturbance = green, without disturbance = blue

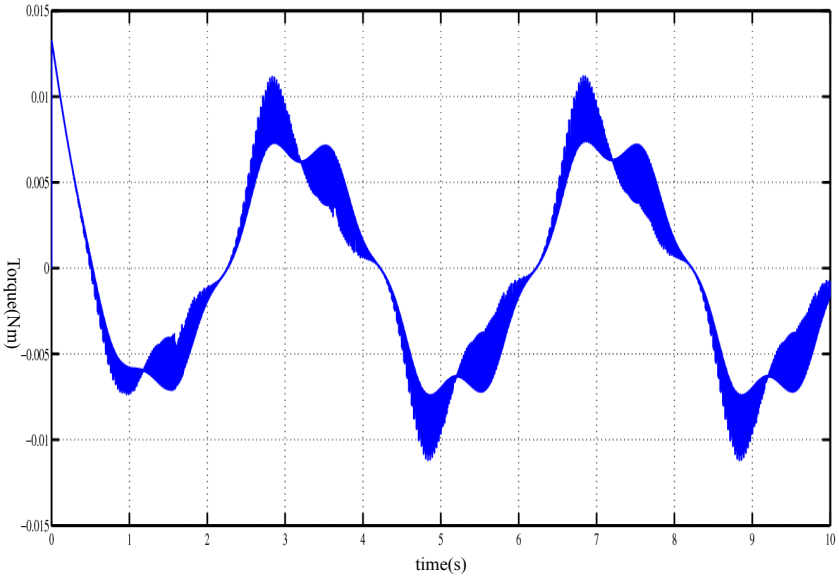


(c) τ_ϕ plot without disturbance

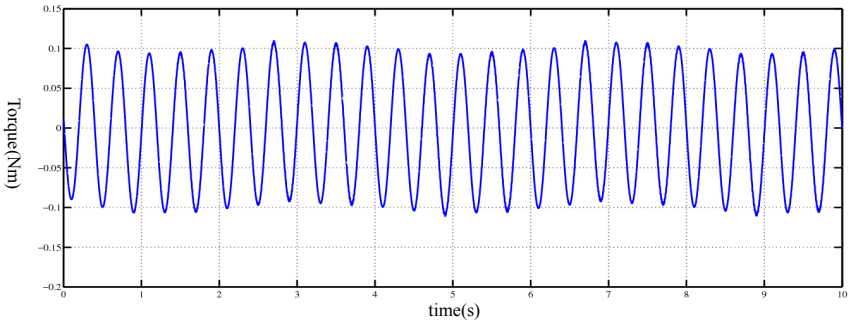
Figure 6.7: $\hat{I}_\theta$, $\hat{I}_\psi$ and τ_ϕ plots



(a) τ_ϕ plot, with disturbance

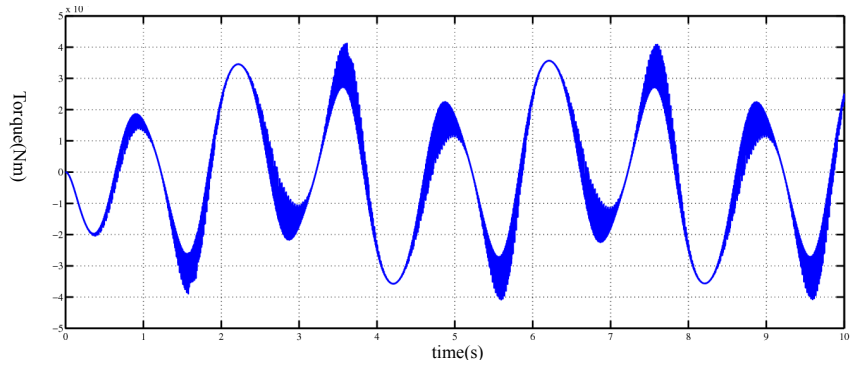


(b) τ_θ plot without disturbance

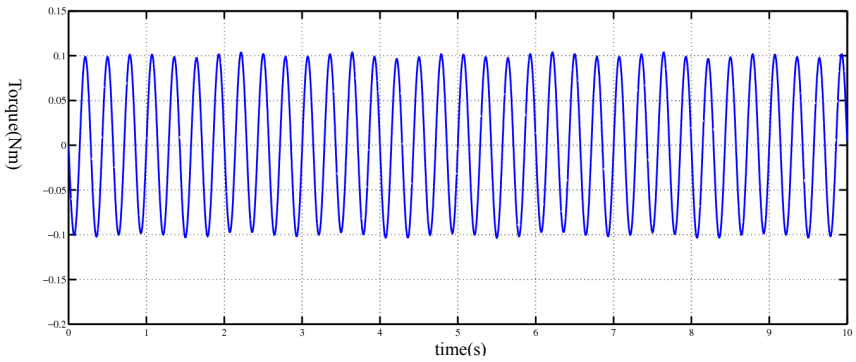


(c) τ_θ plot with disturbance

Figure 6.8: τ_θ and τ_ϕ plots



(a) τ_ψ plot, without disturbance



(b) τ_ψ plot with disturbance

Figure 6.9: τ_ψ plot

Chapter 7

Attitude Control 2

7.1 Chapter Overview

In this chapter a second attitude controller is developed for tracking the reference attitude generated by the translational controller. The attitude controller developed in the previous chapter could only ensure the bounded error tracking of the reference attitude in the presence of disturbances. Thus in this chapter the task is to develop a controller that can theoretically guarantee asymptotic tracking of the reference attitude in the presence of disturbances. To accomplish this the backstepping method is modified by combining it with sliding mode control. As highlighted in section 2.3.2 sliding backstepping control has the disadvantage of requiring *a priori* knowledge of the upper bounds of the uncertainties. In an effort to mitigate this problem an adaptive sliding backstepping approach is considered in which the sliding gain is no longer static but is varied according to the sliding manifold error. Lyapunov based analysis shows that the adaptive sliding backstepping controller is able to guarantee asymptotic tracking even in the presence of unknown disturbances. The work developed in this chapter will be presented at the IFAC World Congress 2014[**?**]Tinashe2

7.2 Sliding Mode Control

Sliding mode control has its roots in the work of Russian researchers led by S.V.Emelyanov[77] in the 1950s. The research at that time was mainly concerned with variable structure control. The notion of discontinuous controls was not a new phenomenon as researchers had come across it while solving optimal control problems(i.e bang-bang control) but then it had been viewed as a nuisance that needed to be eliminated. However it was the Russian mathematician A.F. Filipov in his 1961 seminal paper[78] that provided the mathematical foundation for what is now called sliding mode control.

The basic idea behind sliding mode control is the notion that 1^{st} order systems are easier to control compared to general n^{th} order systems[1]. Thus in sliding mode control the requirement is to drive the system to a manifold in the state-space defined by stable 1^{st} order dynamics, strictly speaking it is when the system reaches this manifold that sliding mode occurs. Sliding mode control has desirable robustness characteristics since the dynamics of the system on the sliding manifold are independent of any plant parameters.

Consider the second order system(7.1) in which $f(\mathbf{x})$ is an unknown but bounded function.

$$\ddot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u \tag{7.1}$$

Assume that the uncertainty can be expressed as $f(\mathbf{x}) = f_{nom}(\mathbf{x}) + \hat{f}(\mathbf{x})$ where $f_{nom}(\mathbf{x})$ is the nominal value and $|\hat{f}(\mathbf{x})| < F$ is the unknown part with upper bound F. The requirement is to make the system asymptotically stable about the origin($x = 0$), thus let the sliding manifold be defined as $s((x), t) = 0$[45].

$$s(\mathbf{x}, t) = \dot{x} + \lambda x = 0 \tag{7.2}$$

It is important to take note of the properties of the chosen sliding manifold.

- the system dynamics on the sliding manifold are exponentially stable being governed by the 1^{st} order differential equation $\dot{x} = -\lambda x$.
- the system dynamics on the sliding manifold do not depend on any plant

[1] this does not apply for higher-order sliding mode

parameters, it is from this property that sliding mode derives its robust characteristics

- the convergence of the sliding mode regime to the origin can be varied by varying the parameter λ

In designing the control u the equivalent control method[45] shall be used in which the control u is given by:

$$u = u_{eq} + u_{sw} \tag{7.3}$$

Where u_{eq} is the control that would maintain $\dot{s} = 0$ if the uncertainty was absent while u_{sw} is a discontinuous term that ensures that the system reaches the sliding manifold. Designing for u_{eq} we have:

$$\begin{aligned} \dot{s} &= \ddot{x} + \lambda \dot{x} \\ &= g(x)u + \lambda \dot{x} + f_{nom}(x) \quad \text{for } \dot{s} = 0 \text{ we need } u_{eq} \text{ given by} \\ u_{eq} &= -f_{nom}(x) - \frac{\lambda \dot{x}}{g(x)} \end{aligned} \tag{7.4}$$

The next step is to now design for the discontinuous component u_{sw} such that the following inequality is satisfied[45].

$$\frac{1}{2}\frac{d}{dt}s^2 \leq -\eta|s| \tag{7.5}$$

The meaning of this condition can be understood from two points of view.

1. Consider the candidate Lyapunov function $V = \frac{s^2}{2}$ it is evident that (7.5) translates to requiring that the $\dot{s}$ dynamics be stable in the Lyapunov sense

2. the s^2 term can be viewed as a measure of the distance from the sliding surface as such the requirement in (7.5) can be thought of as requiring the distance from the sliding manifold to be always non-increasing

Applying (7.5) gives:

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}s^2 &= s\left[f(x)+g(x)\left(u_{eq}+u_{sw}\right)+\lambda\dot{x}\right] \quad \text{recall } u_{eq} = -f_{nom}(x) - \frac{\lambda\dot{x}}{g(x)} \\ &= s\left[\hat{f}(x)+g(x)u_{sw}\right] \end{aligned} \tag{7.6}$$

Choosing u_{sw} as :

$$u_{sw} = -\frac{K}{g(x)}sgn(s) \tag{7.7}$$

where sgn is the signum function:

$$sgn(s) = \begin{cases} +1 & \text{if } s > 0 \\ -1 & \text{if } s > 0 \end{cases}$$

And the gain K is chosen such that:

$$K = F + \eta, \quad \eta > 0 \tag{7.8}$$

Substituting all this into (7.6) gives:

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}s^2 &= s\left[\hat{f}(x) - Ksgn(s)\right] \\ &= s\hat{f}(x) - K|s| \\ &\leq \eta|s| \end{aligned} \tag{7.9}$$

Thus the control u is given by :

$$u = -f_{nom}(x) - \frac{\lambda\dot{x}}{g(x)} - Ksgn(\dot{x}+\lambda x) \tag{7.10}$$

The typical behavior of a system under sliding mode control can thus be divided into two regimes. If the initial conditions are off the sliding manifold the system enters the first regime which we call the reaching phase. In this regime because of the sliding condition (7.5) the system takes some finite to reach the sliding manifold. On the sliding manifold the system enters the second regime in which the dynamics are governed by sliding manifold equation. Figure 7.1 shows

a simplified phase portrait of a system under sliding mode control.

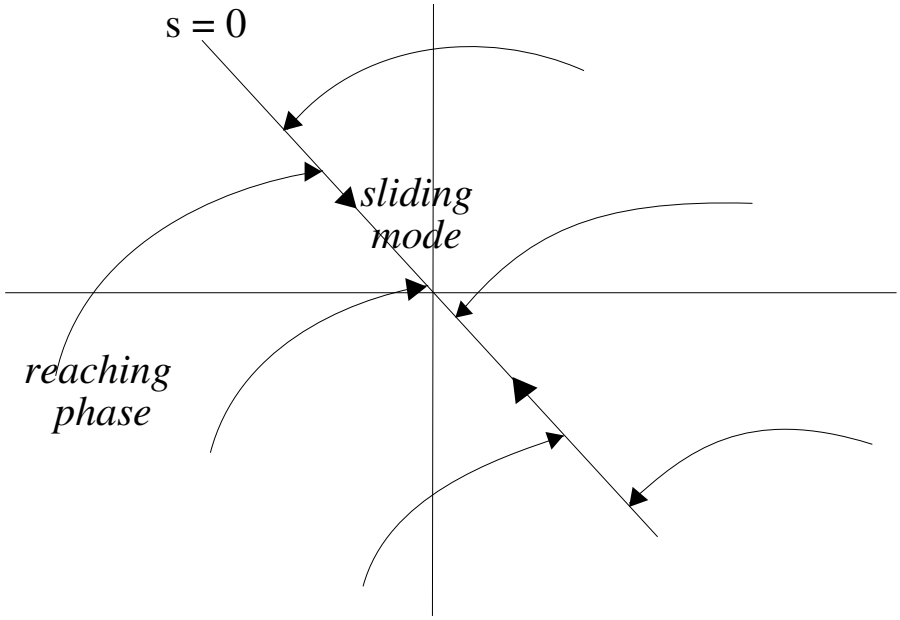


Figure 7.1: Sliding mode controlled system phase portrait[45]

The discontinuous nature of the sliding mode control presents problems since in practice actuators cannot switch instantaneously but have some finite time delay. The effect of this is that there will be high frequency chattering in the control as is depicted in figure 7.2.

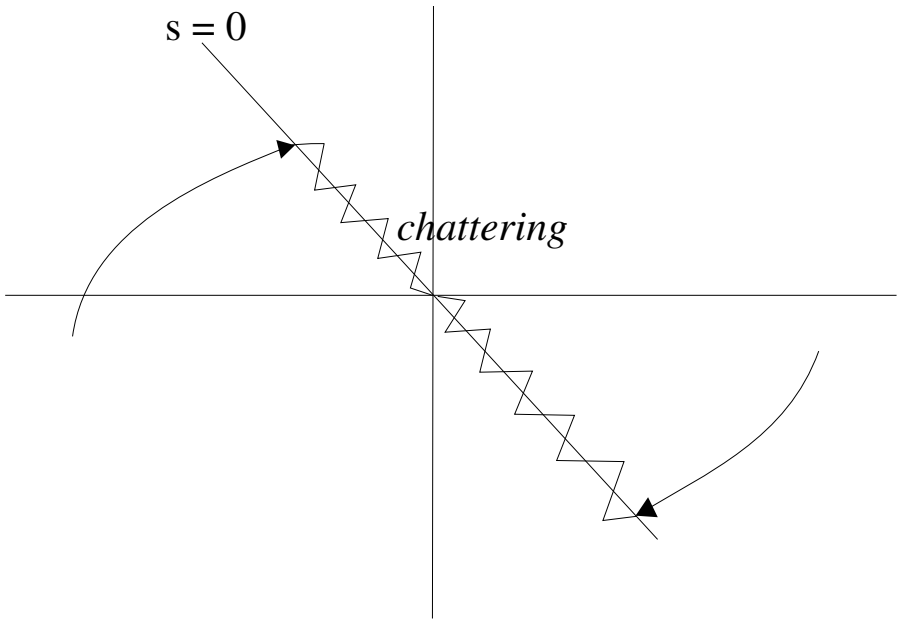


Figure 7.2: Chattering in practical sliding mode control[45]

To try to alleviate the chattering a boundary layer can be defined around the sliding manifold in which the control discontinuity is smoothed out. Effectively this amounts to using some continuous approximation of the signum function such as the inverse tangent function or the hyperbolic tangent function as is shown in figure 7.3. This approach however has the drawback that the slide mode is lost since this approach makes the boundary layer attractive not the sliding manifold. Instead of achieving asymptotic convergence using the boundary layer approach translates to bounded error tracking in which the bounds are proportional to the size of the boundary region[45].

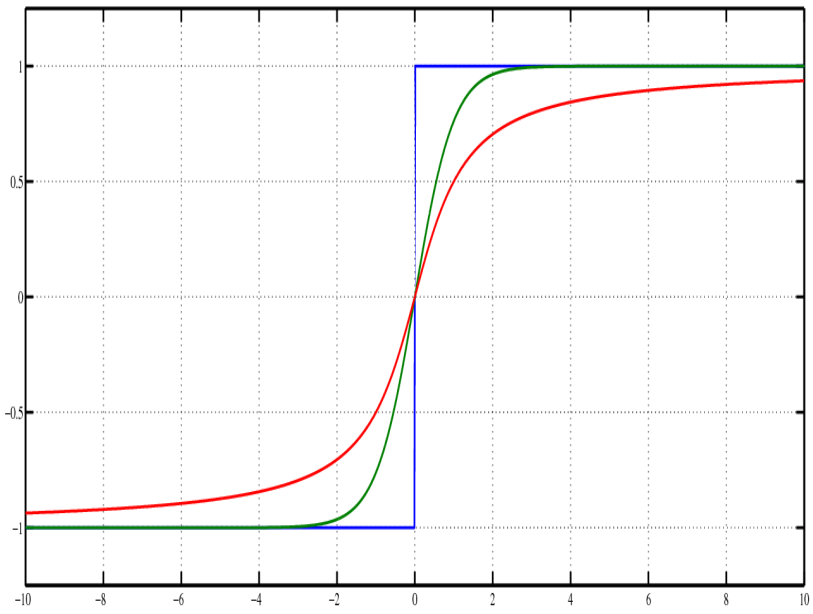


Figure 7.3: Signum function approximation, blue = $sgn(.)$, green = $tanh(.)$, red = $\frac{\pi}{2}arctan(.)$

Example. *Consider the second order system* (7.11) *in which* I *is an unknown parameter and* $\delta(t)$ *is a bounded but unknown function.*

$$\ddot{x} = Isinx + \delta(t) + u \tag{7.11}$$

Also it is known that $I < I_{max}$ *where* $I_{max} > 0$ *and* $|\delta(t)| < \Delta \ \forall t > 0$. *The requirement is to regulate* x *about the origin.*

Comparing the structure of (7.11) with (7.1) it is clear that:

$$f(x) = Isinx + \delta(x), \quad g(x) = 1$$

Assuming that there is no information about the uncertainties except the upper bounds this implies that $f_{nom}(x) = 0$. The control u is chosen as:

$$u = -\lambda\dot{x} - Ksgn\left(\dot{x} + \lambda x\right), \quad \lambda > 0 \tag{7.12}$$

$$K = I_{max} + \Delta + \eta, \quad \eta > 0 \tag{7.13}$$

To illustrate the effect of the continuous approximation of the signum function two different scenarios are simulated in which the signum function is replaced by $tanh(10x)$ and $tanh(500x)$. These two functions are plotted in figure 7.4 for comparison.

The two controls were simulated with the initial conditions $x = 0.1$, $\dot{x} = 0$, from the results depicted in figure 7.5-7.7 it is clear that in as much as a bigger boundary layer removes the chattering however this deteriorates the controller's performance.

7.3 Adaptive Sliding Mode Control

Sliding mode control has been shown to be a powerful robust control method however the requirement of knowledge of the uncertainty upper bounds is difficult to satisfy in real life. As such a lot of effort has been put to research into methods in which this requirement is relaxed, adaptive sliding mode control is one such method which seeks to remove the requirement of *a priori* knowledge of the uncertainty bounds. The general approach of adaptive sliding mode control is to make the gain dynamic such that it is adapted in a way that it can counteract the effect of the uncertainties. To achieve this adaptation one of the methods that has been used is intelligent controllers.

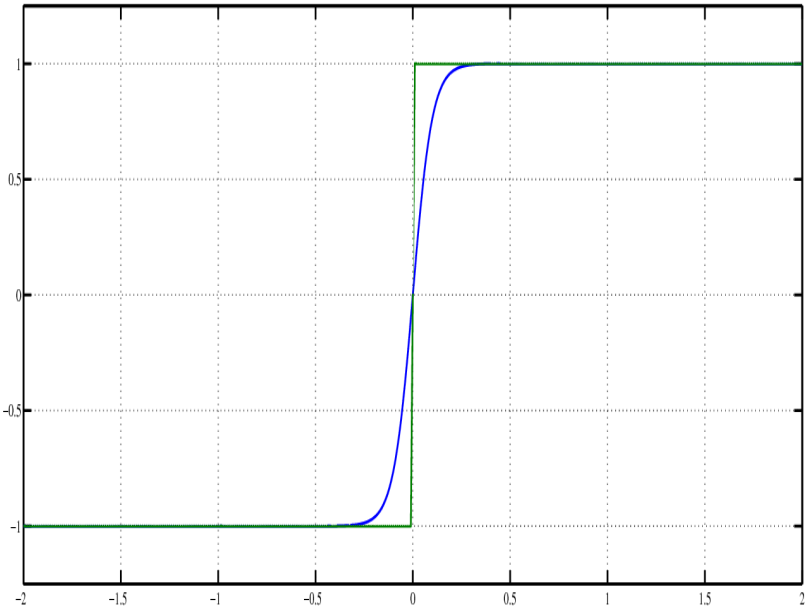


Figure 7.4: Signum function approximation: Green = $tanh(10x)$ and blue = $tanh(500x)$

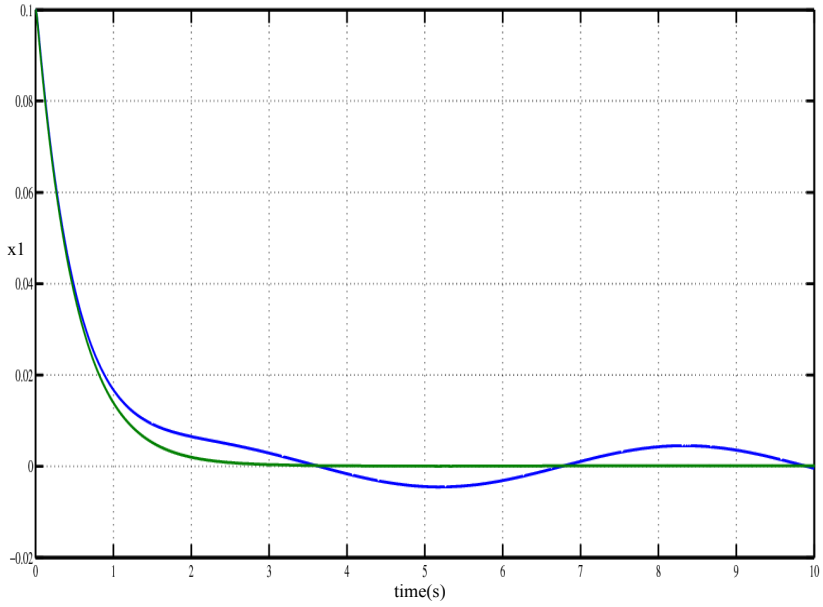


Figure 7.5: x plot: Green = $tanh(10x)$ and blue = $tanh(500x)$

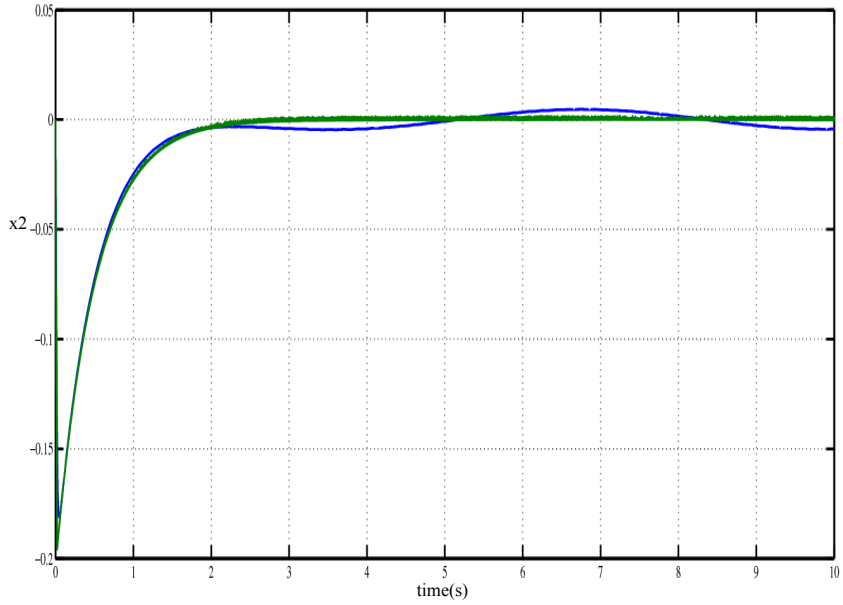


Figure 7.6: $\dot{x}$ plot: Green = $tanh(10x)$ and blue = $tanh(500x)$

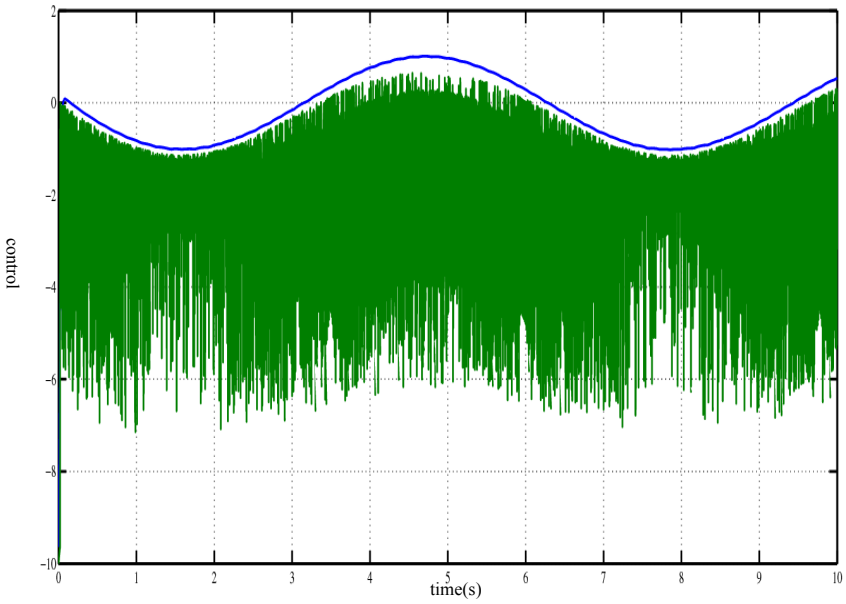


Figure 7.7: u plot: Green = $tanh(10x)$ and blue = $tanh(500x)$

Fuzzy logic based adaptive sliding mode controllers were developed in [47],[79] neural networks were considered in the work of Hu et al[80]. In this chapter the focus is on two adaptive sliding mode control methods developed in [48] and [81].

7.3.1 Method 1

To illustrate the first method consider again the general second order system described by equation (7.1). Also assume that the upper bounds of the uncertainty are unknown. The choice of the sliding manifold is still unchanged and thus also the equivalent control will still be given by (7.4). Since the upper bound of $\hat{f}(x)$ is unknown assume that there exists a positive constant K^* such that:

$$|\hat{f}(x)| < K^*, \quad \forall t > t_0 \tag{7.14}$$

Recall the sliding condition in (7.5), augmenting it with a quadratic term of the estimation error of the constant K^* where the estimation error is given by $\tilde{K} = K^* - \hat{K}$ with $\hat{K}$ being the estimate. The new sliding condition becomes:

$$\frac{1}{2}\frac{d}{dt}\left(s^2 + \frac{\tilde{K}^2}{\gamma_1}\right) < -W(x), \quad \gamma_1 > 0 \text{ and } W(x) \text{is positive definite[81]} \tag{7.15}$$

Evaluating (7.15) along the solutions of (7.1) gives:

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\left(s^2 + \frac{\tilde{K}^2}{\gamma}\right) &= s\dot{s} + \frac{\tilde{K}\dot{\tilde{K}}}{\gamma} \\ &= s\left[f(x) + g(x)u + \lambda\dot{x}\right] + \frac{\tilde{K}\dot{\tilde{K}}}{\gamma}\end{aligned} \tag{7.16}$$

Let $u = u_{eq} + u_{sw}$ with u_{eq} being the same as in (7.4) and $u_{sw} = -\frac{\hat{K}sgn(s)}{g(x)}$. Substituting u into (7.16):

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\left(s^2 + \frac{\tilde{K}^2}{\gamma}\right) &= s\left[f_{nom}(x) + \hat{f}(x) + g(x)\left(u_{eq} + u_{sw}\right) + \lambda\dot{x}\right] + \frac{\tilde{K}\dot{\tilde{K}}}{\gamma} \\ &= s\left[\hat{f}(x) - \hat{K}sgn(s)\right] + \frac{\tilde{K}\dot{\tilde{K}}}{\gamma} \\ &= s\left[\hat{f}(x) - K^*sgn(s)\right] + \tilde{K}|s| + \frac{\tilde{K}\dot{\tilde{K}}}{\gamma} \qquad (7.17)\end{aligned}$$

Choosing the update law for $\hat{K}$ as:

$$\dot{\hat{K}} = \gamma|s| \qquad (7.18)$$

Finally gives :

$$\frac{1}{2}\frac{d}{dt}\left(s^2 + \frac{\tilde{K}^2}{\gamma}\right) \leq -\eta|s| \qquad (7.19)$$

Application of La Salle's theorem to (7.15) shows that $x(t)$ will converge to the largest invariant set $M = \{x|W(x) = 0\}$. In the case of (7.19) $W(x) = \eta|s|$ and thus it is guaranteed that $x(t)$ will converge to the sliding manifold $s(x,t) = 0$. A closer look however at the adaptation law (7.18) shows that this adaptation law tends to over-estimate the sliding mode gain K. Consider the case where the system' s initial conditions are off the sliding manifold, from (7.18) $\hat{K}$ will continue to increase even though the system trajectory might be getting closer to the sliding manifold. Even when $s(x,t) = 0$ the sliding gain estimate does not go to zero which is an over-estimate of the effect of the uncertainties. Another problem that comes with the developed update law is that if the boundary layer method is used the sliding mode is never reached (i.e $s(x,t) \neq 0$) meaning that the estimate $\hat{K}$ will always be increasing.

7.3.2 Method 2

The disadvantages of the adaptive law (7.18) were also highlighted in the work of Plestan[81] in which they proposed an improved adaptive law which is presented here. Consider the parameter update law:

$$\dot{\hat{K}} = \begin{cases} \gamma|s(x,t)|sign(|s(x,t)| - \epsilon) & \text{if } \hat{K} > \mu \\ \mu & \text{if } \hat{K} \leq \mu \end{cases} \tag{7.20}$$

with $\hat{K}(0) > 0$, $\gamma > 0$, $\epsilon > 0$ and $\mu > 0$. Note that the parameter μ represents the lower limit of the estimate $\hat{K}$ thus $\hat{K} > \mu \ \forall \ t$, also by making μ positive this guarantees that the estimate $\hat{K}$ is always positive. The parameter ϵ defines a boundary region $B(x)$ about the sliding manifold $B(x) = \{x|s(x,t) - \epsilon < 0\}$. Outside the boundary region $B(x)$ with the additional condition that $\hat{K} > \mu$, the update law (7.20) is similar to that in equation (7.18). Inside the boundary region $B(x)$ however the update law is such that the estimate is decreasing, this avoids the gain over-estimation that is characteristic of the first update law. Again it can be seen that the parameter μ guards against the estimate decreasing to become negative which would lead to instability.

Example. *Consider the system given by equation* (7.21).

$$\ddot{x} = Isinx + \delta(t) + u \tag{7.21}$$

where I is an unknown constant parameter and $\delta(t)$ is an unknown bounded function. The requirement is to regulate x about the origin.

Using the two adaptive sliding mode control methods that have been devel-

oped gives the control and update laws:

$$s(x,t) = \dot{x} + \lambda x \tag{7.22a}$$

$$u = -\lambda\dot{x} - \hat{K}sgn(s(x,t)) \tag{7.22b}$$

$$\dot{\hat{K}} = \gamma|s(x,t)| \tag{7.22c}$$

$$\dot{\hat{K}} = \begin{cases} \gamma|s(x,t)|sgn\left(|s(x,t)| - \epsilon\right) & \text{if } \hat{K} > \mu \\ \mu & \text{if } \hat{K} \leq \mu \end{cases} \tag{7.22d}$$

The following values for controller and update law parameters are chosen: The controller with the two update laws was simulated in MATLAB for the case

$$\begin{array}{cc} \lambda = 2 & \epsilon = 0.1 \\ \mu = 0.01 & \gamma = 5 \end{array}$$

were the disturbance $\delta(t)$ is a sinusoid of unit amplitude and initial conditions = $(x = 1, \dot{x} = 0)$. The results as shown in figure 7.8-7.11 show that both controllers have nearly the same regulation performance however the adaptive scheme of the second method utilises lower gains than the first method as was expected.

7.4 Sliding Backstepping Control

Two techniques for integrating sliding mode control with backstepping control are developed in this section. The first method relies on a Lyapunov based approach while the second method hinges on the appropriate choice of a sliding manifold. It was stated earlier in the previous chapter that the backstepping technique strictly speaking is not a control synthesis technique but rather a way of constructing a Lyapunov function. From this viewpoint the backstepping method can be used to construct the Lyapunov function and other methods such as sliding mode control can then be used to make the derivative of the Lyapunov function nonpositive even if uncertainties are present.

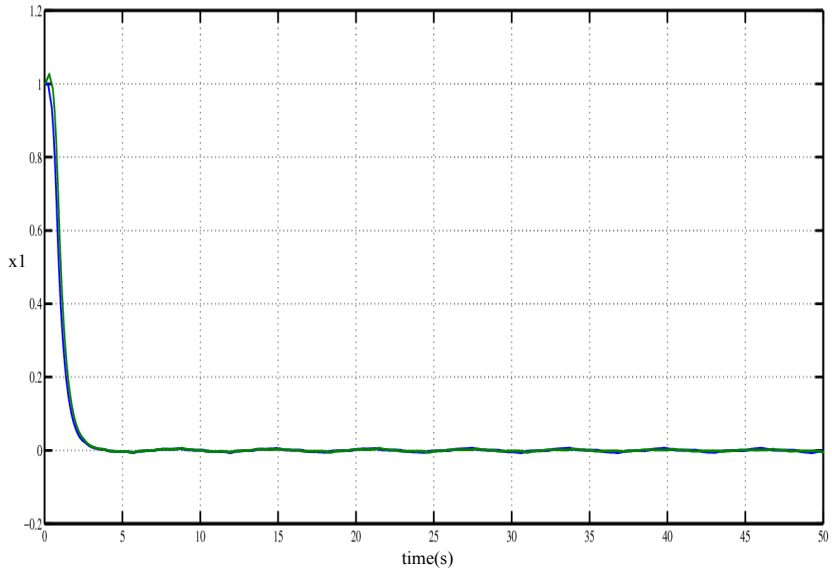


Figure 7.8: x plot: Blue = method 1 adaptation, green = method 2 adaptation

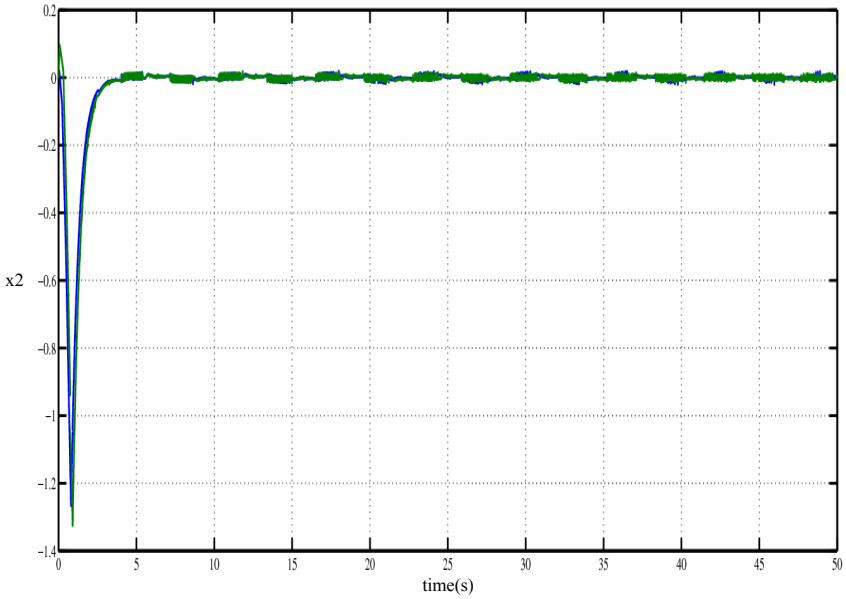


Figure 7.9: $\dot{x}$ plot: Blue = method 1 adaptation, green = method 2 adaptation

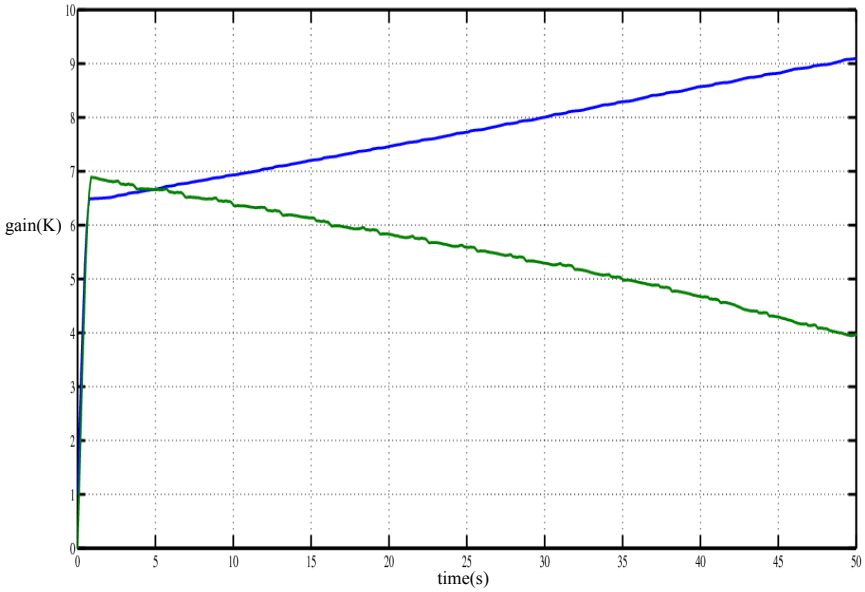


Figure 7.10: $\hat{K}$ plot: Blue = method 1 adaptation, green = method 2 adaptation

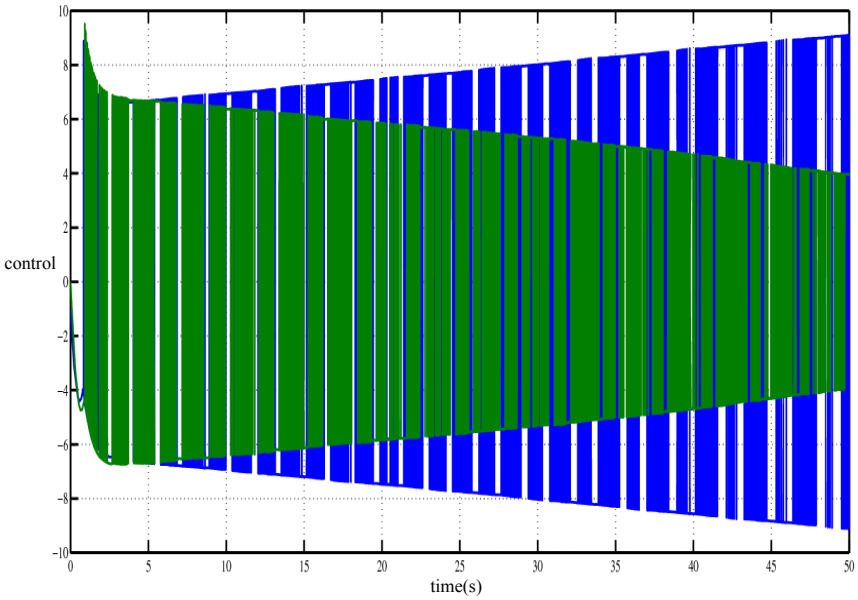


Figure 7.11: u plot: Blue = method 1 adaptation, green = method 2 adaptation

To illustrate this approach consider the second order system described by (7.23).

$$\dot{x}_1 = f_1(x_1) + x_2 \tag{7.23a}$$
$$\dot{x}_2 = f_2(x_1, x_2) + u \tag{7.23b}$$

Let $f_2(x_1, x_2)$ be an unknown but bounded function with upper bound F such that (7.23) has a matched uncertainty. Applying the backstepping technique with the following choice of Lyapunov function and stabilising function:

$$V_1(x_1) = \frac{x_1^2}{2}, \quad \alpha_1(x_1) = -f_1(x_1) - \lambda_1 x_1, \quad \lambda_1 > 0$$

Defining the error variable $z = x_2 - \alpha_1(x_1)$ the system dynamics can be transformed to:

$$\dot{x}_1 = -\lambda_1 x_1 + z_1 \tag{7.24a}$$
$$\dot{z}_1 = f_2(x_1, x_2) - (z_1 - \lambda_1 x_1) \frac{\partial \alpha_1}{\partial x_1}(x_1) + u \tag{7.24b}$$

According to the Backstepping lemma a control Lyapunov function exists for this system and is given by:

$$V_2(x_1, z) = \frac{x_1^2}{2} + \frac{z^2}{2} \tag{7.25}$$

The control u can be designed such that the derivative of (7.25) nonpositive.

$$\begin{aligned} \dot{V}_2 &= x\dot{x} + z\dot{z} \\ &= x_1 \left[-\lambda_1 x_1 + z_1\right] + z \left[f_2(x_1, x_2) - (z - \lambda_1 x_1) \frac{\partial \alpha}{\partial x_1}(x_1) + u \right] \end{aligned} \tag{7.26}$$

Using the equivalent control approach let $u = u_{eq} + u_{sw}$.

$$\dot{V}_2 = -\lambda_1 x_1^2 + z_1 \left[x_1 + f_2(x_1, x_2) - (z_1 - \lambda_1 x_1) \frac{\partial \alpha_1}{\partial x_1}(x) + u_{eq} + u_{sw} \right] \tag{7.27}$$

If u_{sw} and u_{eq} are chosen as:

$$u_{eq} = -x_1 + (z_1 - \lambda_1 x_1) \frac{\partial \alpha_1}{\partial x_1}(x_1) - \lambda_2 z,\ \lambda_2 > 0 \tag{7.28}$$

$$u_{sw} = -Ksgn(z),\ K = F + \eta,\ \eta > 0 \tag{7.29}$$

Substituting in (7.27) gives:

$$\begin{aligned} \dot{V}_2 &= -\lambda_1 x_1^2 + z\left[f_2(x_1, x_2) - Ksgn(z)\right] \\ &\leq -\lambda_1 x_1^2 - \eta|z| \end{aligned} \tag{7.30}$$

From (7.30) it can be seen then that the requirement of regulation of x_1 about the origin is achieved by the sliding backstepping control given by (7.31).

$$u = -x_1 + (z_1 - \lambda_1 x_1) \frac{\partial \alpha_1}{\partial x_1}(x_1) - \lambda_2 z - Ksgn(z) \tag{7.31}$$

An alternative approach of formulating the sliding backstepping involves defining $z = 0$ as our sliding surface. This choice of the sliding manifold is reasonable since $z = 0$ implies that $x_2 = \alpha_1(x_1)$ which stabilizes the x_1 subsystem described by (7.23a). Applying the sliding mode technique, the equivalent control u_{eq} is designed by considering the nominal dynamics of $\dot{z}$.

$$\begin{aligned} \dot{z} &= -\left(z - \lambda_1 x_1\right) \frac{\partial \alpha_1}{\partial x_1}(x_1) + u_{eq} \\ u_{eq} &= \left(z - \lambda_1 x_1\right) \frac{\partial \alpha_1}{\partial x_1}(x_1),\ \text{to ensure that } \dot{z} = 0 \end{aligned} \tag{7.32}$$

For designing the discontinuous control component u_{sw} the following sliding condition should be met:

$$\frac{1}{2}\frac{d}{dt}z^2 \leq \eta|z|,\ \eta > 0 \tag{7.33}$$

Evaluating the left hand of the sliding condition:

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}z^2 &= s\left[f_2(x_1,x_2) - (z-\lambda_1 x_1)\frac{\partial \alpha_1}{\partial x_1}(x_1) + u\right] \\ \text{let } u &= u_{eq} + u_{sw},\ u_{sw} = -Ksgn(z),\ K = F + \eta \\ &= z\left[f_2(x_1,x_2) - Ksgn(z)\right] \\ &\leq -\eta|z| \end{aligned} \tag{7.34}$$

The second sliding backstepping controller formulated using the alternative method is therefore given by :

$$u = (z-\lambda_1 x_1)\frac{\partial \alpha_1}{\partial x_1}(x_1) - Ksgn(z) \tag{7.35}$$

Comparing the two controllers (7.31) and (7.35), the second controller is seen to (7.35) have a simpler structure only requiring one gain parameter. Although simplicity is desirable it might also be viewed as a handicap in this case as there being only one parameter to vary reduces the designer's freedom. To compare the performance of these two controllers consider the performance of the controllers when controlling the system given in the previous example. In this example it is assumed that the upper bound for I and $\delta(t)$ is known. Let $x_1 = x$ and $x_2 = \dot{x}$ the system will be described by the equations:

$$\dot{x}_1 = x_2 \tag{7.36a}$$

$$\dot{x}_2 = Isinx_1 + \delta(t) + u \tag{7.36b}$$

Comparing this to the form of the general system (7.23) gives the following correspondences.

$$\begin{aligned} f_1(x_1) = 0, \quad f_2(x_1,x_2) &= Isinx_1 + \delta(t) \\ F &= I_{max} + \Delta \end{aligned}$$

where I_{max} and Δ are the upper bounds of I and $\delta(t)$ respectively. Thus applying

the two sliding backstepping techniques gives the two controls:

$$u_1 = -\left(1 + \lambda_1\lambda_2\right) x_1 - \left(\lambda_1 + \lambda_2\right) x_2 - Ksgn(x_2 + \lambda_1 x_1) \tag{7.37a}$$

$$u_2 = -\lambda_1^2 x_1 - \lambda_1 x_2 - Ksgn(x_2 + \lambda_1 x - 1) \tag{7.37b}$$

The two controls were simulated in MATLAB for the case where $\delta(t)$ is a sinusoid of unit amplitude and the true value of I is 2 . For the controller gains the following values were used.

$$\lambda_1 = 2, \lambda_2 = 0.5, K = 6.5$$

From the plots of the simulation results shown in figures 7.12-7.14 one can see that there is little difference in the controls both have nearly the same responses for $x_1(t)$ and $x_2(t)$.

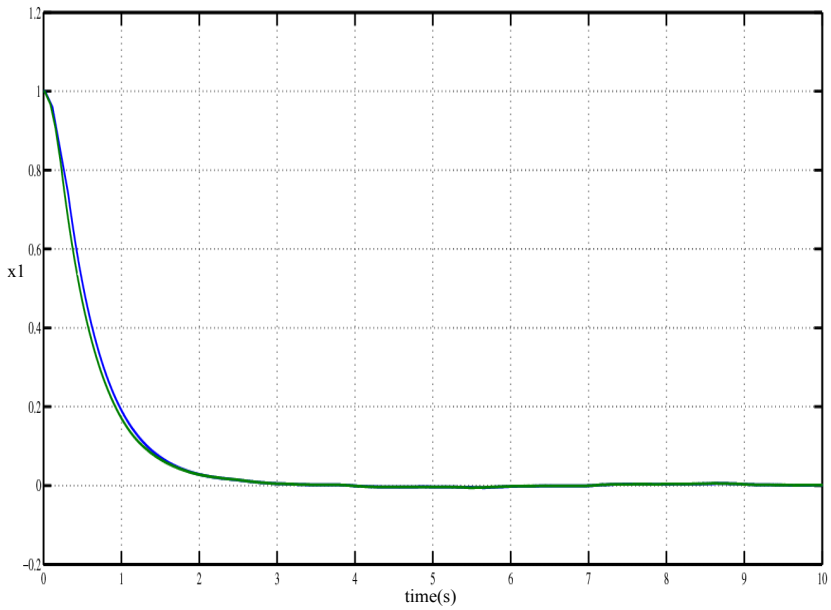


Figure 7.12: $x_1(t)$ plot: Blue = u_1, green = u_2

It should be easy to see that the sliding backstepping control that has been

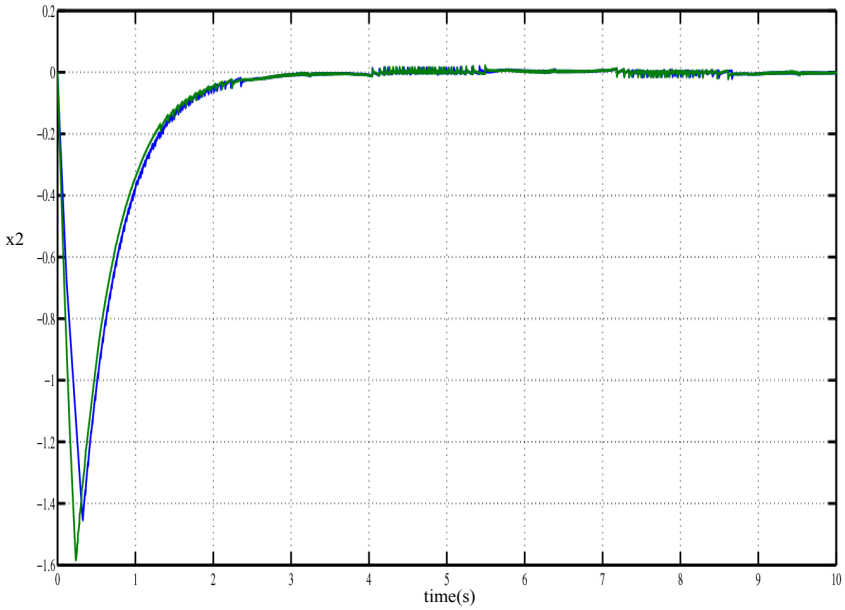


Figure 7.13: $x_2(t)$ plot: Blue = u_1, green = u_2

developed here can be modified by combining it with the adaptive techniques of the previous section to come up with an adaptive sliding backstepping scheme in which *a priori* knowledge of the uncertainty bounds is not required. Such a controller is what is used in the next section for the attitude controller.

7.5 Attitude Control

In this section all the techniques that have been developed through out the chapter are combined providing the main result of this chapter, an adaptive sliding backstepping attitude controller. For combining sliding mode control and backstepping control the second method presented in the previous section is used, the adaptation rule will be developed using the method in section 7.3.2. The general approach however follows that developed in the previous chapter and thus some non-essential steps will not be explicitly stated.

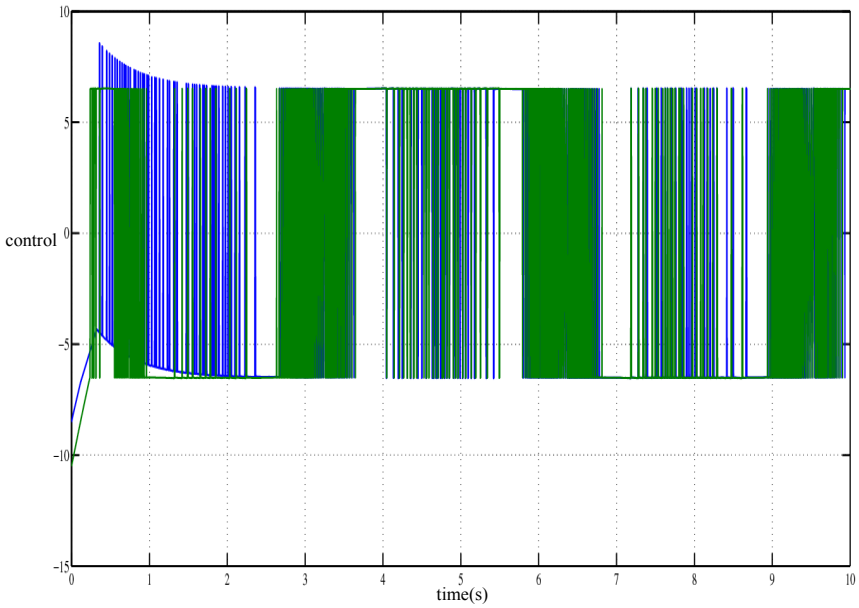


Figure 7.14: Control input plot: Blue = u_1, green = u_2

7.5.1 Orientation Controller

From section 6.5.1 the error dynamics of ρ_{13} are:

$$\dot{\chi} = e_{\rho_{13}} \tag{7.38a}$$

$$\dot{e}_{\rho_{13}} = -\dot{\rho}_{13d} + q\rho_{11} - \rho_{21}p \tag{7.38b}$$

$$\dot{p} = I_\phi qr + \delta_\phi(t) + \tau_\phi \tag{7.38c}$$

Applying the backstepping technique and choosing the following stabilising functions:

$$\alpha_1(\chi) = -\lambda_1\chi, \quad z_1 = e_{\rho_{13d}} - \alpha_1(\chi)$$

$$\alpha_2(\chi, z_1) = \frac{1}{\rho_{12}}\left[-\dot{\rho_{13d}} + \chi + q\rho_{11} + \lambda_1 z_1 - \lambda_1^2\chi + \lambda_2 z_1\right], \quad z_2 = p - \alpha(\chi, z_1)$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$

Thus the plant dynamics can be expressed in terms of χ, z_1 and z_2:

$$\dot{\chi} = -\lambda_1 \chi + z_1 \tag{7.39a}$$

$$\dot{z}_1 = -\chi - \lambda_2 z_1 - \rho_{21} z_2 \tag{7.39b}$$

$$\dot{z}_2 = I_\phi qr - (z_1 - \lambda_1 \chi) \frac{\partial \alpha_2}{\partial \chi} + (\chi + \lambda_2 z_1 + \rho_{21} z_2) \frac{\partial \alpha_2}{\partial z_1} + \delta_\phi(t) + \tau_\phi \tag{7.39c}$$

Assume that there exists a positive constant K^* such that:

$$K^* > I_\phi qr + \delta_\phi(t) \tag{7.40}$$

Let the sliding manifold be defined by $z_2 = 0$. The sliding manifold is made attractive if the following sliding condition is satisfied:

$$\frac{1}{2} \frac{d}{dt} z_2^2 \leq -\eta |z_2| \tag{7.41}$$

Applying the sliding and adaptive techniques developed in the previous section gives the following control and adaptation law:

$$\tau_\phi = (z_1 - \lambda_1 \chi) \frac{\partial \alpha_2}{\partial \chi} - (\chi + \lambda_2 z - 1 + \rho_{21} z_2) \frac{\partial \alpha_2}{\partial z_1} - \hat{K} sgn(z_2) \tag{7.42}$$

$$\dot{\hat{K}} = \gamma |z|_2 \tag{7.43}$$

However as stated in section 7.3 this kind of update law tends to over estimate gains the alternative update law that was presented in section 7.3.2 will be used. Thus the complete control and update laws for the ρ_{13} dynamics are:

$$\tau_\phi = (z_1 - \lambda_1 \chi) \frac{\partial \alpha_2}{\partial \chi} - (\chi + \lambda_2 z - 1 + \rho_{21} z_2) \frac{\partial \alpha_2}{\partial z_1} - \hat{K} sgn(z_2) \tag{7.44}$$

$$\dot{\hat{K}} = \begin{cases} \gamma |z_2| sgn(|z_2| - \epsilon) & \text{if } \hat{K} > \mu \\ \mu & \text{if } \hat{K} \leq \mu \end{cases} \tag{7.45}$$

7.5.2 Yaw Controller

The procedure for designing the yaw controller is very similar to the one described above for the orientation controller thus for brevity's sake the yaw controller and

update law is only stated.

$$\tau_\psi = (z_1 - \lambda_1 \chi_\psi) \frac{\partial \alpha_2}{\partial \chi_\psi} - \left(z_2 \frac{cos\phi}{cos\theta} - \chi_\psi - \lambda_2 z_1 \right) \frac{\partial \alpha_2}{\partial z_1} \tag{7.46a}$$

$$\dot{\hat{K}} = \begin{cases} \gamma |z_2| sgn(|z_2| - \epsilon) & \text{if } \hat{K} > \mu \\ \mu & \text{if } \hat{K} \leq \mu \end{cases} \tag{7.46b}$$

where z_1,z_2 and the stabilising functions are given by:

$$\begin{aligned} \alpha_1(\chi) &= -\lambda_1 \chi_\psi, \quad z_1 = e_\psi - \alpha_1(\chi) \\ \alpha_2(\chi, z_1) &= \frac{cos\theta}{cos\phi} \left[-\chi_\phi + \dot{\psi}_d + \frac{sin\phi}{cos\theta} q - \lambda_1 z_1 + \lambda_1^2 \chi_\psi - \lambda_2 z_1 \right] \\ z_2 &= r - \alpha_(\chi, z_1) \end{aligned}$$

7.5.3 Simulation Results

The attitude controller developed in this chapter was simulated in MATLAB/SIMULINK, the task was to track the time varying reference trajectories for the attitude angles given by:

$$\begin{aligned} \phi_d(t) &= sin(\pi t) \\ \theta_d(t) &= sin(0.5\pi t) \\ \psi_d(t) &= -0.75 sin(\pi t) \end{aligned}$$

The system was perturbed by a sinusoidal input with unit amplitude and the hyperbolic tangent function was used as the continuous approximation of the signum function . The following values were chosen by trial and error for the controller parameters.

Parameter	ϕ controller	θ controller	ψ controller
λ_1	6	6	6
λ_2	4	4	4
λ_3	0	0	0
γ	5	5	5
ϵ	1	1	1
μ	0.01	0.01	0.01

Looking at the plots of the angles in figures 7.15-7.17 it can be seen that there is almost no difference between the system's response when disturbances are present and when they are absent thus showing the strong disturbance rejection characteristics of the controller. Also from the plots of the gains in figures 7.18-7.20 the adaptation law is seen to ensure that the gains are not just increasing but rather gives the smallest estimate which will counteract the effect of the uncertainties.

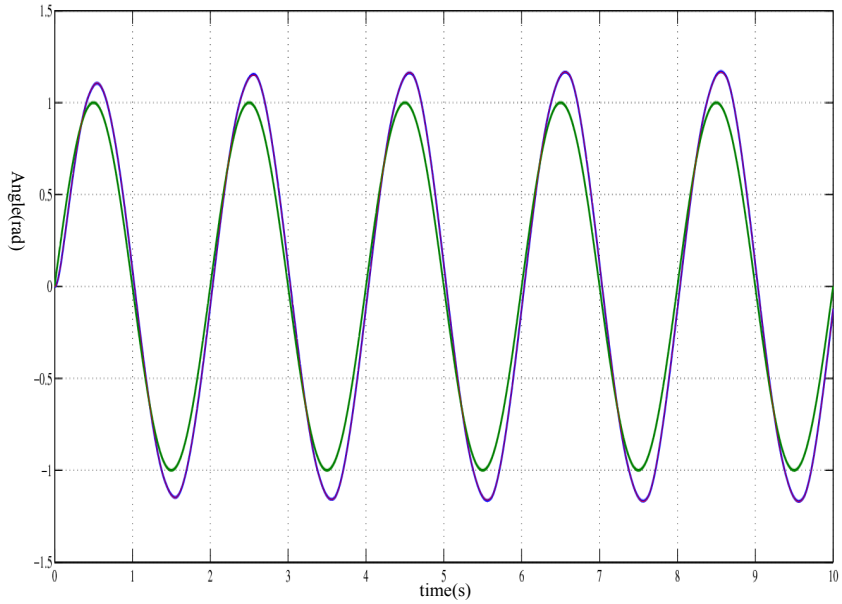


Figure 7.15: ϕ plot: Green = reference signal, blue = without disturbance, red = with disturbance

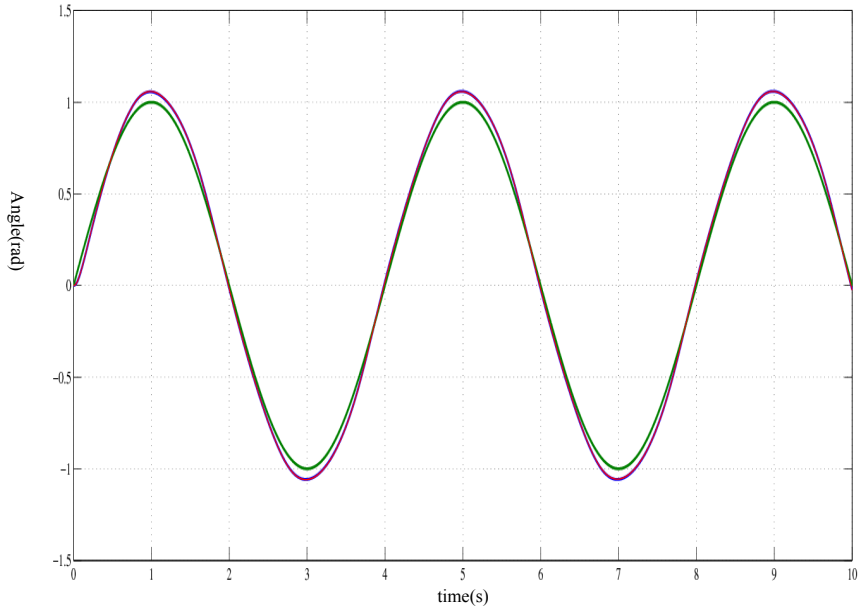


Figure 7.16: θ plot: Green = reference signal, blue = without disturbance, red = with disturbance

7.6 Conclusion

The attitude controller that was developed in chapter 6 only guarantees bounded error tracking in the presence of disturbances, in this chapter the task was to improve on the results of chapter 6 and develop a controller that can still achieve asymptotic tracking in the presence of uncertainties. To that end two approaches of incorporating sliding mode control into the backstepping scheme are presented. Sliding mode control has the advantage of being able to guarantee asymptotic convergence of tracking errors even in the presence of uncertainty. However sliding mode control has two major drawbacks, firstly one needs to have *a priori* knowledge of the bounds of the uncertainty to design the controller. Secondly sliding mode controllers exhibit high frequency chattering which is not desirable. To counteract these drawbacks an adaptive sliding backstepping controller is developed in which the adaptive component is used to do away with the requirement of knowing the uncertainty upper bounds. An update law is developed which is designed to ensure that the minimum possible gains are used thus avoiding

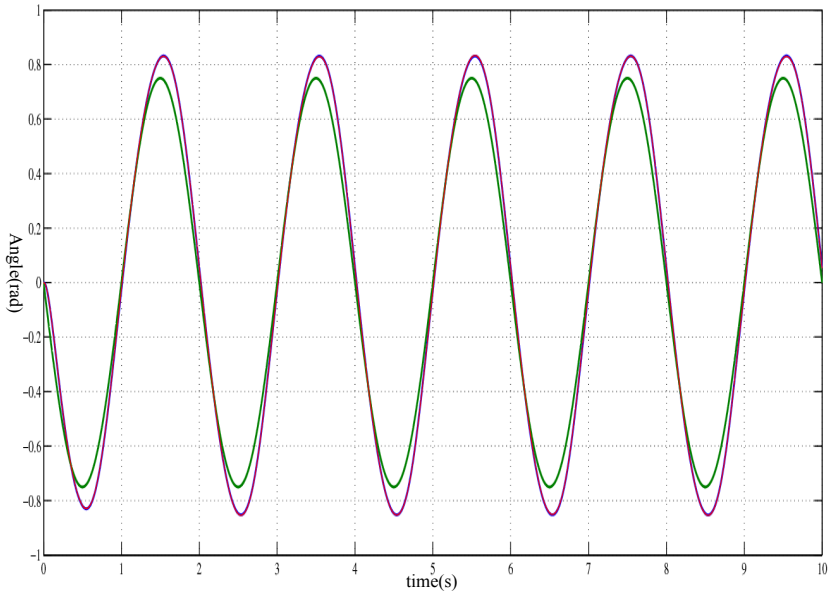


Figure 7.17: ψ plot: Green = reference signal, blue = without disturbance, red = with disturbance

any over-estimation of gain. For alleviating the chattering phenomenon in the developed controller continuous approximations of the signum function are used. Simulations of the developed adaptive sliding backstepping attitude controller show that asymptotic tracking is indeed achieved even in the presence of disturbances.

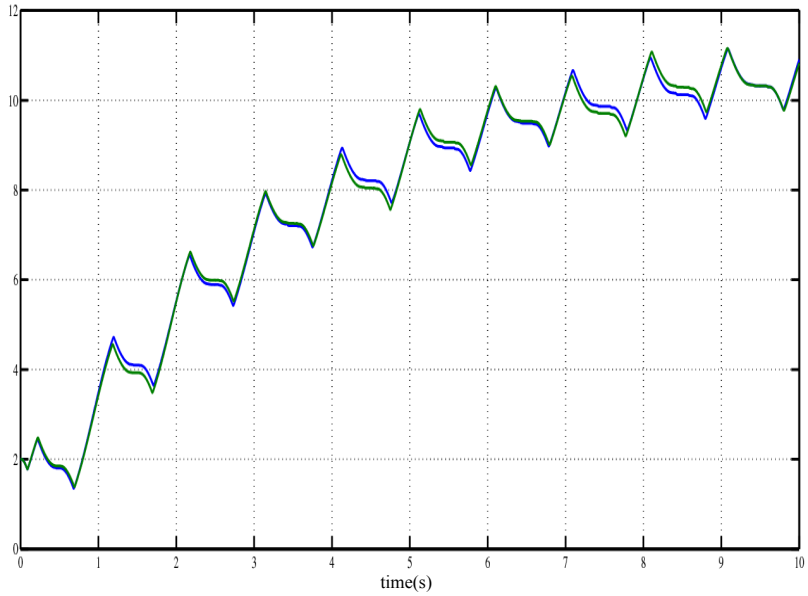


Figure 7.18: $\hat{K}_\phi$ plot: Blue = without disturbance, green = with disturbance

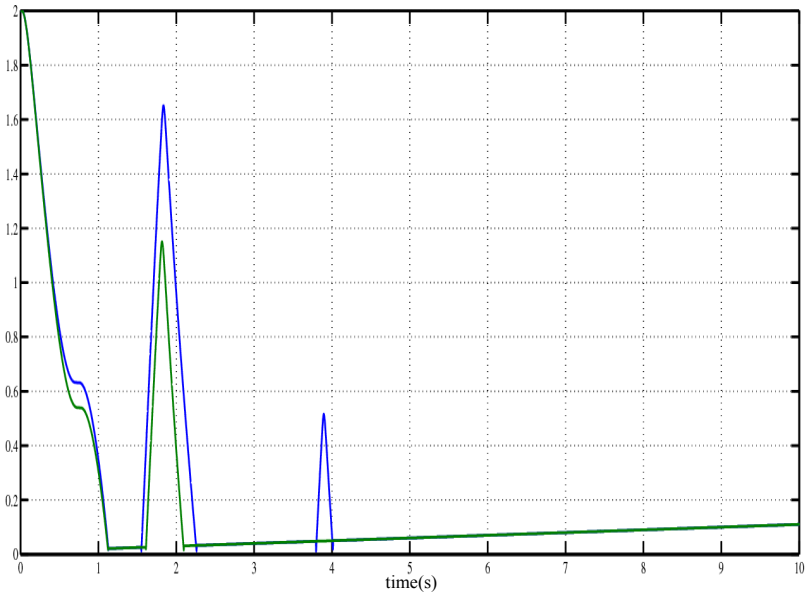


Figure 7.19: $\hat{K}_\theta$ plot: Blue = without disturbance, green = with disturbance

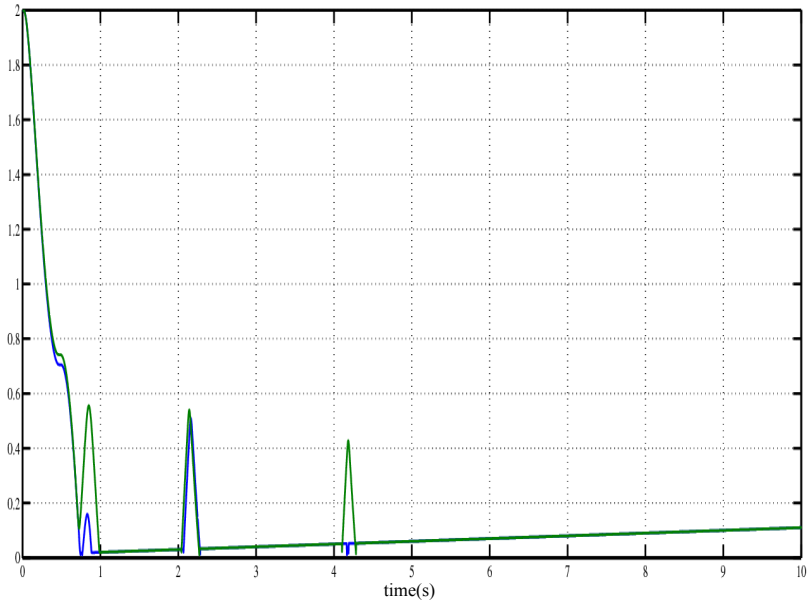


Figure 7.20: $\hat{K}_\psi$ plot: Blue = without disturbance, green = with disturbance

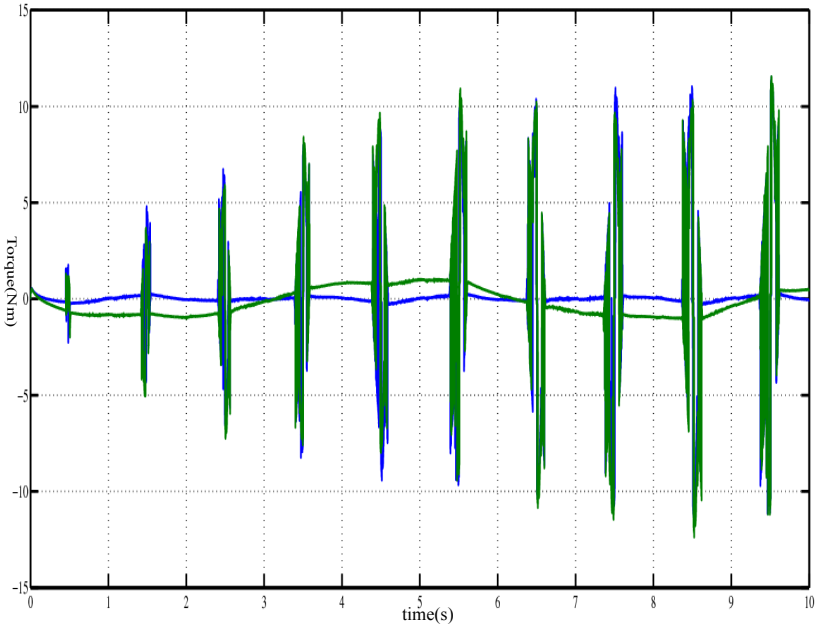


Figure 7.21: τ_ϕ plot: Blue = without disturbance, green = with disturbance

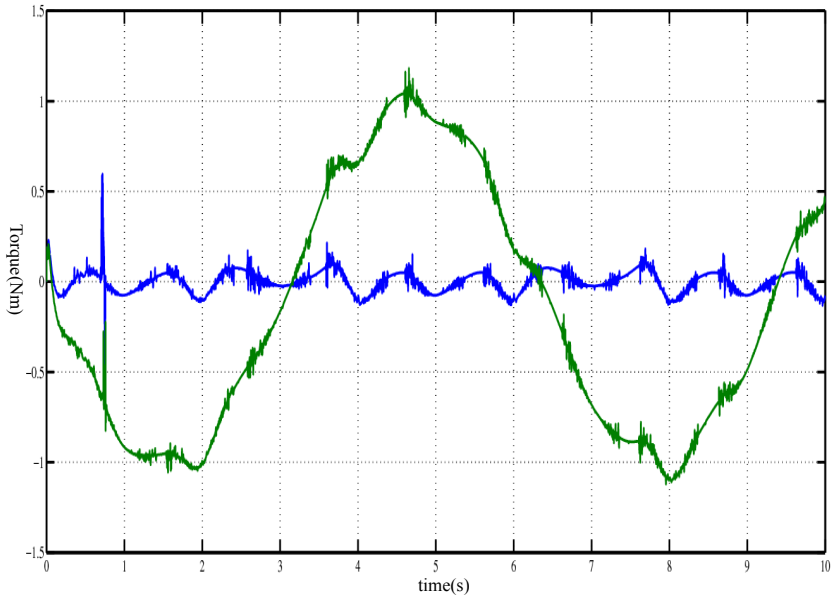


Figure 7.22: τ_θ plot: Blue = without disturbance, green = with disturbance

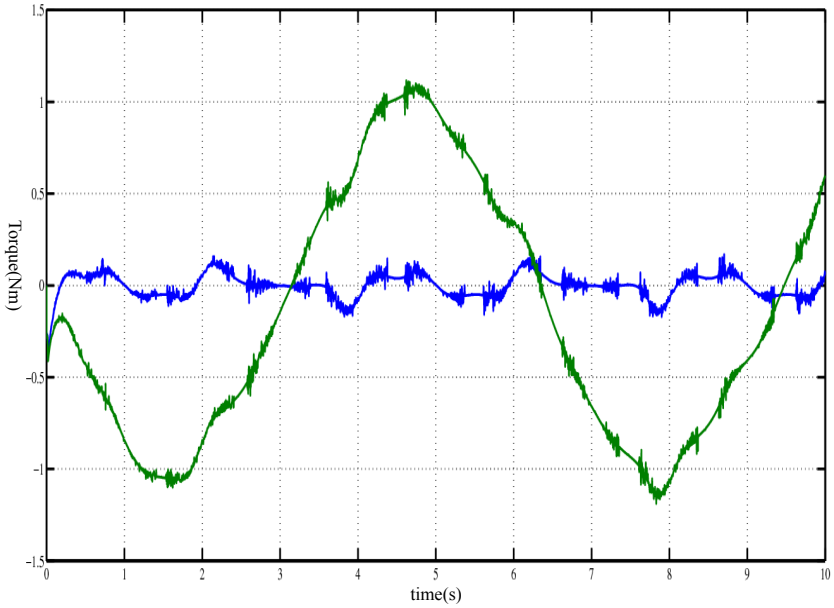


Figure 7.23: τ_ψ plot: Blue = without disturbance, green = with disturbance

Chapter 8

Quadrotor UAV Trajectory Control

8.1 Chapter Overview

In the previous three chapters controllers have been developed for the translational and orientation subsystems with simulations being done for each subsystem independently. In this chapter the subsystems are combined and simulations of the complete quadrotor system are presented. For the orientation subsystem the control algorithm developed in chapter 7 is used as this is to guarantee asymptotic trajectory tracking as opposed to bounded error tracking of the controller developed in chapter 6. The simulations whose results are presented in this chapter are for the quadrotor executing two maneuvers. The first maneuver is a simple hover in which the quadcopter is required to attain a given height and hold that height. The second maneuver consists of the quadcopter rising to a given height and then performing a figure "8". For both these maneuvers the simulations were performed with and without disturbances in the system.

8.2 Simulation Results

For the translational controller the following gain parameters are chosen.

$$M_1 = 3 \quad L_1 = 1.2 \quad K_1 = 0.5$$
$$M_2 = 3 \quad L_2 = 1.2 \quad K_2 = 0.3$$
$$M_3 = 3 \quad L_3 = 1.2 \quad K_3 = 1.5$$

As stated in the preceding section for the orientation control the algorithm developed in chapter 6 s used. The gains of which are chosen as:

Parameter	ϕ gains	θ gains	ψ gains
λ_1	3	12	3
λ_2	2	6	2
λ_3	1	6	1
K	5	5	5
μ	0.01	0.01	0.01
ϵ	0.1	0.1	0.1

To simulate the effect of disturbances a disturbance torque vector and a disturbance force vector are included. For the simulations presented here the elements of both the torque and force disturbance vectors are sinusoids of unit amplitude.

8.2.1 Hovering Maneuver

For the hovering maneuver the reference trajectory is given by:

$$x_d^I = 0$$
$$y_d^I = 0$$
$$z_d^I = -(1 - \exp(\frac{t}{10}))$$

From the results shown in figures 8.1-8.11 it can be seen that in the ideal case where disturbances are absent the controller achieves perfect tracking of the reference even though the inertia is unknown. In the presence of disturbances the controller manages to track the reference with a maximum deviation of only 0.1m.

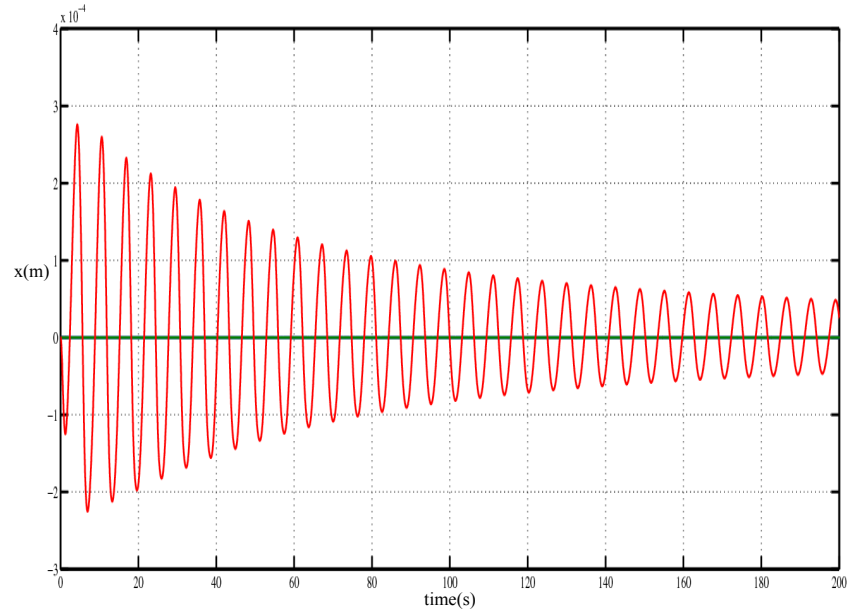


Figure 8.1: Hover maneuver x plot: Green = without disturbance, red = with disturbance

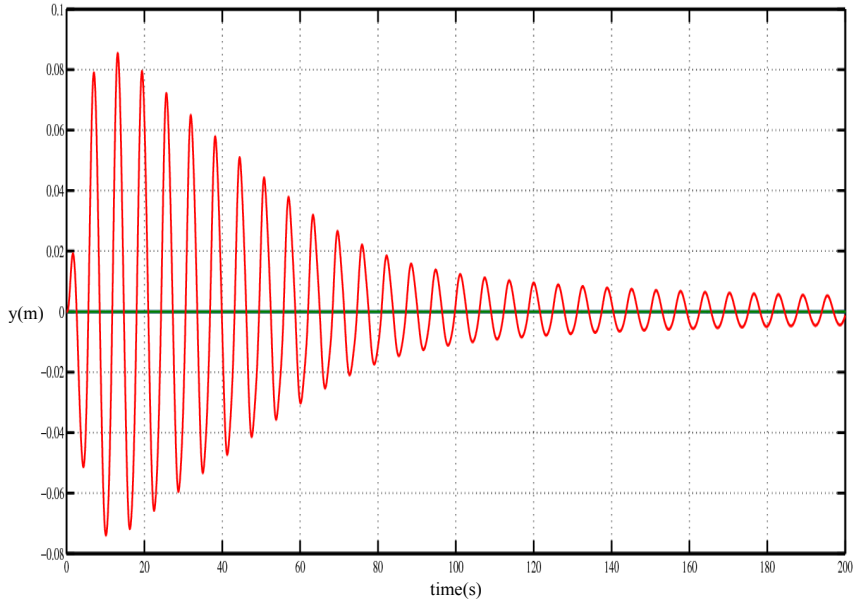


Figure 8.2: Hover maneuver y plot: Green = without disturbance, red = with disturbance

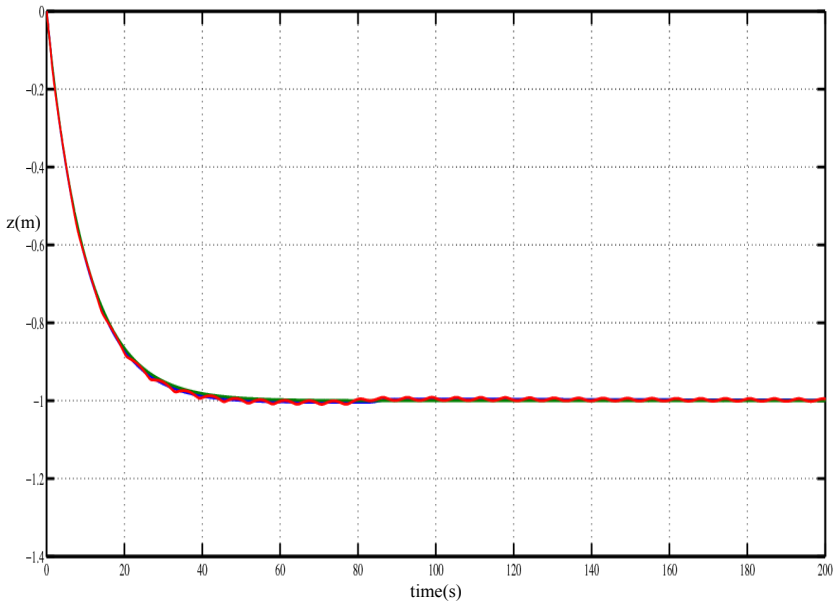


Figure 8.3: Hover maneuver z plot: Blue = reference trajectory, green = without disturbance, red = with disturbance

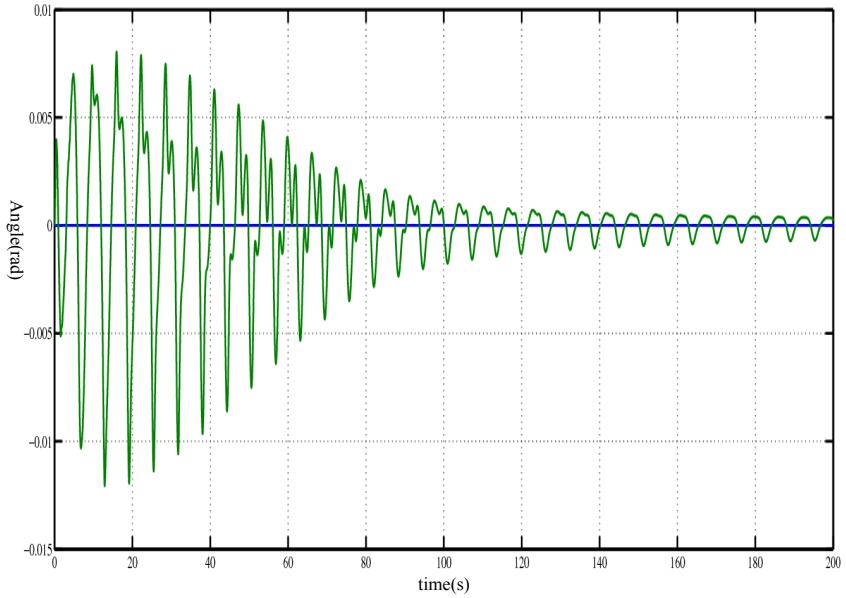


Figure 8.4: Hover maneuver ϕ plot: Blue = without disturbance, green = with disturbance

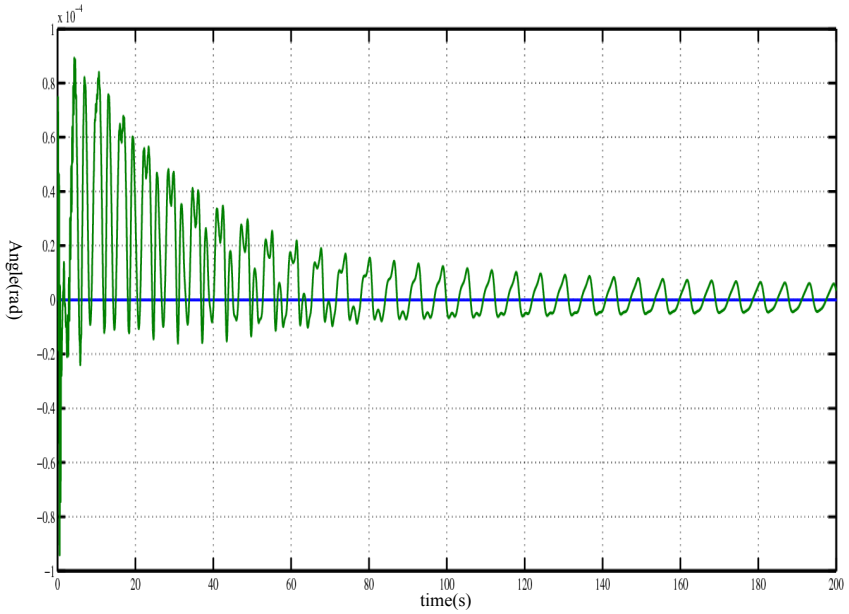


Figure 8.5: Hover maneuver θ plot: Blue = without disturbance, green = with disturbance

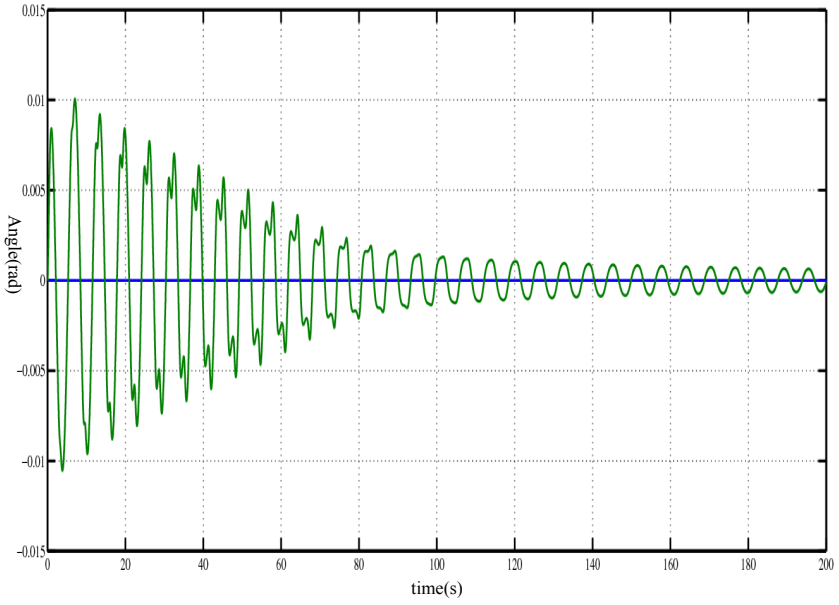


Figure 8.6: Hover maneuver ψ plot: Blue = without disturbance, green = with disturbance

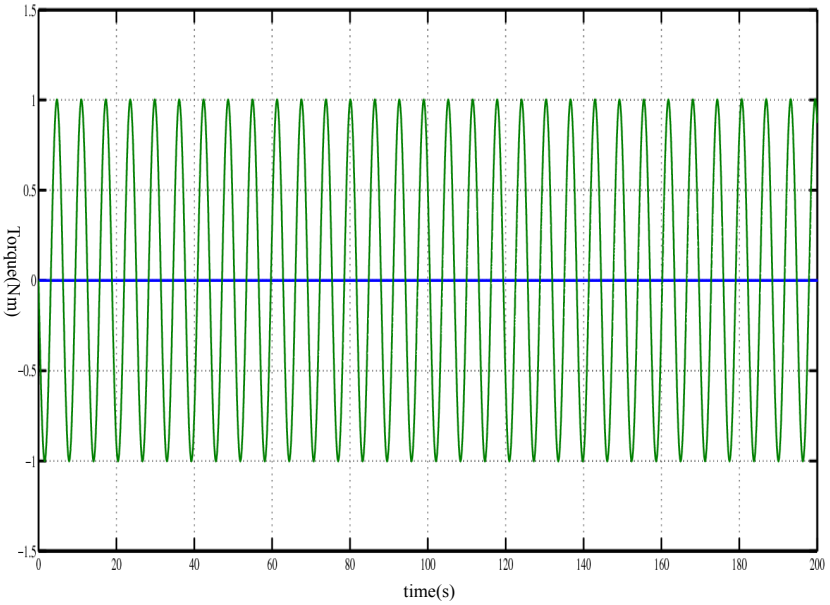


Figure 8.7: Hover maneuver τ_ϕ plot: Blue = without disturbance, green = with disturbance

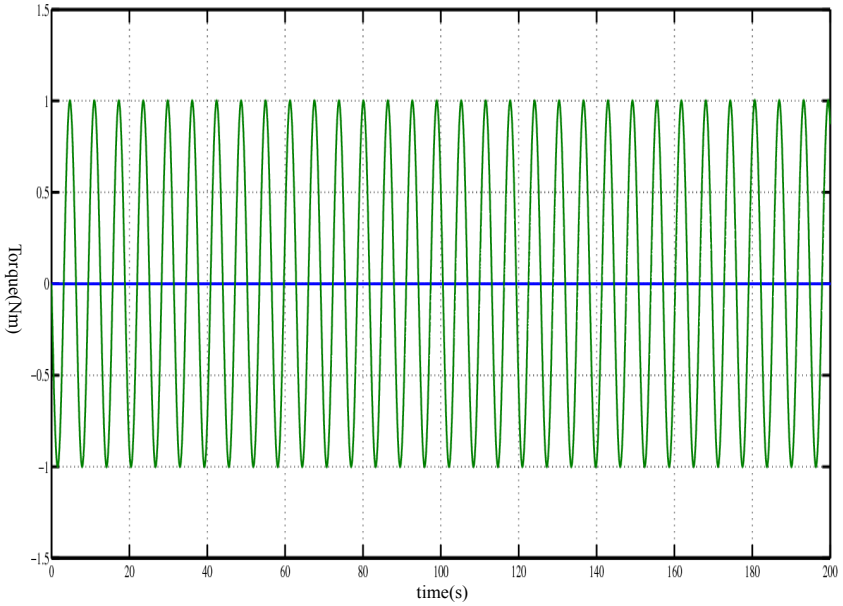


Figure 8.8: Hover maneuver τ_θ plot: Blue = without disturbance, green = with disturbance

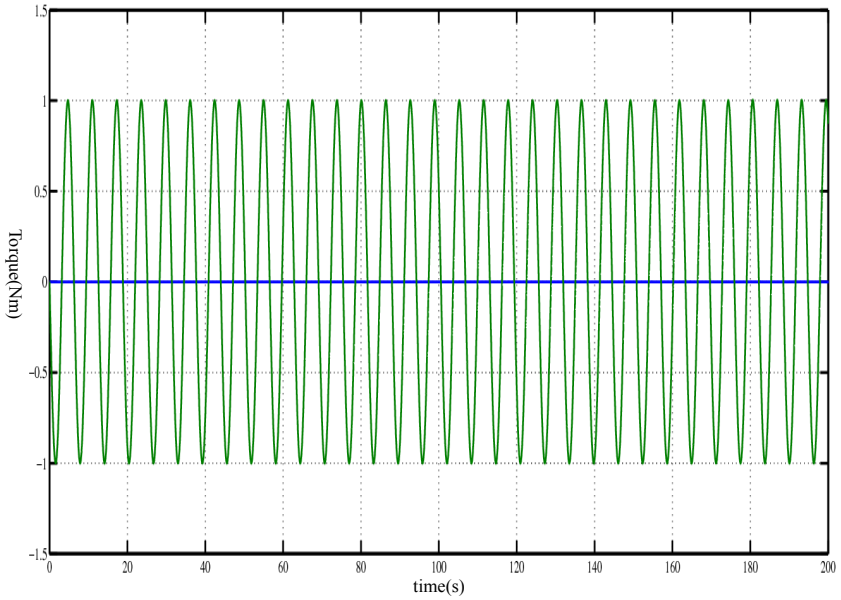


Figure 8.9: Hover maneuver τ_ψ plot: Blue = without disturbance, green = with disturbance

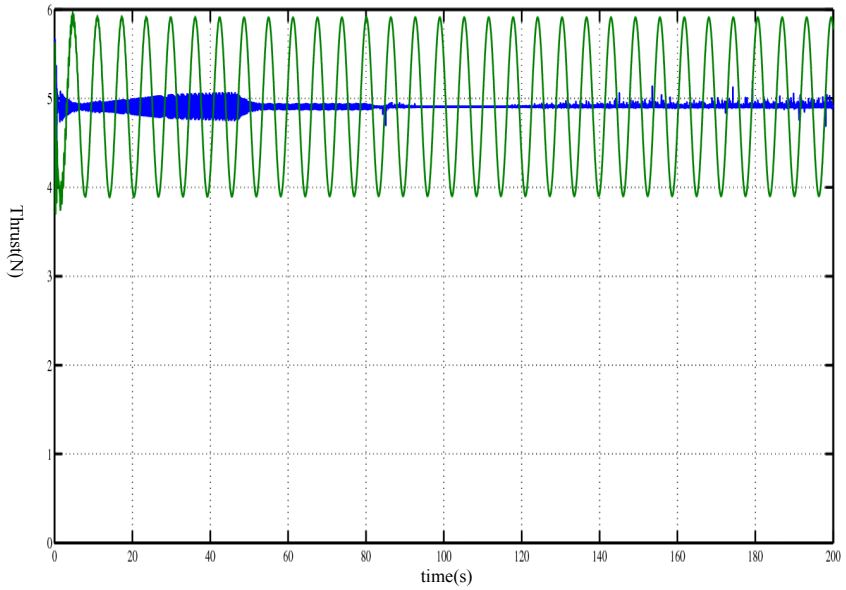


Figure 8.10: Hover maneuver Thrust plot: Blue = without disturbance, green = with disturbance

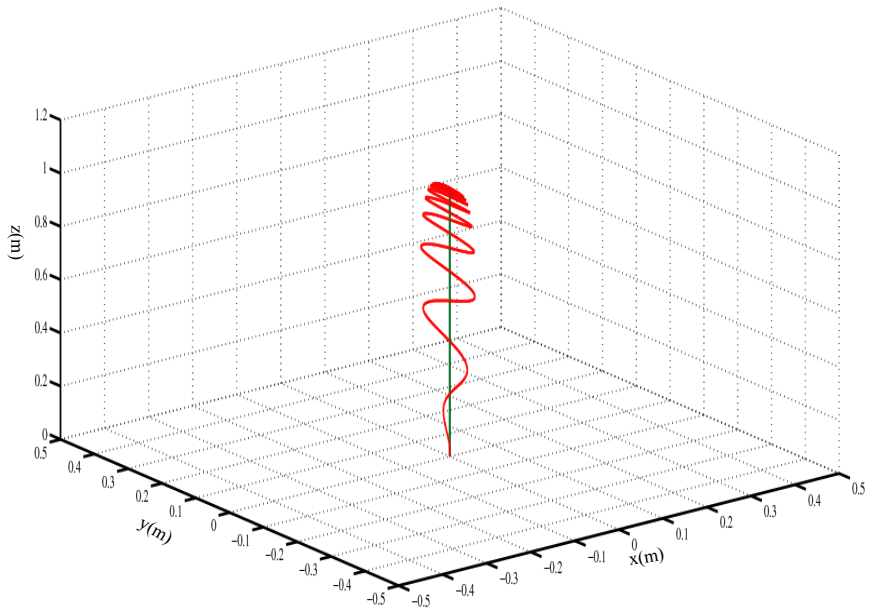


Figure 8.11: Hover maneuver 3D plot: Green = without disturbance, red = with disturbance

8.2.2 Figure 8 Maneuver

For the figure 8 maneuver the reference trajectory is given by[1]:

$$p_d^I(t) = \begin{pmatrix} 0 & 0 & -(1-\exp(\frac{t}{10})) \end{pmatrix}^T \quad \text{for } t < 50$$

$$p_d^I(t) = \begin{pmatrix} sin(\frac{\pi(t-50)}{150}) \\ sin(\frac{\pi(t-50)}{37.5}) \\ -(1-\exp(\frac{t}{10})) \end{pmatrix} \quad \text{for } t \geq 50$$

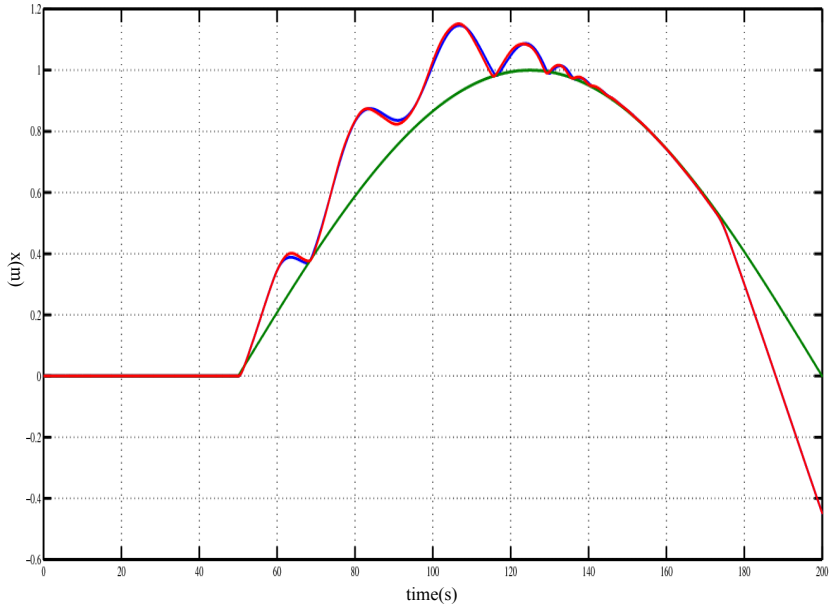


Figure 8.12: Figure 8 maneuver x plot: Green = reference trajectory, blue = without disturbance, red = with disturbance

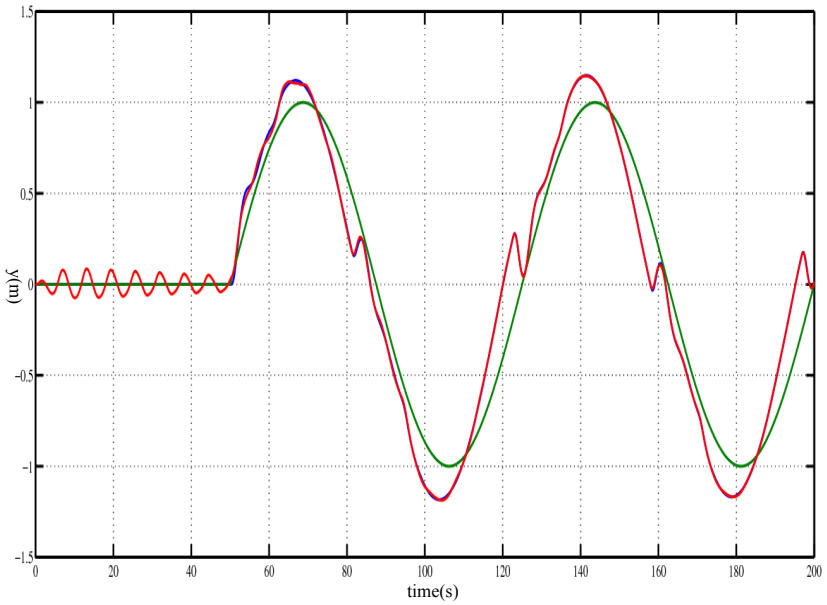


Figure 8.13: Figure 8 maneuver y plot: Green = reference trajectory, blue = without disturbance, red = with disturbance

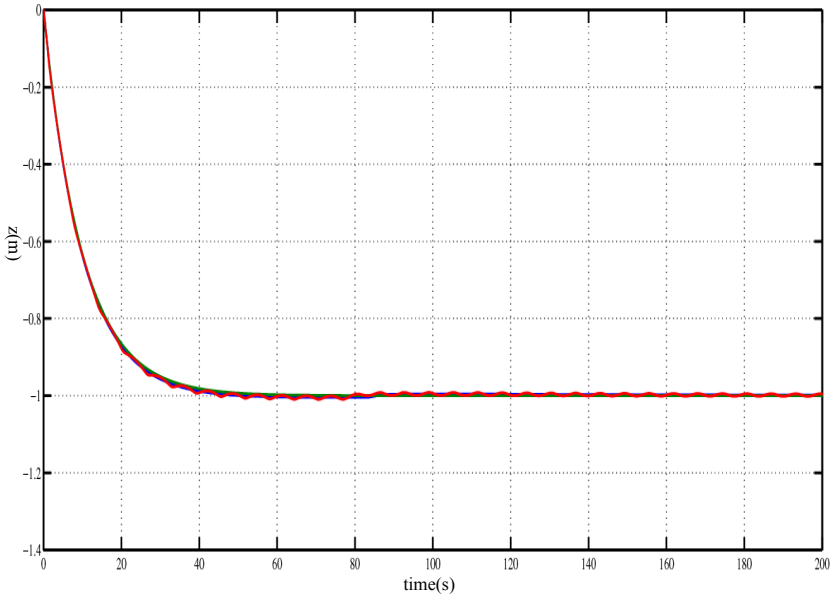


Figure 8.14: Figure 8 maneuver z plot: Green = reference trajectory, blue = without disturbance, red = with disturbance

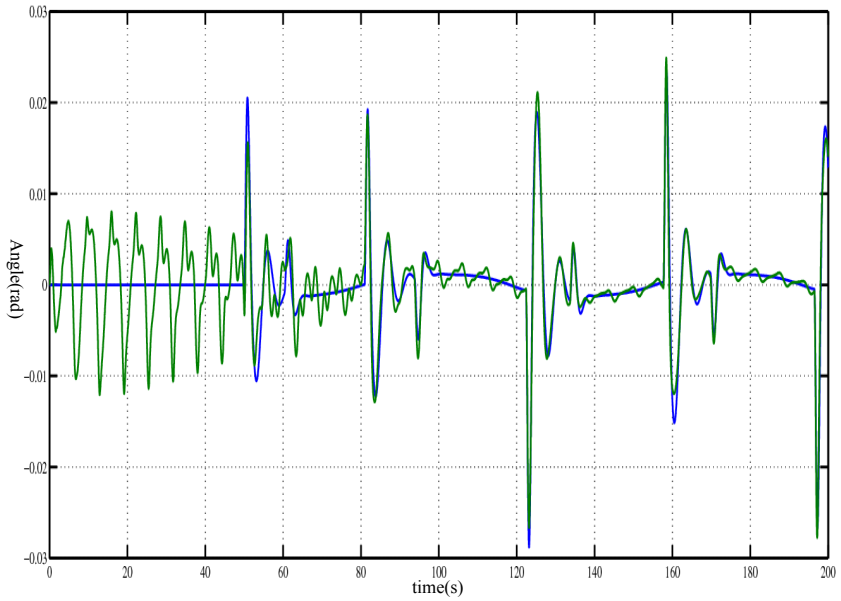


Figure 8.15: Figure 8 maneuver ϕ plot: Blue = without disturbance, green = with disturbance

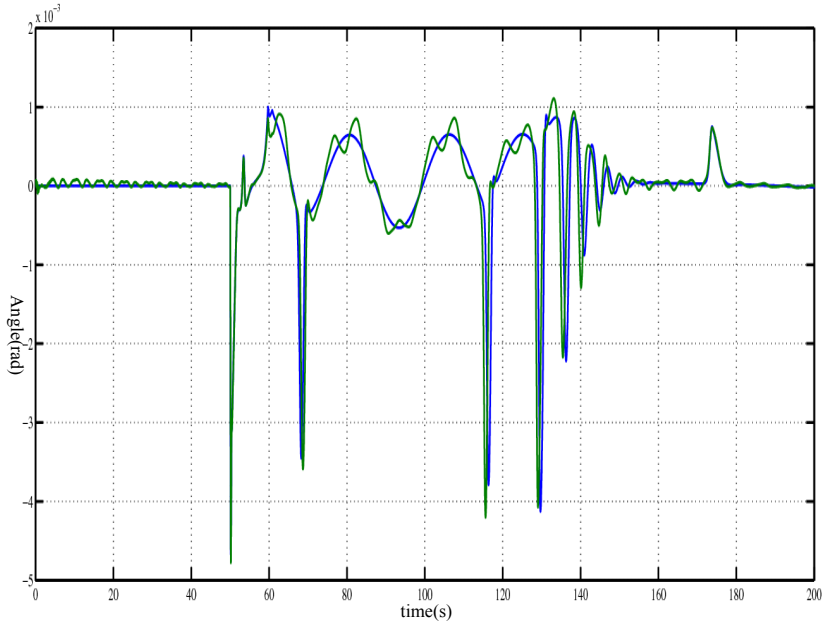


Figure 8.16: Figure 8 maneuver θ plot: Blue = without disturbance, green = with disturbance

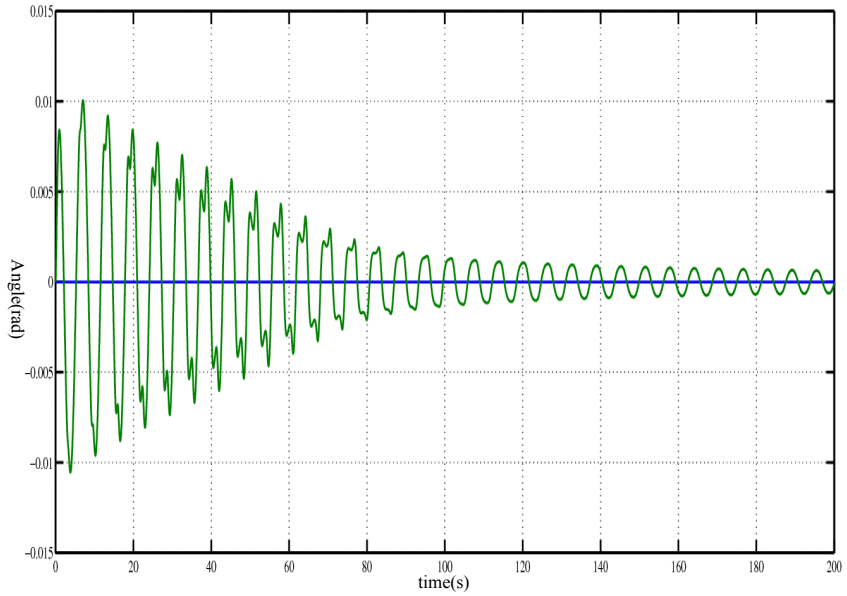


Figure 8.17: Figure 8 maneuver ψ plot: Blue = without disturbance, green = with disturbance

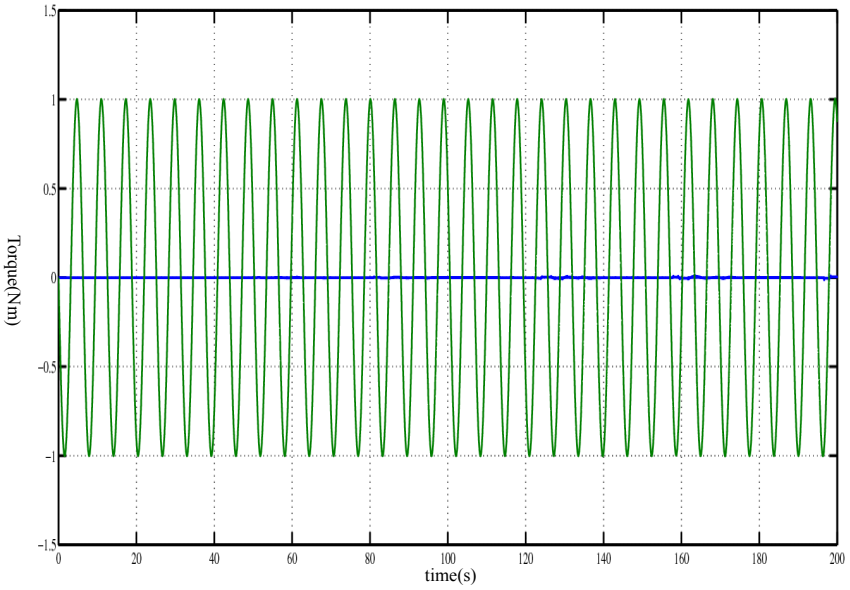


Figure 8.18: Figure 8 maneuver τ_ϕ plot: Blue = without disturbance, green = with disturbance

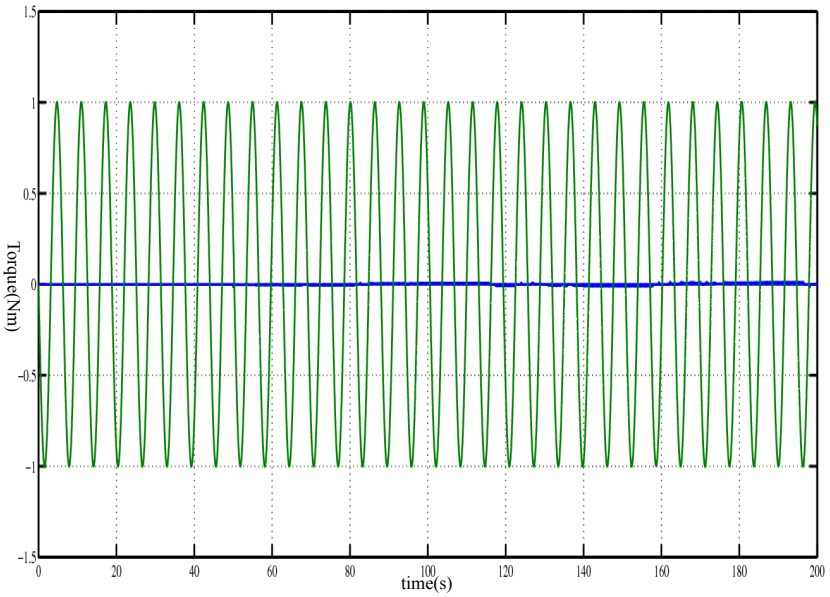


Figure 8.19: Figure 8 maneuver τ_θ plot: Blue = without disturbance, green = with disturbance

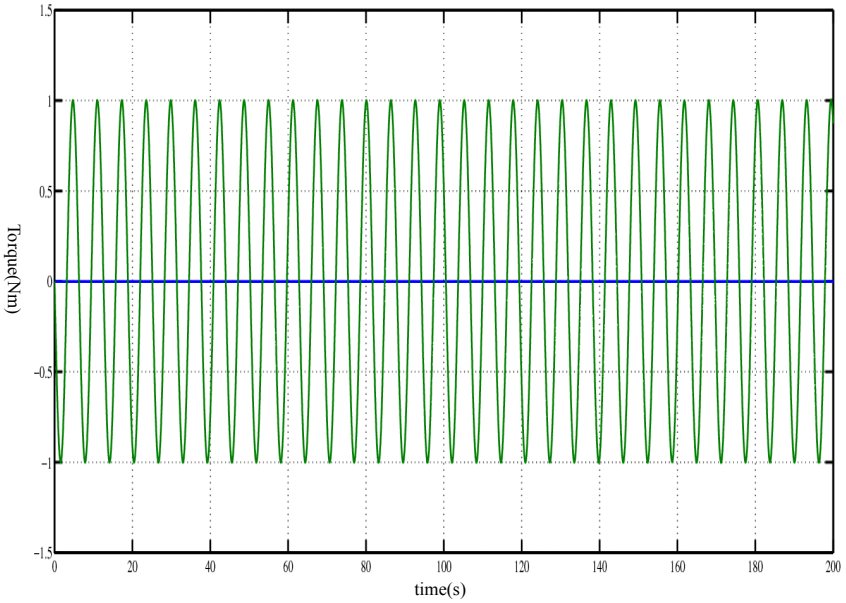


Figure 8.20: Figure 8 maneuver τ_ψ plot: Blue = without disturbance, green = with disturbance

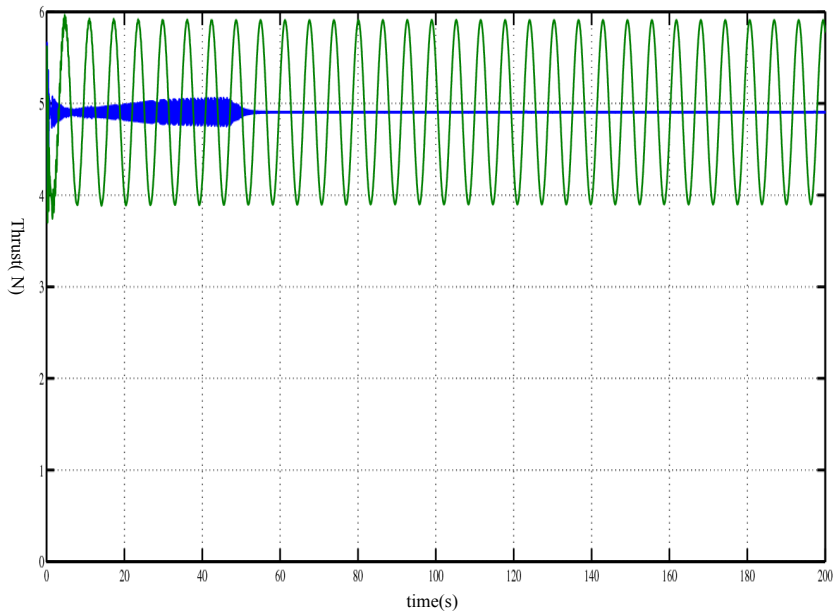


Figure 8.21: Figure 8 maneuver Thrust plot: Blue = without disturbance, green = with disturbance

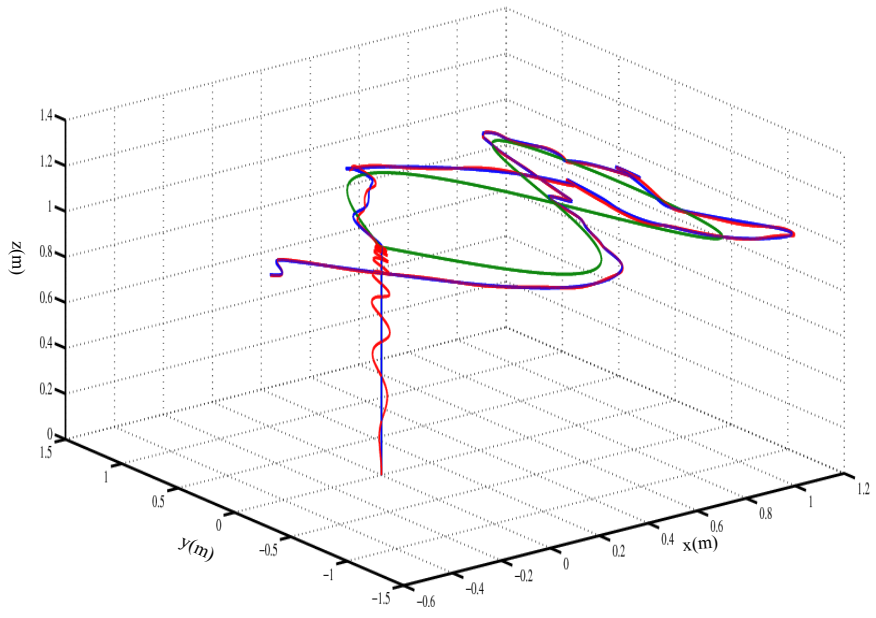


Figure 8.22: Figure 8 maneuver 3D plot: Green = reference trajectory, blue = without disturbance, red = with disturbance

From the results shown in figure 8.12-8.22 it is shown that the controller tracks the reference trajectory in an acceptable way. It should be noted that the gains that are used for these simulations are not optimised values and thus do not represent the best performance of the controller. Despite this the results do show the robustness of the controller as the response does not differ much for the case when disturbances are present and when they are absent.

8.3 Conclusion

In this chapter simulation results are presented which serve to demonstrate the effectiveness of the controllers that have developed in this work. The controller that is demonstrated in this chapter combines the translational controller of chapter 5 and the adaptive sliding backstepping attitude controller of chapter 7. Simulation results of this controller show that the controller does exhibit the expected robustness and tracking qualities.

Chapter 9

Conclusions and Recommendations

9.1 Conclusions

A robust nonlinear trajectory tracking controller has been developed for a quadrotor UAV in this work. By taking advantage of the strict feedback inter-connection of the translational and orientation subsystems a backstepping based high level control strategy is devised for the quadrotor system. The controller is divided into two sub-controllers a translational and an attitude controller.

For the translational subsystem a novel robust bounded controller is developed based on the result of A.R Teel. This is achieved by adding to the nonlinear saturated controller of Teel sliding mode like terms. The boundedness of the controller perfectly captures the limited nature of the thrust that can be developed by the quadcopter's rotors. Also the boundedness of the controller allows for the development of conditions on the gain parameters which ensure that the quadcopter does not overturn during flight. This requirement is necessary because the Euler angle parameterisation which is used will not be defined if the quadcopter is upside down. Simulations of this robust bounded controller showed that compared to A.R. Teel's control law it performed better with regards to disturbance rejection.

Two controllers were developed for the attitude subsystem. The first control law is based on the adaptive backstepping technique. The formulation and

application of the adaptive backstepping control technique is rigorously covered especially with regards to cases in which the uncertainty is both matched and unmatched. To improve on the performance of this technique a modification via nonlinear damping is presented. Integrating the adaptive backstepping controller and nonlinear damping provides a controller that can guarantee bounded error tracking in the presence of parametric and unmodeled uncertainties.

To improve on the performance of the adaptive backstepping attitude controller an alternative controller is developed in chapter 7 in which backstepping techniques are combined with sliding mode control methods. The advantage of this approach is that theoretically it can guarantee that the tracking error can tend to zero asymptotically which is better than the bounded tracking error guarantee of the adaptive backstepping based controller. However sliding mode control has two major drawbacks which are the chattering of the sliding controller and also the requirement of *a priori* knowledge of the uncertainty upper bounds. To eliminate the chattering effect continuous approximations of the signum function are used. The relaxation of the upper bounds requirement is achieved by making use of adaptive techniques to provide real time estimation of the upper bounds. Conventional parameter adaptation laws were successfully applied but despite achieving successful tracking without knowledge of the uncertainty upper bounds this adaptation law was found to over estimate the upper bound leading to unnecessarily large control effort. To improve its performance modifications are made to the adaptation law such that it gives the smallest estimate which makes the system stable. The combination of backstepping control and sliding mode control is achieved via two approaches one is a Lyapunov based technique while the other is based on the right choice of a sliding manifold. It is shown using simulations that the two approaches give similar responses but the method based on choosing a sliding manifold is found to result in simpler controls with a reduced number of gain parameters. Finally the developed adaptive sliding backstepping control is applied to the attitude control problem and simulation results show that this control law ensures asymptotic tracking even in the presence of disturbances.

Having developed the controllers for the translational and orientation subsystems the controllers are integrated to form the complete quadcopter trajectory

controller. Simulation results of the integrated system which are presented in chapter 8 show that the controller exhibits robustness as was expected providing good trajectory tracking for the quadcopter system.

The lack of any experimental verification might appear as a huge over sight in the development of this work. It should be noted however that development of the necessary hardware platform for testing is in itself a major undertaking. A typical UAV testbed would require a high performance vision based motion capture system to enable location of the UAV. Also proprioceptive sensor systems comprising of an assortment of sensors such as accelerometers, gyroscopes, barometers , magnetometers and cameras would be needed to make the UAV aware of its environment. The multitude of sensors would require development of advanced fusion algorithms such as kalman filters, particle filters e.t.c. It should be evident then that the development of a UAV testbed presents a formidable challenge which is best tackled as an independent research in its own right. Its the author's strong believe that the simulation results provided do serve to adequately verify the feasibility of the methods that have been developed in this work.

9.2 Recommendations

The results of simulations for the controllers developed in this work have given good results. However in order to fully test the feasibility of the proposed controllers a quadcopter testbed needs to be constructed to provide physical verification of the results we have developed. The development of such a testbed will allow the extension of this work to involve such areas as cooperative control of a team of UAVs, SLAM based UAV control and sensor fusion for UAV navigation.

Euler angles that were used in this work introduce restrictions in the kind of manoeuvres that the quadcopter can execute due to their local validity. As such the developed controller does not cater for aerobatic manoeuvres such as flips. As an extension to the work already done investigations into tracking aerobatic trajectories can be done by considering for example using several Euler angle sequences to represent the attitude or considering the use of quaternions. Another

approach that can be investigated to achieving aerobatic manoeuvering is to use geometric control methods which involves coordinate free representation of the UAV dynamics.

Finally another possible line of extension of the work presented here is in developing a fully bounded controller for the quadcopter system. In this work bounded control is used for the translational subsystem only however it should be easy to see that if the thrust is bounded so also will be the torque such that it makes sense to require not just the translational controller but also the attitude controller to be bounded.

Appendix A

In this Appendix the rotation matrix for the Z-Y-X Euler angle representation is derived.

.1 Vehicle Frame

Consider the setup where the body fixed frame is such that its axes are aligned with the inertial axes as is depicted in figure 1 below.This shall be denoted as the vehicle frame(F_v).

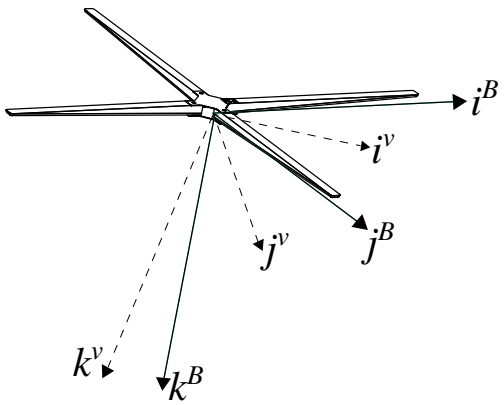


Figure 1: Vehicle and body fixed frame

.2 Vehicle-1 Frame(F^{v1})

The vehicle-1 frame is the result of rotating the vehicle frame about the Z axis by the yaw angle(ψ) such that if the pitch and roll were zero then $\hat{i}^{v1}$ would point in the $\hat{i}^{B}$ direction. The transformation matrix from F^{v} to F^{v1} is given by:

$$R_{v}^{v1}(\psi) = \begin{pmatrix} cos\psi & sin\psi & 1 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

.3 Vehicle-2 Frame(F^{v2})

The vehicle-2 frame is obtained by rotating the vehicle-1 frame about the $\hat{j}^{v1}$ axis by the pitch angle θ such that if the roll angle was zero $\hat{j}^{v2}$ would point in the $\hat{j}^{B}$ direction. The transformation matrix from F^{v1} to F^{v2} is given by :

$$R_{v1}^{v2}(\theta) = \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \tag{2}$$

.4 Body Frame F^{b}

The body frame is finally obtained by rotating the vehicle-2 frame about the $\hat{i}^{v2}$ axis by the roll angle ϕ such that $\hat{i}^{b}$ points in the airframe nose direction, $\hat{j}^{b}$ points out the right wing and $\hat{k}^{b}$ points downwards. The transformation matrix from F^{v2} to F^{b} is given by :

$$R_{v2}^{b}(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{pmatrix} \tag{3}$$

The transformation from the vehicle frame to the body frame is the composite of all the above rotation and is given by:

$$
\begin{aligned}
R_v^b(\phi,\theta,\psi) &= R_{v2}^b(\phi)\,R_{v1}^{v2}(\theta)\,R_v^{v1}(\psi) \\
&= \begin{pmatrix} cos\psi & sin\psi & 1 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix}
\begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{pmatrix} \\
&= \begin{pmatrix} cos\theta cos\psi & cos\theta sin\psi & -sin\theta \\ sin\phi sin\theta cos\psi - cos\phi sin\psi & sin\phi sin\theta sin\psi + cos\phi cos\psi & sin\phi cos\theta \\ cos\phi sin\theta cos\psi + sin\phi sin\psi & cos\phi sin\theta sin\psi - sin\phi cos\psi & cos\phi cos\theta \end{pmatrix}
\end{aligned}
$$

The inverse transformation from the body frame to the vehicle frame is given by:

$$
R_b^v(\phi,\theta,\psi) = \begin{pmatrix} cos\psi cos\theta & cos\psi sin\theta sin\phi - sin\psi cos\phi & sin\psi sin\phi + cos\psi sin\theta cos\phi \\ sin\psi cos\theta & cos\psi cos\phi + sin\psi sin\theta sin\phi & sin\psi sin\theta cos\phi - sin\phi cos\psi \\ -sin\theta & cos\theta sin\phi & cos\theta cos\phi \end{pmatrix}
$$

References

[1] I. Raptis and K. Valavanis, *Linear and Nonlinear Control of Small Scale Helicopters*. Springer, 2011. ix, 10, 16, 21, 25, 124

[2] Jean-Jacque E. Slotine and Weiping Li, *Applied Nonlinear Control*. Prentice Hall, 1991. x, 10, 58, 88, 89, 91, 92

[3] Sperry, Elmer Ambrose, *Encyclopaedia Britannica, Ultimate Reference Suite*. Chigaco : Encyclopaedia Britannica, 2013. 1

[4] Lt. Kendra L. B. Cook, "The Silent Force Multiplier: The History and Role of UAVs in Warfare," in *IEEE Aerospace Conference*, 2007. 1

[5] National Museum of the US Air Force, "Ryan-BQM-34 FIREBEE," http://www.nationalmuseum.af.mil/factsheets/factsheet.asp?id=3363, October 2013. 1

[6] Northrop Grumman, "Media resources," http://www.northropgrumman.com/MediaResources/Pages/PhotoGallery.aspx, October 2013. 2

[7] General Atomics, "Image Library," http://media.ga.com/image-library/unmanned-aircraft-systems/aircraft-platforms/, October 2013. 2

[8] Zak Sarris, "Survey of UAV Applications in Civil Markets (June 2001)," in 9^{th} *IEEE Mediterranean Conference on Control and Automation*, 2001. 2

[9] Hai Chen, Xin-min Wang, and Yan Li, "A Survey of Autonomous Control of UAVs," in *International Conference on Artificial Intelligence and Computational Intelligence*, 2009. 2

[10] Gabe Hoffman, Dev G. Rajnarayan, Steven L. Waslander, David Dostal, Jung Soon Jang, and Claire J. Tomlin, "The Stanford Testbed f Autonomous Rotorcraft for Multi-Agent Control," in *The* 23^{rd} *Digital Avionics Systems Conference*, 2004. 3

[11] Samir Bouabdalah and Roland Siegwart, "Full Control of a Quadrotor," in *IEE/RSJ International Conference on Intelligent Robots and Systems*, 2007. 3

[12] Nathan Michael, Daniel Mellinger, Quentin Lindsey, and Vijay Kumar, "The GRASP Multiple Micro-UAV Test Bed," *IEEE Robotics and Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010. 3, 8, 9

[13] Jack B. Kuipers, *Quaternions and Rotation Sequences*. Princeton University Press, 1999. 6

[14] Herbert Goldstein, *Classical Mechanics*, 2nd ed. Addison-Wesley Publishing Company, Inc, 1981. 6

[15] Eric W. Weisstein, "Cayley-Klein parameters," http://mathworld.wolfram.com/Cayley-KleinParameters.html, October 2013. 6

[16] Eric Weisstein, "Euler parameters," http://mathworld.wolfram.com/EulerParameters.html, October 2013. 6

[17] Taeyoung Lee, Melvin Leok, and Harris N. McClamroch, "Geometric Tracking Control of a Quadrotor UAV on SE(3)," in 49^{th} *IEEE Conference on Decision and Control*, 2010. 7

[18] Grant R. Fowles and George L. Cassiday, *Analytical Mechanics*, 6^{th} ed. Fort Worth: Saunders College Publishing, 1999. 7, 24

[19] D. Padfield, *Helicopter Flight Dynamics*. Blackwell Science: Oxford, 1996. 7

[20] R.W. Prouty, *Helicopter Performance, Stability and Control*. Krieger Publishing Company, 1990. 7

[21] R. Olfati-Saber, "Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles," Ph.D. dissertation, Department of Electrical Engineering and Computer Science, MIT, 2001. 7

[22] Hoger Voos, "Nonlinear State-Dependent Riccati Equation Control of a Quadrotor UAV," in *IEEE International Conference on Control Applications*, 2006. 7

[23] L.A. Sanchez, O. Santos, H. Romero, S. Salazar, and R. Lozano, "Nonlinear and Optimal Real-Time Control of a Rotary Wing UAV," in *American Control Conference*, 2012. 7

[24] Li-Chun Lai, Chi-Ching Yang, and Chia-Ju Wu, "Time-Optimal Control of a Hovering Quad-Rotor Helicopter," *Journal of Intelligent and Robotic Systems*, vol. 45, pp. 115–135, 2006. 7

[25] Younes M. Al-Younes and Mohammed A. Al-Jarrah, "Linear vs Nonlinear Control Techniques for a Quadrotor Vehicle," in 7^{th} *International Symposium on Mechatronics and its Applications for a Quadrotor*, 2010. 8, 11

[26] Samir Bouabdallah, Andre Ńoth, and Roland Siegwart, "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004. 8

[27] Yong Zeng, Qiang Jiang, Qiang Liu, and Hua Jing, "PID vs MRAC Control Techniques Applied to a Quadrotor's Attitude," in 2^{nd} *Conference on Instrumentation, Measurement, Computer, Communication and Control*, 2012. 8

[28] Hao Liu, Geng Lu, and Yisheng Zhong, "Robust LQR Attitude Control of 3-DOF Laboratory Helicopter for Aggressive Maneuvers," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 10, 2013. 8

[29] Moussa Boukhnifer, Ahmed Chaibet, and Cherif Larouci, "Experimental H_∞ Robust Control of Aerial Vehicle Flight," in 19^{th} *Mediterranean Conference on Control and Automation*, 2011. 8, 9

[30] Mingyun Lv, Yanpeng Hu, and Peng Liu, "Attitude Control for Unmanned Helicopter using H_∞ Loop Shaping Method," in *International Conference on Mechatronic Science, Electrical Engineering and Computer*, 2011. 8

[31] M. Chen and M. Huzmezan, "A Combined MBPC/2 DOF H_∞ Controllers for a Quadrotor UAV," in *IEEE Conference on Decision and Control*, 2007. 9

[32] M. Huzmezan and M. Chen, "A Simulation Model and H_∞ Loop Shaping Control of a Quad Rotor Unmanned Air Vehicle," in *IASTED Conference on Modeling, Simulation and Optimization*, 2003. 9

[33] Yibo Li and Shuxi Song, "A survey of Control Algorithms for QuadRotor Unmanned Helicopter," in *IEEE* 5^{th} *International Conference on Advanced Computational Intelligence*, 2012. 9

[34] Kostas Alexis, George Nikolakopoulos, and Anthony Tzes, "Constrained-Control of a Quadrotor Helicopter for Trajectory Tracking under Wind Gusts Disturbances," in 15^{th} *Mediterranean Electrotechnical Conference*, 2010. 9

[35] Andrew R. Teel, "Global Stabilization and Restricted Tracking for Multiple Integrators with Bounded Controls," *Systems and Control Letters*, vol. 18, 1992. 9, 34, 36, 37

[36] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a four rotor mini rotorcraft," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 510–516, 2004. 10

[37] F. Kendoul, D. Lara, I. Fantoni, and R. Lozano, "Nonlinear control for systems with bounded inputs: Real-time embedded control applied to UAVs," in 45^{th} *IEEE Conference on Decision and Control*, 2006. 10

[38] Bin Zhou, Guangren Duan, and Liu Zhang, "Nonlinear control for global stabilization of multiple-integrator system by bounded controls," *Journal of Control Theory Applications*, vol. 6, no. 3, pp. 293–299, 2008. 10

[39] E. N.Johnson and S. K. Kannan, "Nested Saturation with Guaranteed Real Poles," in *American Control Conference*, 2003. 10, 40

[40] N.Marchand and A. Hably, "Improving the performance of nonlinear stabilization of multiple integrators with bounded controls," in *IFAC World Congress*, 2005. 10

[41] Ahmad Hably and Nicolas Marchand, "Global stabilization of a four rotor helicopter with bounded inputs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007. 10

[42] Samir Boubdallah and Roland Siegwart, "Backstepping and Sliding Mode Techniques Applied to an Indoor Micro Quadrotor," in *IEEE International Conference on Robotics and Automation*, 2005. 10, 11

[43] F. Sharifi, M. Mirzaeiand, and Y. Zhang, "Fault Tolerant Control of a Quadrotor UAV using Sliding Mode Control," in *Conference on Control and Fault Tolerant Systems*, 2010. 10

[44] L. Besnard, Y.B Shtsessel, and B. Landrum, "Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer," *Journal of the Franklin Institute*, pp. 1–27, 2011. 10

[45] Rong Xu and Umit Ozguner, "Sliding Mode Control of a Quadrotor Helicopter," in 45^{th} *IEEE Conference on Decision and Control*, 2006. 10

[46] D. Munoz and D. Sbarbaro, "An adaptive sliding mode controller for discrete nonlinear systems," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 3, pp. 574–581, 2000. 10

[47] C.W. Tao, M.L. Chan, and T.T Lee, "Adaptive fuzzy sliding mode controller for linear systems with mismatched time varying uncertainties," *IEEE Transactions on Systems, Man and Cybernetics - Part B Cybernetics*, vol. 33, pp. 283–294, 2003. 10, 96

[48] Ying-Jeh Huang, Tzu-Chun Kuo, and Shin-Hung Chang, "Adaptive sliding-mode control for nonlinear sytems with uncertain parameters," *IEEE Transactions on Systems, Man and Cybernetics-Part B Cybernetics*, vol. 38, pp. 534–539, 2008. 10, 96

[49] H. Lee and V.I. Utkin, "Chattering suppression methods in sliding control systems," *Annual Reviews in Control*, vol. 31, pp. 179–188, 2007. 11

[50] A. Saberi, P.V. Kokotovic, and H.J. Sussman, "Global Stabilization of Partially Linear Composite Systems," in 28^{th} *Conference on Decision and Control*, 1989. 11, 55

[51] I. Kanellakopoulos, P.V. Kokotovic, and A.S. Morse, "A toolkit for nonlinear feedback systems," *Systems and Control Letters*, vol. 18, pp. 83–92, 1992. 11, 55

[52] Guilherme V. Raffo, Manuel G. Ortega, and Fransisco R. Rubio, "Backstepping/Nonlinear H_∞ Control for Path Tracking of a Quadrotor Unmanned Aerial Vehicle," in *American Control Conference*, 2008. 11

[53] Tarek Madani and Abdelaziz Benallegue, "Backstepping Control for a Quadrotor Helicopter," in *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2006. 11

[54] K.M. Zemalache, L. Beji, and H. Marref, "Control of an underactuated system: Application to a four rotor rotorcraft," in *IEEE International Conference on Robotics and Biomimetics*, 2005. 11

[55] Samir Bouabdallah and Roland Siegwart, "Full Control of a Quadrotor," in *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2007. 11

[56] Ashfaq Ahmad Mian and Wang Daobo, "Modelling and Backstepping based Nonlinear Control Strategy for a 6DOF Quadrotor Helicopter," *Chinese Journal of Aeronautics*, vol. 21, pp. 261–268, 2008. 11

[57] D.Taylor, P.V. Kokotovic, R. Marino, and I. Kanellakopoulos, "Adaptive regulation of nonlinear systems with unmodeled dynamics," *IEEE Transactions on Automatic Control*, vol. 34, pp. 405–412, 1989. 11, 65

[58] I. Kanellakopoulos, P.V. Kokotovic, and D. Marino, "An extended direct scheme for robust adaptive nonlinear control," *Automatica*, vol. 27, pp. 247–255, 1991. 11, 65

[59] M. Kristic, I. Kanellakopoulos, and P. Kokotovic, "Adaptive nonlinear control without overparameterisation," *Systems and Control Letters*, vol. 19, pp. 177–185, 1992. 11, 70

[60] Suseong Kim, Deawon Lee, and H.Jin Kim, "Image Based Visual Servoing for Autonomous Quadrotor with Adaptive Backstepping Control," in 11^{th} *International Conference on Control, Automation and Systems*, 2011. 11

[61] Thilina Fernando, Jiten Chandiramani, and T. Lee and Hector Gutierrez, "Robust Adaptive Geometric Controls on SO(3) with an Application to the Attitude Dynamics of a Quadrotor UAV," in 50^{th} *IEEE Conference on Decision and Control and European Control Conference*, 2011. 12

[62] M. Rios-Bolivar, A.S.I Zinober, and H. Sira-Ramirez, "Dynamical Sliding Mode Control via Adaptive Input-Output Linearization: A Backstepping Approach," in *Robust Control via Variable Structure and Lyapunov Techniques*, F. Garofalo and L. Gliemlo, Eds. Springer-Verlag, 1996. 12

[63] Ali J. Koshkouei and Alan S I. Zinober, "Adaptive Backstepping Control of Nonlinear Systems with Unmatched Uncertainty," in 39^{th} *IEEE Conference on Decision and Control*, 2000. 12

[64] Loring W. Tu, *An Introduction to Manifolds*, 2^{nd} ed. Springer, 2010. 16

[65] Bob Palais and Richard Palais, "Euler's fixed point theorem: The axis of rotation," http://vmm.math.uci.edu/PalaisPapers/EulerFPT.pdf, October 2013. 16

[66] Martin D. Crossley, *Essential Topology.* Springer, 2005. 17

[67] Marcello R. Napolitano, *Aircraft Dynamics From Modeling to Simulation.* John Wiley and Sons, Inc., 2012. 19

[68] Mark W. Spong, Seth Hitchinson, and M. Vidyasagar, *Robot Modeling and Control.* John Wiley and Sons, Inc, 2006. 22

[69] B. Mettler, *Identification, Modeling and Characteristics of Miniature Rotorcraft.* Kluwer Academic Publishers, Norwell, 2003. 25

[70] Robert Mohoney, Vijay Kumar, and Peter Corke, "Multirotor Aerials Vehicles Modeling, Estimation and Control of Quadrotor," *IEEE Robotics and Automation Magazine*, September 2012. 27

[71] H.J. Sussman, Y. Yang, and H.J. Sussman, "On the Stabilizability of Multiple Integrators by means of Bounded Feedback Controls," in *IEEE Conference on Decision and Control*, 1991. 36

[72] Nicolas Marchand, "Further Results on Global Stabilization for Multiple Integrators with Bounded Controls," in 42^{nd} *IEEE Conference on Decision and Control*, 2003. 40, 41

[73] M. Kristic, I. Kanellalopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. John Wiley and Sons,Inc., 1995. 56, 58, 60, 66, 70

[74] A. Feuer and A.S. Morse, "Adaptive control of single input single output linear systems," *IEEE Transactions on Automatic Control*, vol. 23, pp. 557–568, 1978. 74

[75] B.R. Barmish, M. Corless, and G. Leitmann, "A class of stabilizing controllers for uncertain dynamical systems," *SIAM Journal of Optimization and Control*, vol. 21, pp. 246–255, 1983. 74

[76] E. Sontag, "Smooth stabilization implies coprime factorization," *IEEE Transactions on Automatic Control*, vol. 34, pp. 435–443, 1989. 74

[77] A.J. Fossard and T. Floquet, "Introduction: An Overview of Classical Sliding Mode Control," in *Sliding Mode Control In Engineering*, W. Perruquetti and Jean Pierre Barbot, Eds. Marcel Dekker Inc, 2002. 87

[78] A.F. Filipov, "Application of the theory of differential equations with discontinuous right-hand sides to non-linear problems in automatic control," in 1^{st} *IFAC International Conference*, 1961. 88

[79] R.J. Wai, "Fuzzy sliding mode control using adaptive tuning technique," *IEEE Transactions on Industrial Electronics*, vol. 54, pp. 586–594, 2007. 96

[80] H.Hu and P.Y. Woo, "Fuzzy supervisory sliding mode and neural network control for robotic manipulators," *IEEE Transactions on Industrial Electronics*, vol. 53, pp. 929–940, 2006. 96

[81] F. Plestan, Y. Shtessel, V. Bregeault, and A. Pozynak, "New methodologies for adaptive sliding mode control," *International Journal of Control*, vol. 83, pp. 1907–1919, 2010. 96, 98