Agile Flight Control Techniques for a Fixed-Wing Aircraft

by

Frantisek Michal Sobolic

B.S., Aerospace Engineering
University of Michigan (2006)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author .....
Department of Aeronautics and Astronautics
May 22, 2009

Certified by .....
Jonathan P. How
Professor
Thesis Supervisor

Accepted by .....
Prof. David L. Darmofal
Associate Department Head
Chair, Committee on Graduate Students

Agile Flight Control Techniques for a Fixed-Wing Aircraft

by

Frantisek Michal Sobolic

Submitted to the Department of Aeronautics and Astronautics
on May 22, 2009, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

As unmanned aerial vehicles (UAVs) become more involved in challenging mission objectives, the need for agility controlled flight becomes more of a necessity. The ability to navigate through constrained environments as well as quickly maneuver to each mission target is essential. Currently, individual vehicles are developed with a particular mission objective, whether it be persistent surveillance or fly-by reconnaissance. Fixed-wing vehicles with a high thrust-to-weight ratio are capable of performing maneuvers such as take-off or perch style landing and switch between hover and conventional flight modes. Agile flight controllers enable a single vehicle to achieve multiple mission objectives. By utilizing the knowledge of the flight dynamics through all flight regimes, nonlinear controllers can be developed that control the aircraft in a single design.

This thesis develops a full six-degree-of-freedom model for a fixed-wing propeller-driven aircraft along with methods of control through nonconventional flight regimes. In particular, these controllers focus on transitioning into and out of hover to level flight modes. This maneuver poses hardships for conventional linear control architectures because these flights involve regions of the post-stall regime, which is highly nonlinear due to separation of flow over the lifting surfaces. Using Lyapunov backstepping control stability theory as well as quaternion-based control methods, control strategies are developed that stabilize the aircraft through these flight regimes without the need to switch control schemes. The effectiveness of each control strategy is demonstrated in both simulation and flight experiments.

Thesis Supervisor: Jonathan P. How
Title: Professor

Acknowledgments

First, I would like to thank Raytheon Missile Systems for their support during my graduate school experience in which I am able to advance both my education and my career.

Secondly, I would like to thank Professor How for his guidance and wisdom during my research pursuits at MIT. His constant curiosity with the challenges that I faced and his helpful notions gave me the direction to successfully complete my research.

The members of the Aerospace Controls Laboratory have been extremely supportive throughout my work on this thesis. I would like to offer a special thanks to Buddy Michini, who offered continual guidance throughout my work, especially with software and hardware issues. I would also like to thank Cameron Fraser, Josh Redding, Karl Kulling, Sergio Cafarelli, Frank Fan, Samera Ponda, Dan Levine, Brett Bethke, Brandon Luders, Andrew Whitten and Kenneth Lee for the specific ways in which they have assisted me. I would also like to thank Kathryn Fischer for her support and resourcefulness throughout my time here.

During the dynamic modeling phase of my project, I received invaluable technical support from Dave Robertson, Richard Perdichizzi and Todd Billings. I would like to thank them for guidance and technical advice.

Finally, my family and friends have been a constant source of love, support and inspiration for me throughout my education. My parents, Frank and Maryanne have always been there in many ways to support my aspirations as well as my Aunt Diane and Uncle Tom who supported me and have always been interested in my educational pursuits.

Contents

1 Introduction **13**

1.1 Motivation 14

1.2 Background 16

1.2.1 Quaternion 16

1.2.2 Lyapunov Backstepping Design 16

1.3 Literature Review 17

1.4 Contributions 19

1.5 Approach 20

2 Modeling **21**

2.1 Introduction 21

2.2 Preliminaries 22

2.2.1 Nomenclature 22

2.2.2 Vehicle Description 23

2.2.3 RAVEN Testbed 24

2.3 Tests Performed 25

2.3.1 Introduction 25

2.3.2 Prop-Hang Test 26

2.3.3 Quasi-Steady State Wind Tunnel Test 29

2.4 Equations of Motion 31

2.4.1 System Identification 35

3 Quaternion Based Control **41**

3.1 Introduction 41

3.1.1 Notation 42

3.2 Inner Attitude Loop 42

3.3 Outer Velocity Loop 44

3.4 Thrust Controller 46

3.5 Results 47
3.5.1 Simulation 47
3.5.2 Decoupled Roll Control 47
3.5.3 Transition to Level-Flight 49

4 Nonlinear Lyapunov Backstepping Controller **53**
4.1 Introduction 53
4.2 Controller Outline 54
4.2.1 Simulation 59
4.3 Linearized Hover Controller 61
4.3.1 Flight Test 65
4.4 Lyapunov Quaternion Control 67
4.4.1 Introduction 67
4.4.2 Simulation 70
4.4.3 Hardware Implementation 72

5 Conclusion **81**
5.1 Future Work 83
5.1.1 Improved Dynamic Model 83
5.1.2 Trajectory Linearized Control 83
5.1.3 Path Feasibility Planner 84

A Quaternion Based Method for the Determination of Body Rates using a Motion Capture System **85**
A.1 Introduction 85
A.2 Extracting the Axis Angle 86

List of Figures

1-1 General Atomics Aeronautical Systems Predator B 14
1-2 AeroVironment Raven B UAV 14
1-3 Bell Boeing V-22 Osprey 15
1-4 Lockheed Martin X-35B Joint Strike Fighter 15

2-1 Vehicle and hardware used for controller implementation 23
2-2 Modeling test setup and load cell placement 25
2-3 Aircraft mounted with tinsel hung around control surfaces to determine propeller downwash diameter 27
2-4 Linear least square fit of the radius of the cone produced by propeller downwash 27
2-5 Estimated and actual propeller downwash flow located at the respective control surfaces 28
2-6 Moment coefficient as a function of respective control surface deflection angle 29
2-7 Wind tunnel test setup - mounted upside down due to maneuverability constraints 29
2-8 Sample wind tunnel moment data taken at 30° angle-of-attack 30
2-9 Effect of propeller downwash combined with the free-stream velocity . 31
2-10 Aircraft body and inertial coordinate frames 31
2-11 Measured versus theoretical force coefficients for various free-stream velocities 34
2-12 System Identification: compared state outputs for the sinusoidal velocity test 37
2-13 Stills from the Clik transition maneuver 39
2-14 System Identification: compared state outputs for the hover to transition test 40

3-1 Quaternion-based control system architecture 42

3-2 Vector description of additional body x-velocity necessary to obtain desired inertial velocity 46
3-3 Roll decoupling maneuver state output 50
3-4 3-D visualization of the quaternion controlled transition maneuver . . 51
3-5 Position output for the quaternion-based control transition maneuver 51
3-6 Various state results for the quaternion-based controlled transition to level-flight 52

4-1 Lyapunov-based backstepping control architecture 53
4-2 Simulated Lyapunov-based backstepping control in hover with initial condition offsets 60
4-3 Lyapunov-based backstepping take-off to hover simulation 61
4-4 Experimental linearized Lyapunov-based controller position data about hover. 66
4-5 Lyapunov quaternion control architecture 68
4-6 Sample transition trajectory data with radial basis least square fit . . 70
4-7 Simulated Lyapunov quaternion controlled hover to hover state data . 73
4-8 Simulated Lyapunov quaternion controlled hover to hover control effort 74
4-9 Hardware implemented Lyapunov quaternion controlled hover to hover state data 75
4-10 Hardware Lyapunov quaternion controlled hover to hover control effort 76
4-11 Measured y-position output for the hover to hover maneuver varying the feed-forward predictive gains 76
4-12 Elevator deflection output for the hover to hover maneuver varying the model feed-forward gains 76
4-13 Hardware implemented Lyapunov quaternion controlled take-off to hover maneuver 77
4-14 Lyapunov quaternion controlled take-off to hover state data 78
4-15 Lyapunov quaternion controlled take-off to hover control effort 79

5-1 Trajectory linearized control architecture 83

A-1 Quaternion data due to a pure rotation about the reference z-axis. . . 87
A-2 Pure body z-rotation illustrating axis flip 88
A-3 Continuous quaternion data due to a pure rotation about the reference z-axis. 89

List of Tables

2.1 Clik aircraft parameters 24
2.2 Clik aerodynamic parameters 34

3.1 Simulation quaternion attitude loop gains 47

4.1 Algorithm to convert from DCM to quaternion 69
4.2 Radial basis function values: $\sigma = 1.2$ 71

A.1 Smooth quaternion signal data algorithm 89

Chapter 1

Introduction

Unmanned aerial vehicles (UAVs) are becoming increasingly involved in challenging mission objectives including search and rescue, reconnaissance and other intelligence-gathering roles. The advantage of these vehicles is not only the absence of human presence in a volatile scenario, but also their production value relative to manned vehicles. In general, two types of UAVs are produced, remote piloted and self-piloted. Remote piloted UAVs allow an operator to control the vehicle to perform a mission objective while a self-piloted UAV performs a mission autonomously based on a set of rules preprogrammed prior to flight. Autonomous UAVs are much more complex system, however, emerging technologies would allow them to address much more complex missions.

Even under the categroy of technologically advanced aircraft, classes of UAVs are built to perform tasks based on specific mission scenarios. For instance, the General Atomics Aeronautical Systems Predator, shown in Figure 1-1, is a very common and very well known UAV designed for long-endurance, medium altitude remotely controlled surveillance and reconnaissance operations. With a wingspan of 66ft, weight of 10,000lbs and operational duration of more than 40 hours, this aircraft is able to provide both a front line soldier and operational commander real-time footage with its on-board vision system [1]. The AeroVironment Raven (Figure 1-2), also used for surveillance and reconnaissance but at low altitude, is a highly mobile, light weight aircraft that can be operated manually or programmed for autonomous operation [2].

Figure 1-1: General Atomics Aeronautical Systems Predator B

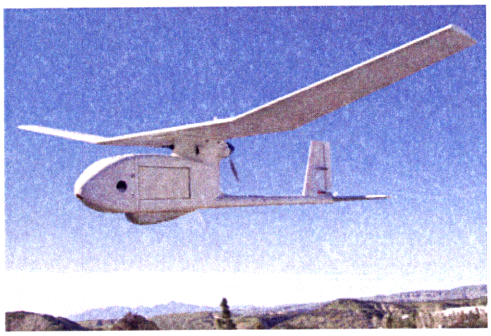


Figure 1-2: AeroVironment Raven B UAV

Each of these UAV classifications have an important role in their aerial missions. For a particular mission, multiple UAVs may be used to provide feedback on related issues from different perspectives. With these various specified aircraft systems, classifications have been made to separate vehicles by the roles which they fulfill yet vehicles that are able to perform multiple missions could declare UAV dominance. An individual UAV is only limited by the user, the amount of autonomy it is granted and the overall capabilities of the aircraft. By advancing the capabilities, thus expanding the range of mission qualifications, a single UAV may be used for multiple mission scenarios.

1.1 Motivation

An aircraft with short take-off and landing (STOL) and vertical take-off and landing (VTOL) capabilities has been an active area of research for many reasons. One of the main interests is their ability to maneuver and land in a constrained environment and still have the fuel efficiency and quickness to proceed to another location. One such vehicle is the Bell Boeing V-22 Osprey shown in Figure 1-3. It is a tiltrotor aircraft with the combined STOL and VTOL capabilities. The vehicle can take-off and land similar to a helicopter and hover at a single position above ground. Once airborne, its engine nacelles can be rotated to convert the aircraft to a turboprop airplane, capable of high-speed, high-altitude flight [3]. There are many reasons

Figure 1-3: Bell Boeing V-22 Osprey

Figure 1-4: Lockheed Martin X-35B Joint Strike Fighter

why this aircraft seems so attractive for multiple missions. Its capabilities include: transporting troops and cargo, air-to-air refueling and landing aboard an aircraft carrier compacting its storage area by retracting its rotors. Another type of vehicle is the X-35B Joint Strike Fighter produced by Lockheed Martin shown in Figure 1-4. This vehicle also has combined ability features that allow it to proceed in a short take-off and vertical landing (STOVL) manner. The nozzle, which is supplemented by two roll control ducts on the inboard section of the wing, together with the vertical lift fan provide the military required STOVL capability [1]. These combined capabilities allows the aircraft to carry a larger payload during take-off and land in constrained environments. Versatile aircraft such as these are in high demand, and it seems therefore fitting to develop UAVs with similar capabilities. UAVs have the potential to maneuver much more aggressively due to the lack of a human pilot, yet controlling them through such maneuvers remains a challenge, even today.

The UAVs that have been designed thus far are built for a mission specific scenario. For a reconnaissance mission, fixed wing vehicles are constrained to fly at speeds above stall limiting them to perform a loitering pattern for persistent surveillance. On the other hand, vehicles such as quadrotors and helicopters have a hovering ability but are hindered by their efficiency in translating from one mission location to the next. However, a vehicle designed to capture the strengths of both a fixed and rotary wing aircraft could be used in either situations and provide this sought-after versatility.

The objective of this thesis is to design a agile flight controller for a fixed wing aircraft, enabling it to follow a desired trajectory through multiple regimes of flight,

including post-stall. Such a controller will enable the aircraft to hover as well as safely transition to steady-level flight. It will also inhabit the capabilities of conventional take-off and landing as well as a perch style landing. With these combined abilities, the agile flight controller gives an aircraft the desirable characteristics of the versatile, manned vehicles mentioned previously.

1.2 Background

1.2.1 Quaternion

A quaternion is a 4-dimensional vector used to describe the transformation of a vehicle in 3-dimensions. The use of quaternions are sometimes favored over other descriptors due to their non-singularity properties at any aircraft attitude. Traditional aeronautic transformations (Euler angles), are hindered by a phenomenon known as gimbal lock. Gimbal lock causes a loss of degree of freedom (DOF) which could lead to controller instability. Since this thesis explores aggressive flight regimes, a quaternion attitude descriptor was chosen to provide a singularity-free rotation from hover to horizontal flight.

1.2.2 Lyapunov Backstepping Design

Lyapunov backstepping control provides a stable controller by developing a prominent functional candidate that satisfies the Lyapunov criteria, known as the control Lyapunov function. The backstepping technique can be summarized as follows:

1. Start with the state furthest from influential control actuators.
2. Introduce a virtual state and a control.
3. Define a control Lyapunov function.
4. Choose the virtual controller such that the control Lyapunov function satisfies the Lyapunov criteria.

5. If the virtual controller involves a control actuator, this is the control law, if not repeat Step 2 with the new virtual state.

These control Lyapunov functions "step" through the dynamics of a system leading to a control methodology that can be used to produce a desired response.

1.3 Literature Review

This research is focused on the aggressive maneuvering of UAVs in a constrained environment. Aggressive maneuvers at low speed require a special type of vehicle capable of maintaining stability and a high level of performance during unconventional missions. This section gives a historical perspective of the previous work done in the areas of aggressive and agile flight and is coupled with a discussion of the control techniques used.

A number of researchers have recently investigated the idea of developing fixed-wing aircraft with hovering capabilities. The first successful manually controlled transitions were performed in 1954 with the Convair XFY-1 "Pogo" [4]. Additionally, a custom designed, radio-controlled (R/C) airplane was developed at Drexel University [5], and possessed the capability to fly in both level-fight and hover. The airplane was manually controlled in level-fight operations and transitioned to a computer-controlled hover configuration upon user input. Successful autonomous transitions from steady level-flight to hover (and vice-versa) have also been performed by researchers at Georgia Tech on a R/C airplane [6]. Researchers from the Massachusetts Institute of Technology successfully demonstrated an autonomous fixed-wing aircraft with the capability to take-off, hover, transition to and from level-flight, and perch on a vertical landing platform. These maneuvers are all demonstrated in the highly space-constrained environment of the Real-time indoor Autonomous Vehicle test Environment (RAVEN) at MIT [7]. The developed flight control system in [7] has two linear controllers designed independently for hover and level-flight configuration. Intelligent switching between these two controllers enables the aircraft to perform transitions from level-flight to hover, and visa-versa.

The control techniques mentioned above are limited to performing in a region prescribed by the linearization method used. Full knowledge of the aircraft's dynamics, including nonlinearities, could solve potential issues of needing multiple controllers in different flight modes and a single control design could be realized. The use of nonlinear controllers provide means of control at all possible flight regimes so, nonlinear decoupling theory and dynamic inversion approaches have been applied to flight control systems [8], [9]. Unfortunately, it was shown that an inverse dynamic approach, even when the dynamics are very well known, may result in the desired linear input/output response but may also include undesirable unstable zero dynamics. Nonlinear Lyapunov-based controllers have the ability to overcome some of these issues [10–13]. In particular, the backstepping approach is used when a vehicles states are influenced through other states. This technique is demonstrated in [14] for a 6-DOF mid-altitude unmanned airship, where a simulated airship tracks a desired trajectory. They prove that the tracking error will converge exponentially to zero since the proposed controller is globally asymptotically stable. Also, [13] demonstrates the same type of trajectory-tracking capability in simulation for a hovercraft moving on a planar surface and an underwater vehicle moving in 3-D space. It is important to note that this control law assumes no parametric uncertainty, therefore an onboard estimator is implemented to predict the values of the states used for feedback in the Lyapunov control algorithm.

The work presented in this thesis follows the work of [13] and [15] to control an aircraft from hover to translational flight. In [15], Knoebel uses an adaptive quaternion-based attitude controller to maintain aircraft performance through poorly known regions of the vehicle dynamics during a transition from hover to level-flight. Gain scheduling was used based on the sensed airspeed over the control surfaces. On the other hand, [13] forms a Lyapunov backstepping controller such that all the closed-loop signals are bounded and the tracking error converges to a neighborhood of the origin that can be made arbitrarily small. It has the capability of following a prescribed trajectory solely based on the vehicles dynamics. Therefore, an accurate and complete model of the system dynamics must be known through all regions of

movement.

1.4 Contributions

Each of the following chapters provide a unique contribution to the overall goal of a transition controller, which are summarized below.

- **Chapter 2:** A full nonlinear dynamic model is derived for a specific vehicle which serves as a testbed for all controllers through simulation and hardware implementation. The process of achieving this high-fidelity model is presented through extensive wind-tunnel and static experimentation. A full system identification is performed to verify the input/output response through the use of an off-board motion capture system which provides the necessary vehicle state information.

- **Chapter 3:** A quaternion-based attitude controller is presented following the work of Ref. [15], but is modified for reference velocity tracking. The velocity error is used to provide the quaternion controller with the desired attitude in order to decrease the velocity error. Controller results are presented in simulation as well as hardware implementation. A full transition from hover to level flight and back to hover is shown.

- **Chapter 4:** A general Lyapunov backstepping technique is introduced with trajectory tracking capabilities. This technique is then applied in simulation to the derived dynamics and the results are shown. To implement this design on hardware, a modified version of Ref. [13] is used that combines the rotational rate tracking capabilities of the quaternion attitude controller, with the state to state influential approach of the backstepping design. Both simulation and hardware results are shown for various trajectories.

1.5 Approach

This thesis is structured as follows. Chapter 2 introduces the fixed-wing aircraft which serves as the vehicle testbed for all of the controllers developed. Here the equations of motion are derived and a full system identification is performed. In Chapter 3, a quaternion-based attitude controller with the ability to follow user defined velocity commands is derived. Chapter 4 introduces a Lyapunov backstepping controller with the ability to follow a user defined position trajectory modified with the quaternion controller and implemented in hardware. Finally, Chapter 5 provides concluding remarks as well as suggested future work.

Chapter 2

Modeling

2.1 Introduction

In order to control a system effectively, a good understanding of the dynamics and its effects on the environment must be modeled. The more that is known about the system, the more affective the controller can be. Controllers developed about linearized models are often used but limit the vehicles ability to the neighborhood encompassing this linear region. One of the most challenging parts in designing a control system for most vehicles is the complexities and interrelated dynamics present, thus a lot of time and effort is contributed to the modeling process.

For fixed-wing aircraft, the dynamics associated with pre-stall configurations are well known and have been studied since early flight. However, performing agile aggressive flight requires an aircraft go beyond the pre-stall configuration and into the poorly understood post-stall flight regime. In the following chapter, a detailed description of the tests that were performed in order to obtain these equations of motion (EOM) is given. Multiple tests are compared to theoretical findings, in particular predictions made using flat plate theory which describes the entire flight regime particularly well at low Reynolds numbers ($< 10^4$). Finally, the EOM are compared to actual flight test data recorded in the Real-time indoor Autonomous Vehicle test ENvironment (RAVEN) concluding that the system identification validates the models accuracy.

2.2 Preliminaries

In the following sections the experimental aircraft as well as the notation and nomenclature are introduced. This aircraft is used throughout the rest of the controller implementation in this thesis. Due to the complexities of modeling, standard aerodynamic notations presented in [16] and [17] are used.

2.2.1 Nomenclature

$\mathbf{p}$	=	Vehicle position vector, m
$\mathbf{v}$	=	Velocity vector, m/s
$\mathbf{w}$	=	Angular rate vector, radians/sec
$\mathbf{q}$	=	4-Dimensional unit quaternion
R_B^I	=	Rotation matrix from vehicle body to inertial frame
$\mathbf{u}_w$	=	Control surface deflection input vector, radians
m	=	Mass of the aircraft, kg
α	=	Angle-of-attack of the wing, radians
ρ	=	Density of air, kg/m^3
δ_t	=	Thrust, N
$(\cdot)_d$	=	Desired value
$(\cdot)_e$	=	Error value
$(\cdot)_a$	=	Measured value
$(\cdot)_p$	=	Details pertaining to the propeller
$(\cdot)_{\mathrm{w}}$	=	Area affected by propeller downwash
$(\cdot)_{\mathrm{nw}}$	=	Area not affected by propeller downwash
$(\cdot)_{\mathrm{ref}}$	=	Reference
$(\cdot)^I$	=	Inertial frame
$(\cdot)^B$	=	Body frame

2.2.2 Vehicle Description

To facilitate the implementation of various controllers, a slightly modified version of the high performance Clik [18] indoor aerobatic plane designed by RC Factory is being used, shown in Figure 2-1(a). This aircraft is extremely maneuverable due to its large control surfaces and high thrust-to-weight ratio. The vehicle is equipped with an Axi Brushless Out-Runner 2203/52 motor with a 20cm Grapner Slowfly Propeller which provides a thrust-to-weight ratio in excess of 1.4. Control deflection actuation is provided by three GWS pico standard servo motors and receives commands on a GWS four-channel micro receiver. The aircraft is also equipped with a 400mAh 2-cell lithium polymer battery which delivers power to an 8-amp JETI electronic speed controller shown in Figure 2-1(b).

(a) Indoor aerobatic Clik aircraft

(b) Vehicle hardware components

Figure 2-1: Vehicle and hardware used for controller implementation

The vehicle is extremely light for its size, weighing approximately 170 grams due to the use of 2.8mm thick Dapron foam material for its body and carbon fiber strips to reinforce structurally weak areas. The aircraft is symmetric about body x-z axes (see Figure 2-10) and made up of flat plates. The total length of the vehicle is 90cm with a wingspan of 84cm and the inertial as well as the surface area estimation is provided through the use of SolidWorks CAD modeling software. More vehicle parameters and details are given in Table 2.1.

Table 2.1: Clik aircraft parameters

Parameter	Description	Value	Units
AR	Aspect Ratio	4.2	
d	Propeller diameter	20.0	cm
A_{ap}	Area of aileron induced by propeller downwash	0.00150	m^2
A_e	Area of the elevator	0.03226	m^2
A_r	Area of the rudder	0.03123	m^2
$A_{a_{nw}}$	Area of the aileron in the free-stream	0.024	m^2
I_{xx}	x-Moment of Inertia	0.00143	kg· m^2
I_{yy}	y-Moment of Inertia	0.00610	kg· m^2
I_{zz}	z-Moment of Inertia	0.00737	kg· m^2
L_{ap}	Moment arm of the aileron affected by prop-wash	0.080	m
L_{ep}	Moment arm of the elevator (cg to center of pressure)	0.533	m
L_{rp}	Moment arm of the rudder (cg to center of pressure)	0.631	m
$L_{a_{nw}}$	Moment arm of the aileron free-stream induced	0.23	m

2.2.3 RAVEN Testbed

Vehicle position and attitude sensing is done off-board through the Real-time indoor Autonomous Vehicle test ENvironment (RAVEN), eliminating the need for onboard sensors which typically are expensive and add unwanted weight. RAVEN provides a well equipped, robust platform for the rapid prototyping of controllers applicable to many different vehicles. This testing environment uses a position and orientation tracking system with an update rate of 120 Hz, minimal delay (20-30msec) and sub-millimeter accuracy with the use of Vicon motion capture camera system [19]. A single camera can be seen at the top of Figure 2-1(a) as the black object with the red ring. The only requirement is that the vehicle be equipped with reflective dots which the cameras use for object recognition.

Since only position and attitude data are directly measured by the system, the states time rate of change must be taken to acquire rate data. This is done by the process of a Kalman filter to attenuate noise produced by differentiating. The filter requires that smooth continuous data is used as the input. Quaternion data however,

is not a smooth continuous signal by the process of extraction, so a special algorithm is implemented to ensure smoothness which is outlined in Appendix A. From this tracking system, state data such as position, velocity, attitude and rotational rate is computed and used for full-state feedback. The state data is then routed to a computer which processes the desired control commands. These control commands are then sent to the R/C transmitter which relays the respective commands to the vehicle, closing the control loop.

2.3 Tests Performed

2.3.1 Introduction

To accurately identify the dynamic model, a JR3 6-axis load cell is placed at the center of gravity which allowed steady-state force and moment data to be taken for all 3 axes (load cell configuration shown in Figure 2-2). With the aid of a low pass filter

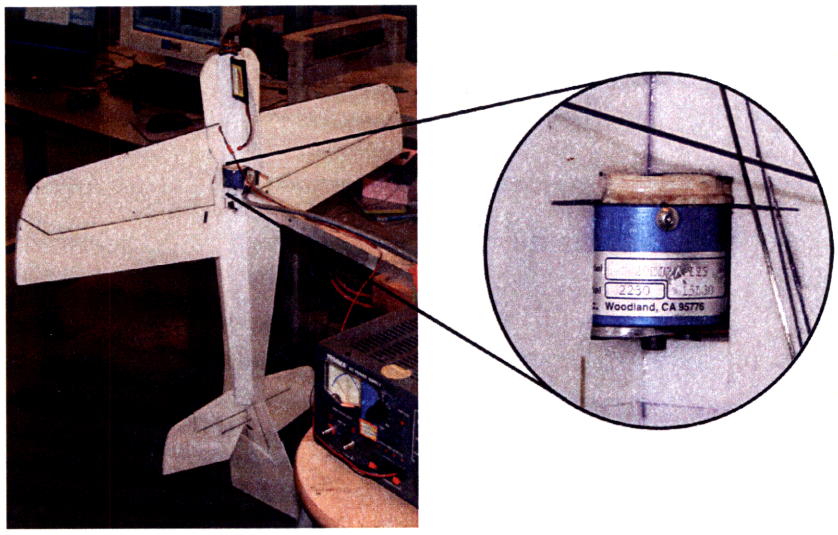


Figure 2-2: Modeling test setup and load cell placement

to attenuate high frequency noise, the multi-axis load cell is able to measure forces within 10^{-3} Newtons and moments within 10^{-2} Newton meters. Two main types of tests were performed, prop-hang and wind tunnel tests. A prop-hang is when the force used to balance out the vehicles weight is solely provided by the propeller thrust.

These tests were performed with the intent of determining the specifics of the vehicles dynamics at high angles-of-attack during a hover to level-flight transition.

2.3.2 Prop-Hang Test

The prop-hang was the first test done and was primarily used to determine the specifics of individual axis moment data and propeller downwash velocity. The propeller downwash velocity is defined as the induced velocity created by the spinning blade and is found by using aspects of propeller momentum theory and conservation of mass as in [7] and [20] and based on inviscid, incompressible flow assumptions. A similar approach is used here but modified experimentally under the assumption that the flow created is in a uniform conical form as a function of the thrust command and distance from the propeller blades instead of a stream tube that extends infinitely and uniformly far downstream. This is an approximation for attempting to capture the loss of efficiency due to the slipstream rotation of the fluid within and outwards from the stream tube, which is one of the major objectives propeller momentum theory. The approximation assumes that the conical formation, in the limit, will approach a pure cylindrical shape as the thrust is increased. In order to measure the radius of the assumed cone shape as a function of distance and thrust, pieces of tinsel were pieced along both the aileron and rudder/elevator control surfaces as shown in Figure 2-3. The thrust is varied to determine the granularity spacing of the tinsel which is necessary to produce an accurate measurement. Figure 2-4 shows a linear least squares fit to the measured radius as a function of thrust. With the cross-sectional area estimated, the flow created is approximated by [20]

$$u_p(\delta_t, l) = \sqrt{\frac{u_\infty^2}{4} + \frac{\delta_t}{2\rho A_{\text{disk}}(\delta_t, l)}} - \frac{u_\infty}{2} \tag{2.1}$$

where u_∞ is the magnitude of the free-stream velocity given as

$$u_\infty = \|\mathbf{v}\|_2. \tag{2.2}$$

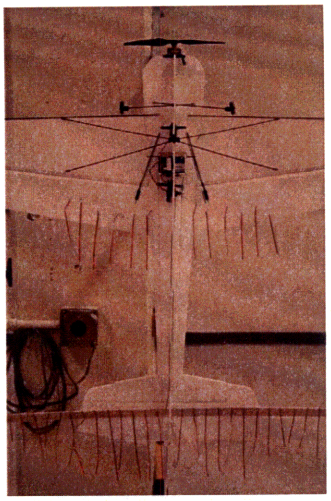


Figure 2-3: Aircraft mounted with tinsel hung around control surfaces to determine propeller downwash diameter

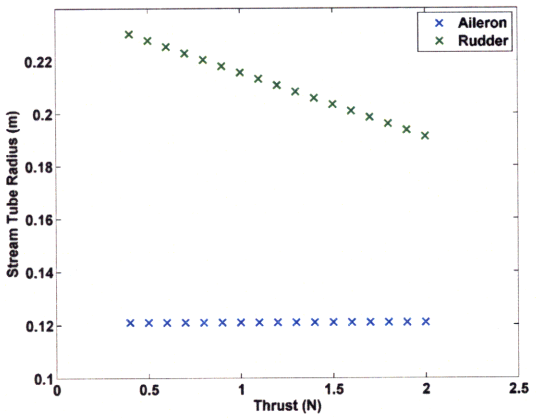


Figure 2-4: Linear least square fit of the radius of the cone produced by propeller downwash

Assuming that during a prop-hang their is no free-stream velocity present, Equation 2.1 reduces to

$$u_p(\delta_t, l) = \sqrt{\frac{\delta_t}{2\rho A_{\text{disk}}(\delta_t, l)}} \tag{2.3}$$

where $A_{\text{disk}}(\delta_t, l)$ is determined experimentally and represents the cross-section of the conical region as a function of thrust (δ_t) and distance (l) from the propeller. The propeller downwash velocity is presented as u_{p_a} or u_{p_r}, depending on whether the aileron (a) or rudder/elevator (r) is the particular aerodynamic region of interest. The rudder and elevator calculations are done together due to very similar measurements and distance from the propeller. Actual measurements were taken using an anemometer and comparisons are shown in Figure 2-5.

The prop-hang test provides means to estimate the moment coefficients for the control surfaces by using the combination of thin airfoil theory and Prandtl's lift-line theory [17] by

$$M = \frac{1}{2}\rho u_p^2 C_{l_{e/r}} A_{e/r} L_{e/r}. \tag{2.4}$$

The coefficient of lift is the last parameter to be estimated. Prandtl's lift-line the-

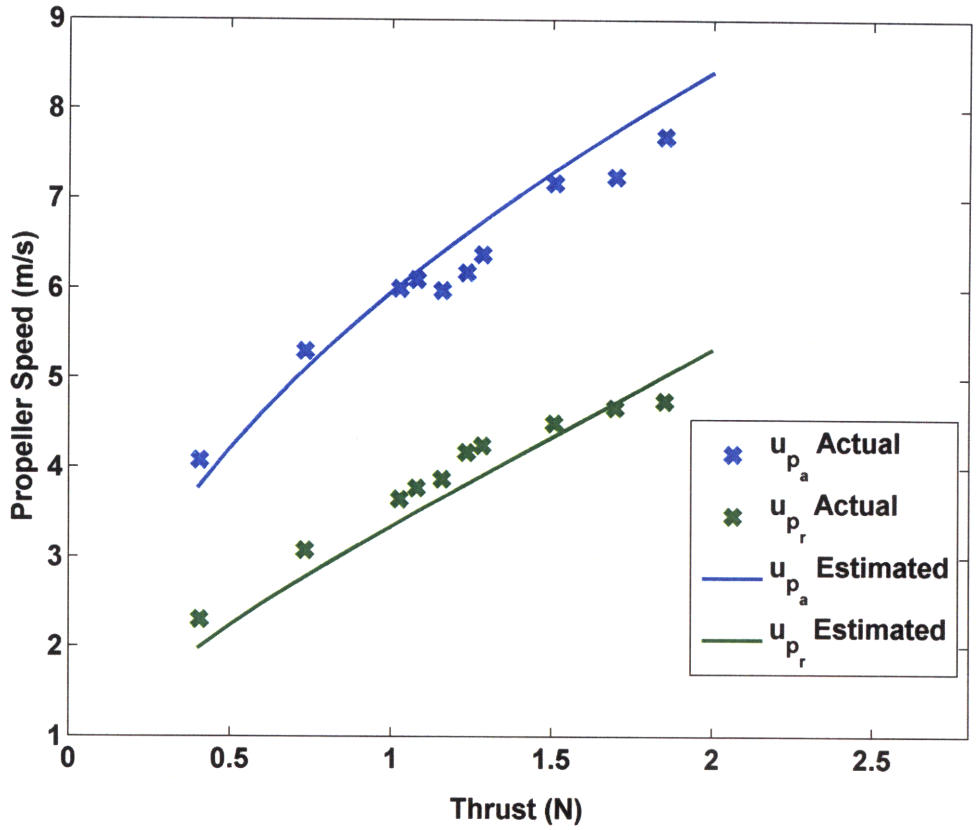
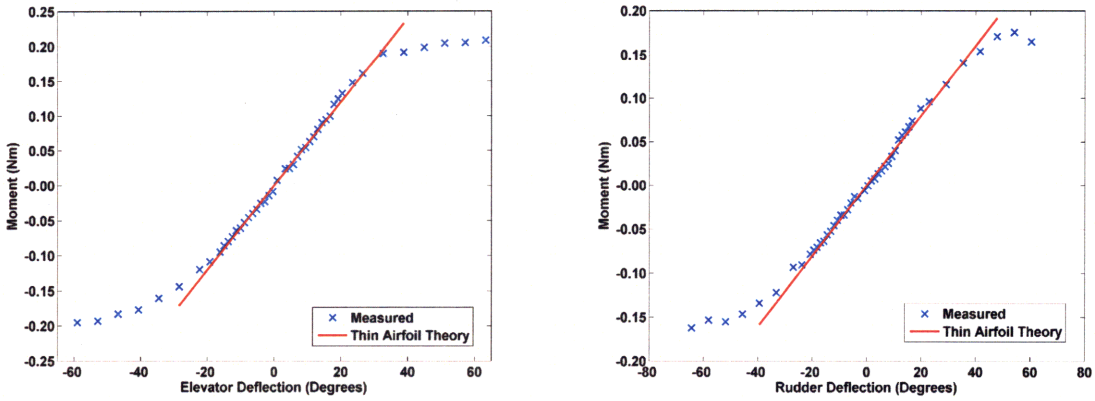


Figure 2-5: Estimated and actual propeller downwash flow located at the respective control surfaces

ory predicts that for an infinite flat plate, C_l is approximated by a value of 2π. A correction accounting for a finite aspect ratio for each surface must be taken into account [21] by

$$C_{L_\alpha} = \frac{C_{l_\alpha}}{1 + C_{l_\alpha}/\pi AR}. \tag{2.5}$$

With the load cell placed at the center of gravity of the vehicle and thrust approximately equal to the weight, measurements were taken to determine the moment created with each control surface deflection. Figure 2-6 shows a comparison between the theoretical and measured moment data about an individual aircraft body axis. Aileron control authority is limited during a prop-hang due to the lack of downwash over these control surfaces. Measurements taken were primarily in the noise of this particular instrument and could not be physically determined. An estimate



(a) Elevator moment coefficient

(b) Rudder moment coefficient

Figure 2-6: Moment coefficient as a function of respective control surface deflection angle

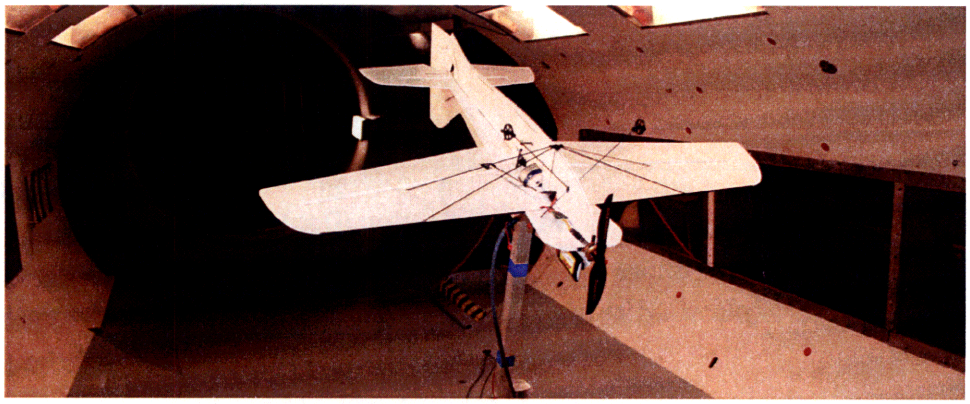


Figure 2-7: Wind tunnel test setup - mounted upside down due to maneuverability constraints

of the propeller drag and moment contribution of the ailerons are discussed more in Section 2.4.

2.3.3 Quasi-Steady State Wind Tunnel Test

The vehicle was tested at multiple free-stream conditions in the wind tunnel (see Figure 2-7). The first set of tests were done at 5 degree increments of angle-of-attack without the propeller on to obtain nominal aerodynamic coefficients. The primary objective was to gain insight on the moments produced by the body/elevator deflection combination. Since elevator coefficient data has been determined in Section 2.3.2, the

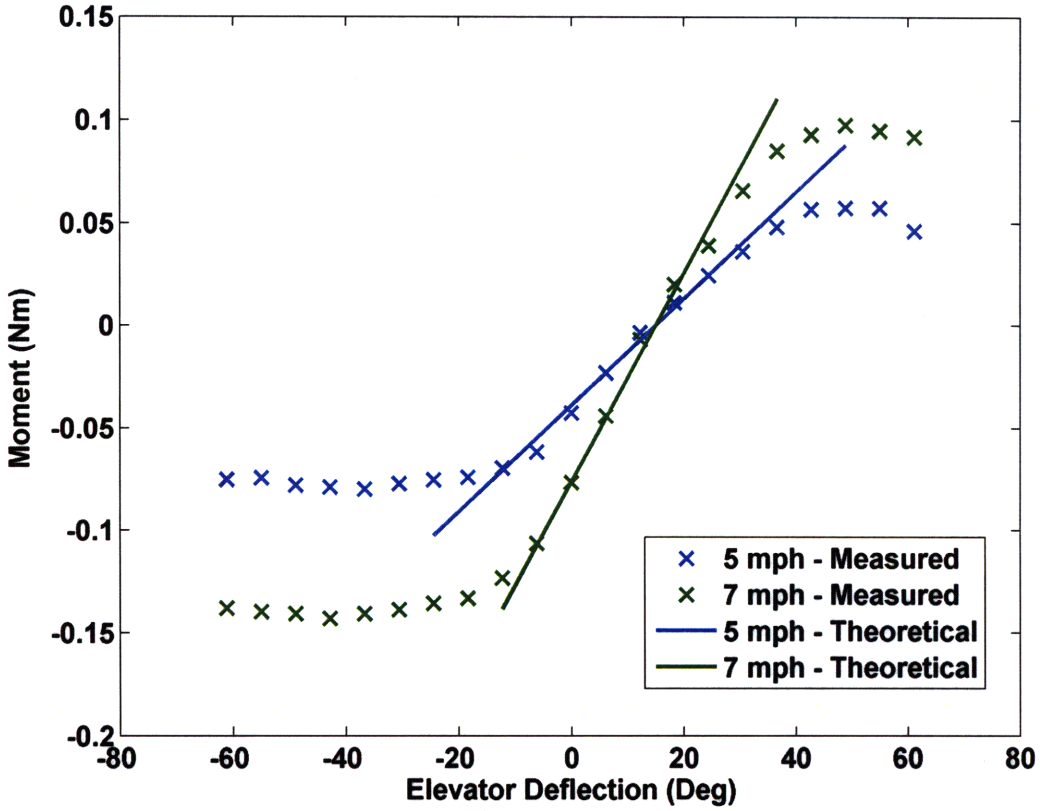


Figure 2-8: Sample wind tunnel moment data taken at 30° angle-of-attack

aircraft moment due to angle-of-attack (C_{m_α}) is found by

$$C_{m_\alpha} = \frac{C_m - C_{m_{\delta_e}} \delta_e}{\alpha} \tag{2.6}$$

Examples of the data found can be seen in Figure 2-8.

Next, tests were performed while incrementing the elevator and throttle commands. It was determined that the vehicle aerodynamic dynamics over the wing be split into two parts, the area in which the propeller downwash affects the aircraft, denoted by $(\cdot)_{\mathrm{w}}$, and the free-stream-only non-affected areas, $(\cdot)_{\mathrm{nw}}$. The area affected by the propeller downwash also experiences the free-stream velocity shown in Figure 2-9. The resulting airspeed and angle-of-attack are given by

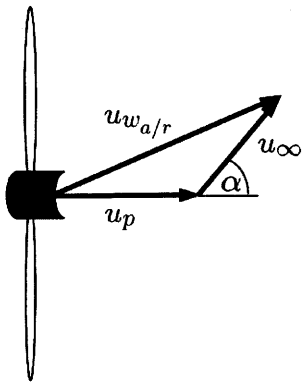


Figure 2-9: Effect of propeller downwash combined with the free-stream velocity

$$u_{w_{a/r}} = \sqrt{u_\infty^2 + u_{p_{a/r}}^2 + 2u_\infty u_{p_{a/r}} \cos(\alpha)} \tag{2.7}$$

$$\alpha_{w_{a/r}} = \arctan \frac{v_z}{v_x + u_{p_{a/r}}}. \tag{2.8}$$

The rest of the wind tunnel aerodynamic data is presented in the next section.

2.4 Equations of Motion

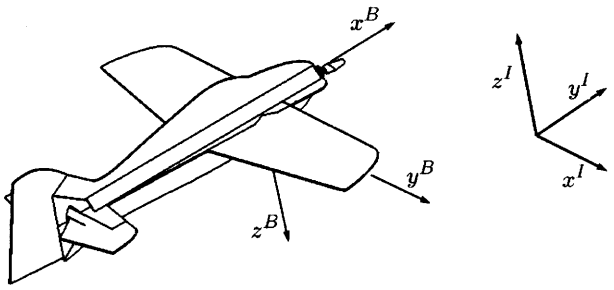


Figure 2-10: Aircraft body and inertial coordinate frames

The set of nonlinear differential equations follows the baseline model described in [16] using Newton's second law for rigid-body dynamics but modified for this particular aircraft's kinematics and dynamics. Equations are in the aircraft body

frame (shown in Figure 2-10) given by

$$\dot{\mathbf{p}}^I = R_B^I \mathbf{v}^B, \tag{2.9}$$

$$\dot{R}_B^I = R_B^I S(\mathbf{w}^B), \tag{2.10}$$

$$J\dot{\mathbf{w}}^B = -S(\mathbf{w}^B)J\mathbf{w}^B + \mathbf{f}_w + G_w \mathbf{u}_w, \tag{2.11}$$

$$M\dot{\mathbf{v}}^B = -S(\mathbf{w}^B)M\mathbf{v}^B + \mathbf{f}_v + \mathbf{g}_v \delta_t. \tag{2.12}$$

where the mass, inertia and thrust force directional matrices are respectively,

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix}, \qquad J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \qquad \mathbf{g}_v = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Due to the symmetrical build of the aircraft, the cross-coupled inertia tensor terms I_{xy}, I_{xz} and I_{yz} are considerably smaller than the coupled terms and are disregarded. The angular velocity cross-product matrix, moment decoupling matrix and control surface deflections are

$$S(w) = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}, \qquad G_w = \begin{bmatrix} g_{11} & 0 & 0 \\ 0 & g_{22} & 0 \\ 0 & 0 & g_{33} \end{bmatrix}, \qquad \mathbf{u}_w = \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix}$$

where

$$\begin{bmatrix} g_{11} \\ g_{22} \\ g_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\rho(u_{\mathrm{w_a}}^2 C_{L_{\delta_{aw}}} A_{ap} L_{ap} + u_\infty^2 C_{L_{\delta_{anw}}} A_{a_{\mathrm{nw}}} L_{a_{\mathrm{nw}}}) \\ \frac{1}{2}\rho u_{\mathrm{w_r}}^2 C_{L_{\delta_e}} A_e L_{ep} \\ \frac{1}{2}\rho u_{\mathrm{w_r}}^2 C_{L_{\delta_r}} A_r L_{rp} \end{bmatrix}. \tag{2.13}$$

The constant parameters $(A_{a_{\mathrm{nw}}}, A_{ap}, A_e, A_r)$ and $(L_{a_{\mathrm{nw}}}, L_{ap}, L_{ep}, L_{rp})$ are the respective aileron, elevator and rudder areas and moment arms given in Table 2.1. In Equation 2.13, the difference in the form of g_{11} is due to the fact that only part of the wing area is affected by the propeller downwash while the rudder and elevator control surfaces are always engulfed. Although the propeller downwash conical area

changes as a function of thrust and distance, it deviates very little over the wings and is modeled as a constant (see Figure 2-4). Development of the remaining force and moment terms will be based on this assumption. This assumption is a major contributor to the total force and moments created because of the low Reynolds number in which the aircraft is flying through, $\sim 10^4$.

The force vector $\mathbf{f}_v$ is the sum of the gravitational and aerodynamic forces in the body frame, given in Equation 2.14.

$$\mathbf{f}_v = R_I^B \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} - \begin{bmatrix} -\cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \begin{bmatrix} \text{Drag} \\ 0 \\ \text{Lift} \end{bmatrix} - \begin{bmatrix} C_{D_x} v_x \\ C_{D_y} v_y \\ C_{D_z} v_z \end{bmatrix}. \quad (2.14)$$

The first term involves the transformation of weight from the inertial to body frame. The second term comprises the aerodynamic contribution of lift and drag forces. Since these are in the wind frame a rotation matrix pre-multiplies the aerodynamic terms to obtain the desired forces in the body frame. The last term in Equation 2.14 represents the viscous drag that is induced by translating through the air. Drag and lift forces are divided into two sections and are described as

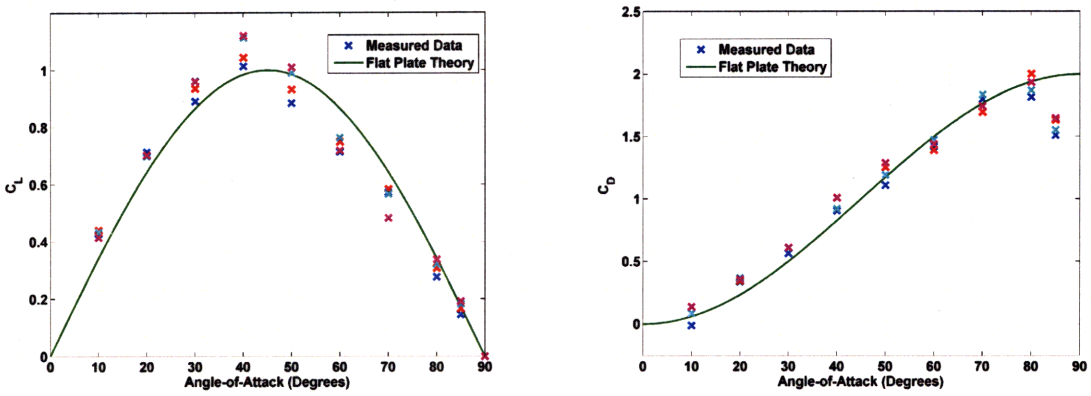
$$\text{Drag} = \frac{1}{2}\rho \left(u_{\text{w}_\text{a}}^2 C_{D_\text{w}} S_\text{w} + u_\infty^2 C_D S_\text{nw} \right) \quad (2.15)$$

$$\text{Lift} = \frac{1}{2}\rho \left(u_{\text{w}_\text{a}}^2 C_{L_\text{w}} S_\text{w} + u_\infty^2 C_L S_\text{nw} \right). \quad (2.16)$$

Figure 2-11 shows the coefficients of lift (C_L) and drag (C_D) for the free-stream section. Since this aircrafts main lifting surface is without camber, the coefficient of lift is symmetric about the body x-y axes. Note that the measured data is in agreement with flat plate theory [22], where

$$C_L = 2\sin(\alpha)\cos(\alpha) \qquad C_D = 2\sin^2(\alpha). \quad (2.17)$$

The areas affected by the prop-wash experience an angle-of-attack ≤ 20 degrees (determined by measuring flows then using Equation 2.8) due to the contribution of



(a) Coefficient of lift

(b) Coefficient of drag

Figure 2-11: Measured versus theoretical force coefficients for various free-stream velocities

the additive propeller flow, and thus Prandtl's classical lifting-line theory [17] can be used:

$$C_{L_{\mathrm{w}}} = C_{L_{\alpha_{\mathrm{w}}}} \alpha_{\mathrm{w_a}} \qquad C_{D_{\mathrm{w}}} = C_{D_{o_{\mathrm{w}}}} + \frac{C_{L_{\mathrm{w}}}^2}{\pi e \mathcal{AR}} \tag{2.18}$$

These also need to be corrected for a finite wing by Equation 2.5. Descriptions of the various parameters are given in Table 2.2. The term $\mathbf{f}_w$ in Eq. 2.11 represents the

Table 2.2: Clik aerodynamic parameters

Parameter	Description
e	Oswald efficiency factor
$C_{L_{\delta_a}}$	Aileron coefficient of lift
$C_{L_{\delta_e}}$	Elevator coefficient of lift
$C_{L_{\delta_r}}$	Rudder coefficient of lift
$C_{D_{\alpha_{\mathrm{w_a}}}}$	Effective drag coefficient
$C_{L_{\alpha_{\mathrm{w_a}}}}$	Effective lift coefficient
$C_{m_{\alpha_{\mathrm{w}}}}$	Moment coefficient from effected propeller downwash
$C_{m_{\alpha_{\mathrm{nw}}}}$	Moment coefficient from free-stream velocity
c	Moment arm from effected propeller downwash
l_{nw}	Moment arm from free-stream velocity
S_{w}	Wing area effected by propeller downwash
S_{nw}	Wing area not effected by propeller downwash

rest of the net torque acting about the aircraft center of gravity (cg),

$$\mathbf{f}_w = \begin{bmatrix} -M_{\text{acc}} - M_{\text{drag}} + \frac{1}{2}\rho L_a L_{a,p} A_a u_{p_a} w_x \\ \frac{1}{2}\rho L_e L_{e,p} A_e u_{p_r} w_y - \frac{1}{2}\rho \left(u_{\text{w}_\text{r}}^2 C_{m_{\alpha_\text{w}}} \alpha_{\text{w}_\text{r}} S_\text{w} c + u_\infty^2 C_{m_{\alpha_\text{nw}}} \alpha S_\text{nw} l_\text{nw} \right) \\ \frac{1}{2}\rho L_r L_{r,p} A_r u_{p_r} w_z \end{bmatrix}. \quad (2.19)$$

M_{acc} and M_{drag} are moment contributions from the acceleration and drag of the propeller, respectively, calculated as [7]

$$M_{\text{acc}} = I_p \dot{\omega}_p \quad (2.20)$$

$$M_{\text{drag}} = \frac{\delta_t d C_p}{2\pi C_T}. \quad (2.21)$$

Equation 2.20 is a function of both the inertia of the propeller about the spinning axis and the rotational acceleration denoted as I_p and $\dot{\omega}_p$ respectively. Due to the relative size of the propeller compared to the vehicles x inertial body tensor, this term is negligible and is not used in the model formulation. Equation 2.21 however, is not negligible and its contribution can be seen during hover when the ailerons deflect in order to counteract its moment, which can be seen Section 3.5.2. The thrust and power coefficients C_T and C_P are estimated for the given propeller using a NACA-standardized table as in [7].

2.4.1 System Identification

The following results compare the measured and simulated states for two types of maneuvers. These maneuvers include sinusoidal inertial y-velocity commands in hover causing the vehicle to oscillate about the body y-axis and the full transition to level-flight and back to hover which uses the controller outlined in Chapter 3. The simulation is given the initial conditions of each state and the input to each of the control surfaces for processing. In order to provide manual control of the vehicle, an external joystick is programmed that commands desired velocity in both the x and y-inertial frame. This allows the user to define a suitable starting position within the

constrained environment. A trigger switch, when executed, commands the desired autonomous maneuver. For each executed maneuver, the body velocities and rotational rate are of interest and are compared in the following plots. It is important to note that the modeling is done primarily to support the transition maneuver on the body x and z forces and the y-axis moment.

Sinusoidal Velocity Inputs

For this test, a sinusoidal input to the y-inertial velocity is commanded and used to determine the accuracy in hover and high angles-of-attack with slight transition to level-flight mode properties. The sinusoidal input is commanded as

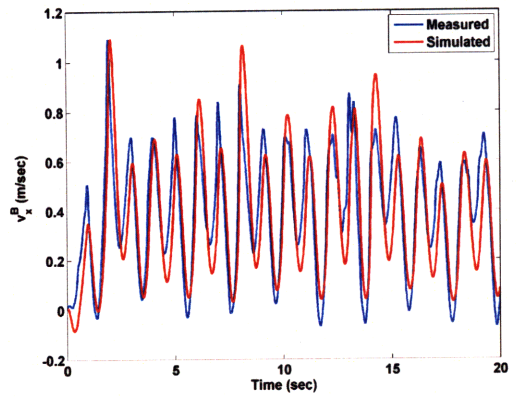
$$v_{y_d}^I = 2\sin(\pi t).$$

This is an important test that is used to determine the accuracy of the model that was accomplished through the prop-hang test. It verifies how well the propeller downwash velocity is modeled as well as moments created by control surface actuation. Since the aircraft is primarily in a hover position, the body x-axis is mainly testing the modeling of the thrust force created by the propeller shown in Figure 2-12(a). Due to inaccuracies in the power supply and un-modeled motor lag dynamics, slight deviations are present. Most of the sinusoidal command can be seen in Figure 2-12(e). For this inertial velocity command, the body z-velocity will mainly experience drag at a very high angle-of-attack. Most of the moment is generated about the y-body axis which is evident in Figure 2-12(d). Deviations occur after peak inputs which are distinctly due to the quasi-steady state modeling.

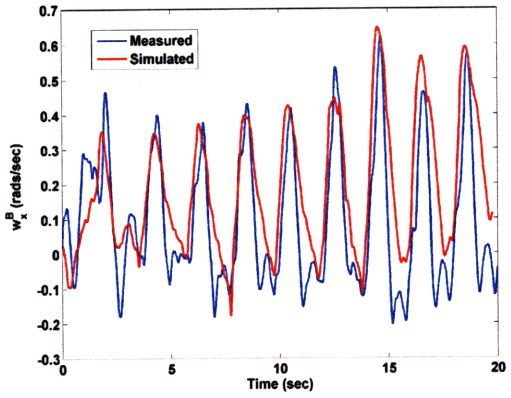
Transition to Hover and Back

The main test is to compare the output during a transition from hover to level-flight and back to hover. To obtain this desired maneuver, an exponential decaying y-velocity,

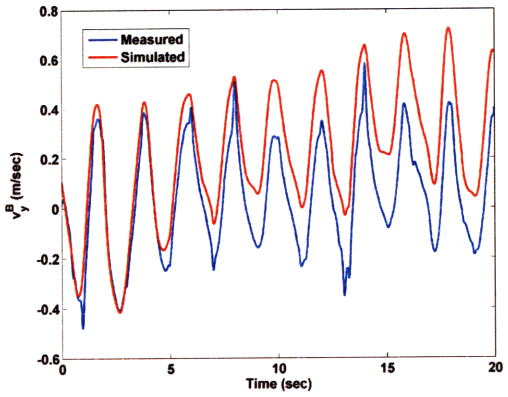
$$v_{y_d}^I = -5.0(1 - e^{-3.0t})$$



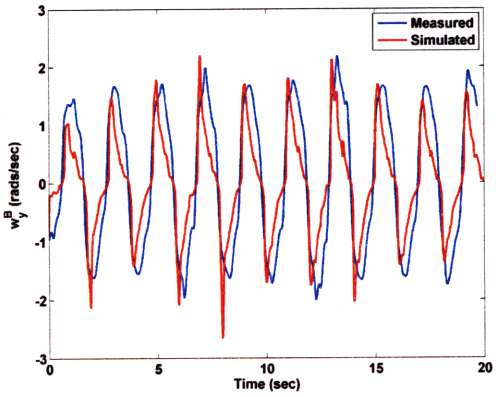
(a) Body x velocity which for this test is mostly thrust



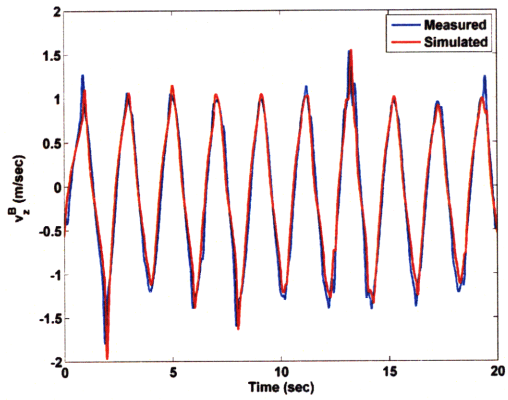
(b) Roll rotational rate compensating for propeller drag



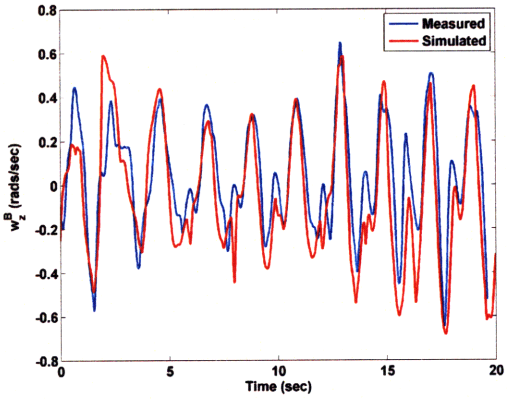
(c) Body y velocity



(d) Pitch rotational rate



(e) Body z velocity which mainly experiences drag



(f) Yaw rotational rate

Figure 2-12: System Identification: compared state outputs for the sinusoidal velocity test

is commanded due to the spatial limitation of a horizontal distance of 9.5 meters. This command allowed the vehicle to transition from hover to steady-level flight as can be seen in Figure 2-13.

A comparison of the measured to simulated state data is provided in Figure 2-14. This data shows two takes of this maneuver which is apparent by the two large spikes in Figure 2-14(a). Larger deviations are noticeable and are due to the very quick control surface actuation necessary to perform the transitions. The largest deviation in force is shown in Figure 2-14(e) at the point where the aircraft transitions from level to hover flight regimes. At this point the aircraft is essentially performing a skid stop, moving considerably quick at this high angle-of-attack. Flow separation makes the drag calculation a bit more obscure and the model over predicts these forces. However, trends in the data are similar and show that even with some un-modeled dynamics, the model can predict a response sufficiently well.

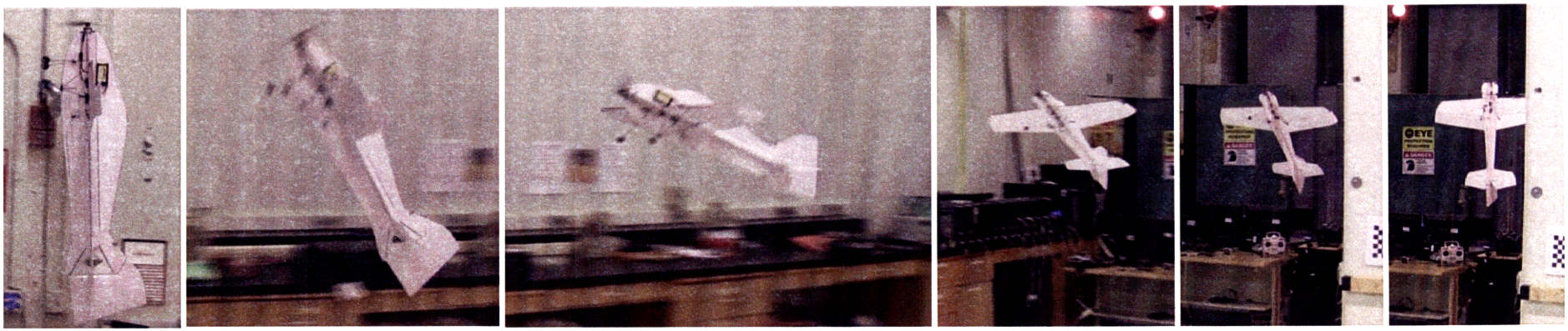
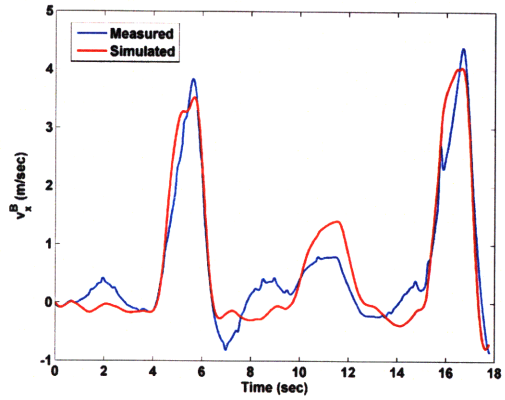
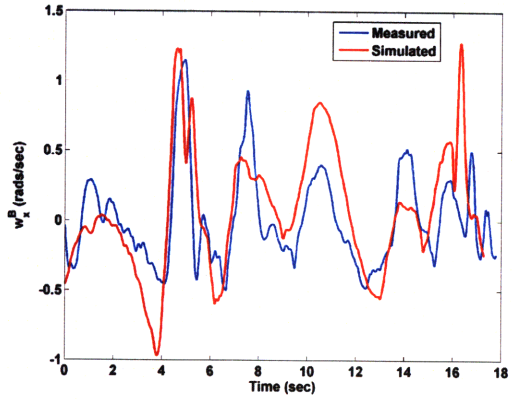


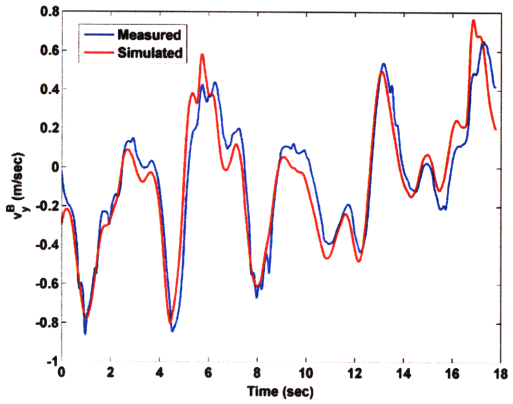
Figure 2-13: Stills from the Clik transition maneuver



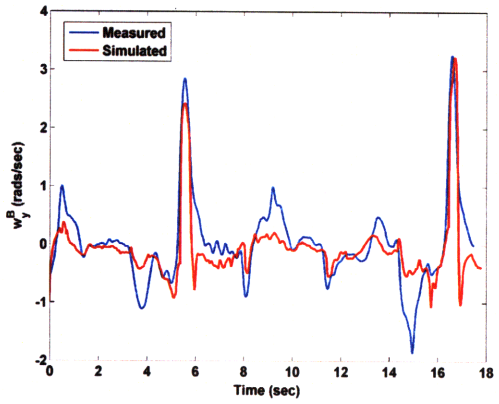
(a) Body x velocity which experiences both drag and lifting forces



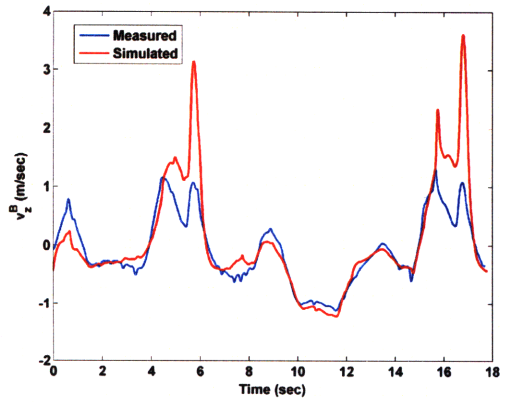
(b) Roll rotational rate



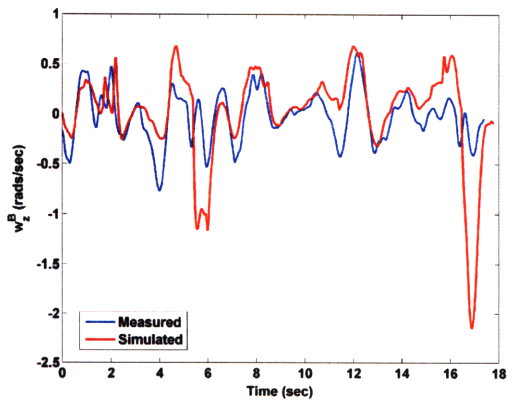
(c) Body y velocity



(d) Pitch rotational rate contributing all the moment needed to perform the transition



(e) Body z velocity which experiences both drag and lifting forces



(f) Yaw rotational rate

Figure 2-14: System Identification: compared state outputs for the hover to transition test

Chapter 3

Quaternion Based Control

3.1 Introduction

One of the most recognizable issues when designing a controller to perform aggressive flight maneuvers is the concern over which attitude descriptor to use. The standard aerodynamic Euler angles suffer from singularity problems due to gimbal lock, a point in which a degree of freedom is lost. Therefore, other descriptors such as the quaternion and direction cosine matrix (DCM) are used, each with its own distinct advantage. In Chapter 2, the aircraft model was pieced together using a DCM, an orthogonal matrix whose inverse (and consequently the transpose due to the properties of orthogonal matrices) represents the reverse transformation. One of the caveats of using this descriptor is the amount of computation that must be done in order to complete a single transformation, 9 multiplications and 6 summations per transformation. Quaternion descriptors are less computationally intensive and, in this chapter, the use of a quaternion based controller is presented. Figure 3-1 shows the controller's inner and outer loop architecture. Section 3.2 explains how the inner loop stabilizes the attitude of the aircraft based on a nominal desired quaternion and rotational rate by utilizing many of the properties of quaternion mathematics. The inner controller also regulates the amount of thrust necessary to perform a maneuver based on velocity errors while attempting to maintain its vertical position. The velocity controller provides the inner loop with an updated desired quaternion based on the error be-

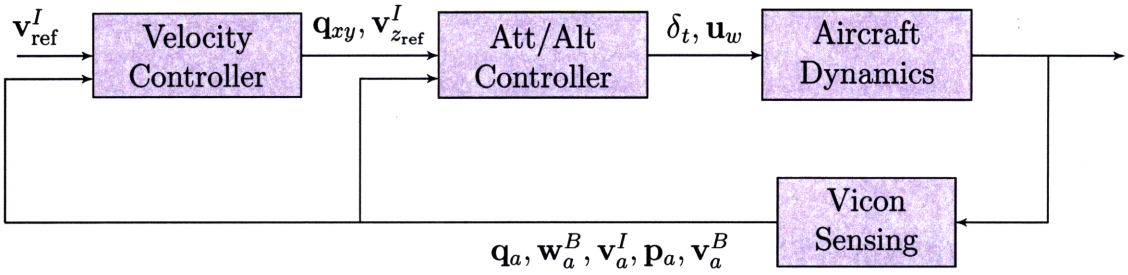


Figure 3-1: Quaternion-based control system architecture

tween commanded and measured velocity, and also an update on vertical position loss. Again, the RAVEN testbed is used to provide all the state data necessary to close the loop.

3.1.1 Notation

Quaternions are defined using a four-element vector, $q = (q_o, q_x, q_y, q_z) = (q_o, \mathbf{q})$, representing a rotation in $\mathbb{R}^3$ space. The basic algebraic form of a quaternion is:

$$q = q_o + q_x\hat{i} + q_y\hat{j} + q_z\hat{k}. \tag{3.1}$$

These four elements have the unit magnitude property in the usual 3-dimensional vector space. The symbol $\otimes$ implies a quaternion multiplication while q^* is the quaternion conjugate defined as

$$q^* = q_o - q_x\hat{i} - q_y\hat{j} - q_z\hat{k}. \tag{3.2}$$

The subscripts in this chapter follow the same nomenclature presented at the beginning of Chapter 2.

3.2 Inner Attitude Loop

This inner attitude loop is a PD controller based on a desired attitude quaternion error and body rates [15]. The controller is developed to maintain a nominal prop-hang orientation. For the sake of avoiding confusion of multiple frame transformations, a

hover orientation is define as

$$q_d^{\text{ref}} = \begin{bmatrix} 1.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}. \tag{3.3}$$

In this orientation, the body z and x-axes are aligned with the inertial y and z-axes respectively. RAVEN provides the measured vehicle quaternion orientation data, and the error deviated from the reference quaternion is calculated using quaternion multiplication as

$$q_e = q_d^{\text{ref}} \otimes q_a^* \tag{3.4}$$

where $(\cdot)^*$ represents the quaternion conjugate. The individual rotational error about the reference quaternion is found by calculating the axis angle interpretation, defined by:

$$[\textbf{axis}, \text{angle}] = \left(\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}, \gamma_{\text{rotation}} \right).$$

To find the rotational error for an individual axis, the total rotation error must first be defined by

$$\gamma_{\text{rotation}} = 2\cos(q_{e_o})$$

and the axis vector error as

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \frac{1}{\sin(\gamma_{\text{rotation}}/2)} \begin{bmatrix} q_{e_x} \\ q_{e_y} \\ q_{e_z} \end{bmatrix}.$$

The axis angle vector is a unit vector by definition and multiplying each component by the total error rotation yields the individual axis errors given by

$$\begin{bmatrix} \phi_e \\ \theta_e \\ \psi_e \end{bmatrix} = \gamma_{\text{rotation}} \begin{bmatrix} a_z \\ a_x \\ a_y \end{bmatrix}.$$

Each one of these axis angular errors are defined from the desired attitude. It is important to note that in this orientation, the commonly viewed roll error, labeled ϕ_e, is about the z-axis. This is consistent with having the body x and inertial z-axes aligned, and similar arguments are made for the other axis errors. The control command that maintain a hover orientation is defined as $\mathbf{u}_w$

$$\mathbf{u}_w = \begin{bmatrix} K_{p_{\delta_a}} & 0 & 0 \\ 0 & K_{p_{\delta_e}} & 0 \\ 0 & 0 & K_{p_{\delta_r}} \end{bmatrix} \begin{bmatrix} \phi_e \\ \theta_e \\ \psi_e \end{bmatrix} + \begin{bmatrix} K_{d_{\delta_a}} & 0 & 0 \\ 0 & K_{d_{\delta_e}} & 0 \\ 0 & 0 & K_{d_{\delta_r}} \end{bmatrix} \mathbf{w}^B \tag{3.5}$$

which is a PD controller on attitude.

3.3 Outer Velocity Loop

The outer velocity loop is a PI controller on the velocity error in the inertial frame. This control command manipulates the desired quaternion to produce an attitude in the direction of decreasing velocity error. One of the goals in performing this transition maneuver is to maintain a desired altitude, therefore the controller will limit the amount of control authority as a function of altitude loss.

In order to redefine a new attitude, a transformation that manipulates the desired quaternion based on the error of the inertial velocity command is developed. Since the objective of this controller is to translate the aircraft in the inertial x and y direction while maintaining altitude, errors between commanded and measured inertial velocities will be used to affect the transformation. Start by defining the inertial

velocity error as

$$\begin{bmatrix} v_{x_e} \\ v_{y_e} \\ v_{z_e} \end{bmatrix}^I = \begin{bmatrix} v_{x_d} - v_{x_a} \\ v_{y_d} - v_{y_a} \\ v_{z_d} - v_{z_a} \end{bmatrix}^I \tag{3.6}$$

and inertial z-error as

$$z_e^I = z_d^I - z_a^I. \tag{3.7}$$

These errors can be used to define the final quaternion transformation

$$q_{xy} = \begin{bmatrix} 1.0 \\ K_{p_{yvel}} v_{y_e} + K_{i_{yvel}} \int v_{y_e} \mathrm{d}t + K_{p_z} \mathrm{sign}(v_{y_a}) z_e^I + K_{v_z} \mathrm{sign}(v_{y_a}) v_{z_d}^I \\ K_{p_{xvel}} v_{x_e} + K_{i_{xvel}} \int v_{x_e} \mathrm{d}t + K_{p_z} \mathrm{sign}(v_{x_a}) z_e^I + K_{v_z} \mathrm{sign}(v_{x_a}) v_{z_d}^I \\ 0.0 \end{bmatrix}. \tag{3.8}$$

So from an intuitive perspective, to obtain a desired velocity in the inertial y-direction, a rotation about the inertial x (second element of q_{xy}) must be performed. The same reasoning is applied to the third element of q_{xy}. The vehicle will need to change its attitude, which is proportional to the error, but if a loss of altitude is sensed, the velocity controller will attenuate the attitude command based on the inertial z-velocity and position error which are the final two terms in Equation (3.8). Note that this is not a unit quaternion and needs to be normalized before performing quaternion multiplication. Quaternion multiplication is a transformation [23], so the new desired quaternion based on an inertial velocity command is

$$q_d = q_d^{ref} \otimes q_{xy}. \tag{3.9}$$

This new desired quaternion is what the inner loop will now act on, deflecting control surfaces in a manner that decreases the inertial velocity error and maintains altitude.

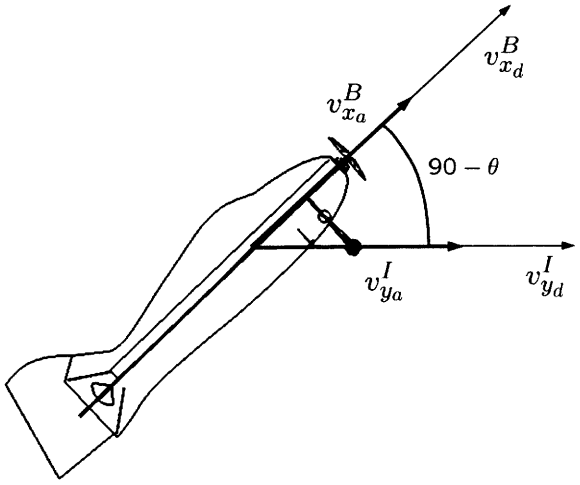


Figure 3-2: Vector description of additional body x-velocity necessary to obtain desired inertial velocity

3.4 Thrust Controller

The thrust controller is a PID controller with a feed-forward weight component, whose main functionality is to maintain a desired altitude. When performing the transition, modifications are used in an attempt to account for anticipated altitude loss and aerodynamic gain. One such modification is an adjustment of the feed-forward weight term as a function of θ, an angle measured from the vertical. The other is attempting to increase the body x velocity in order to decrease the inertial velocity errors. Figure 3-2 shows the velocity vector depiction of this modification, which is given by

$$\Delta v^B_x = \frac{v^I_{y_e}}{\sin\theta} \qquad \text{if} \qquad \theta > 30^o \tag{3.10}$$

Notice that as θ increases to 90^o, the error in the body velocity equals the error in the inertial frame. The complete control law is given as

$$\delta_t = \begin{cases} \frac{mg}{\cos\theta} + K_{p_{\delta_t}} z^I_e + K_{i_{\delta_t}} \int z^I_e \mathrm{d}t - K_{d_{\delta_t}} v^I_z & \text{if } \theta > 30^o \\ K_{p_{\delta_t}} z^I_e + K_{i_{\delta_t}} \int z^I_e \mathrm{d}t - K_{d_{\delta_t}} v^I_z + K_{v_x} \Delta v^B_x & \text{if } \theta < 30^o \end{cases} \tag{3.11}$$

In hover the controller is a regular PID controller where v_z^I provides the damping in the body frame. As the translation occurs, the modifications regulate the thrust, attempting to maintain the desired altitude and decrease the inertial velocity errors.

3.5 Results

3.5.1 Simulation

A simulation was developed to test the capabilities of the controller and to find the gains necessary to stabilize the system in a prop-hang orientation. To make the model more realistic, an ensemble of state data was taken to determine the mean and variance of the measurement noise. Saturators were added to the control actuators as well as time delays to limit their performance to a realistic range. Table 3.1 shows the gains that are used in the model as well as on the actual flight hardware.

Table 3.1: Simulation quaternion attitude loop gains

Gains	Aileron	Elevator	Rudder	Thrust
K_p	1.4	2.0	1.7	0.8
K_i	0.0	0.0	0.0	0.2
K_d	0.2	0.25	0.1	0.33

3.5.2 Decoupled Roll Control

Since the velocity commands are given in an inertial frame, the controller has an additive feature that will track the velocity commands decoupled from the aircraft roll orientation. For instance, if the aircraft is at a roll angle that does not correspond to a single control surface deflection (e.g. elevator) to obtain the desired velocity, the controller will couple the commands from the elevator and rudder.

To produce this roll decoupling feature, a transformation from the reference to the current roll angle quaternion must be calculated. A problem arises since the measurement of the current roll angle is not accurate, due to gimbal lock, and therefore

an intermediate derivation must be computed. This derivation involves the same computation as the inner loop controller but only the roll information is used.

To proceed, transform the measured quaternion into this new intermediate orientation (q_{int}) by defining

$$q_{\text{int}} = q_a \otimes q_{\text{ref}}. \tag{3.12}$$

The conversion from quaternion to roll Euler angle is found by

$$\phi = \arctan \frac{2(q_{o_{\text{int}}} q_{x_{\text{int}}} + q_{y_{\text{int}}} q_{z_{\text{int}}})}{1 - 2(q_{x_{\text{int}}}^2 + q_{y_{\text{int}}}^2)} \tag{3.13}$$

which is the roll angle defined from hover.

Since the desired quaternion (Eq. 3.3) is a transformation in itself (level-flight to hover), the roll transformation has to take place on the z-element of the quaternion, therefore defining the roll decoupling transformation as

$$q_{\text{roll}} = \begin{bmatrix} \cos\frac{\phi}{2} \\ 0.0 \\ 0.0 \\ \sin\frac{\phi}{2} \end{bmatrix} \tag{3.14}$$

which is a unit quaternion by definition. Now Equation (3.9) can be re-written as

$$q_d = q_d^{\text{ref}} \otimes q_{\text{roll}} \otimes q_{xy}. \tag{3.15}$$

This redefined desired quaternion is now independent of the roll angle of the aircraft. With this decoupling feature, a roll rate controller can be used to perform the rolling hover. This is a PI controller on the roll rate error defined as

$$\delta_a = K_{p_{\text{roll}}} w_{x_{\text{err}}} + K_{i_{\text{roll}}} \int w_{x_{\text{err}}} \mathrm{d}t \tag{3.16}$$

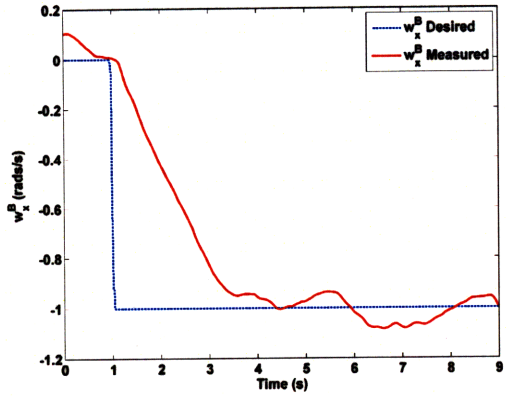
where

$$w_{x_{\text{err}}}^B = w_{x_d}^B - w_{x_a}^B. \tag{3.17}$$

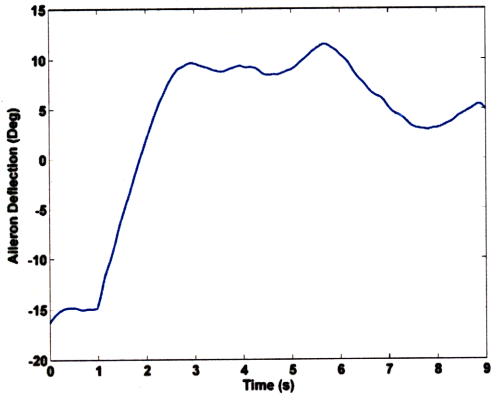
Performing a rolling hover requires that the body x-axis remains aligned with the inertial z-axis. Therefore, the aircraft is able to roll (or rotate) about its body x-axis in a controlled manner. Note that in hover, the effective airflow over the aileron control surfaces are limited to a smaller finite area, A_{ap}. During this maneuver, the thrust may be varying due to the added drag caused by the control surface deflections, causing uneven flow over the control surfaces which is solely provided by the propeller. Since the propeller induces drag (another bounded random process) when spinning about this axis, it is expected that the aircraft will rotate quicker in the direction of the motor. This effect can be seen in Figure 3-3(a) which is rotating with the propeller and has a slightly faster rise time than that of Figure 3-3(c) which is rotating against the motor direction. Also, notice that Figure 3-3(c) has a significantly higher overshoot. This is due to the integrator wind-up of the rotational rate and the added amount of control that is necessary to track the desired rate. Figure 3-3(d) and 3-3(b) both show the amount of aileron deflection necessary to perform these commands. Just to maintain a non-rotating aircraft, the aileron must deflect approximately -15 degrees. Note the very large difference in the amount of control authority necessary to rotate against the propeller than with. Since the same controller is used to perform both rotations, there is a tradeoff between the response rate and lack of damping. The quicker the desired response (characteristics of rise time), the higher the integral gain, which causes more oscillation at constant commands.

3.5.3 Transition to Level-Flight

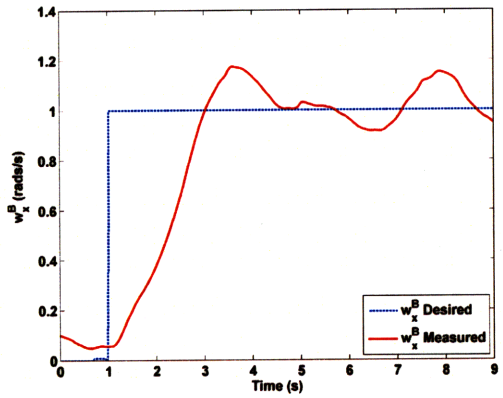
Since inertial velocity serves as the reference command for the outer loop of the quaternion based controller, a joystick is used to manually provide a desired input. A 3-D visualization of the maneuver is shown in Figure 3-4. The goal for this controller is to maintain a commanded inertial z-position while translating from hover to level-flight. Figure 3-5 shows both the desired and measured output of the z position. As expected, a loss in altitude is present during the transition but recovers over time as the velocity command remains constant. Also, since the desired velocity can be viewed as a user defined step input (Figure 3-6(b)), the y-position is a ramp function.



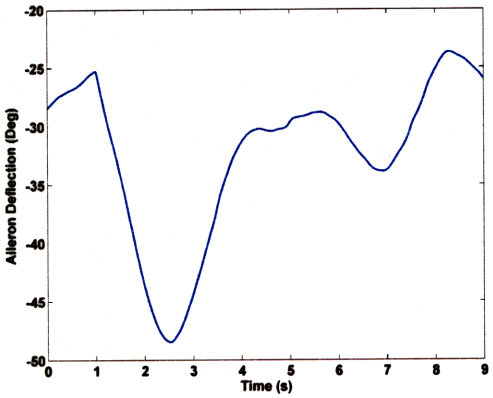
(a) Rotational rate spinning in the same direction as the propeller



(b) Aileron deflection to rotate the aircraft in the same direction as the propeller



(c) Rotational rate spinning in the opposite direction of the propeller



(d) Aileron deflection to rotate the aircraft in the opposite direction of the propeller

Figure 3-3: Roll decoupling maneuver state output

Figure 3-6(a) shows how the body velocities over time, with constant pitch angle (Figure 3-6(c)), stabilize to an equilibrium.

The control commands for this maneuver are shown in Figure 3-6(e) and Figure 3-6(f). Notice that during the transition, the lift dynamics become more influential, and therefore the thrust is used less for supporting the weight and more towards translating the aircraft, which is seen in the dramatic decrease in thrust usage. The elevator is used extensively for the initial rotation, but as the velocity command remains constant, it tends back to a steady-state equilibrium.

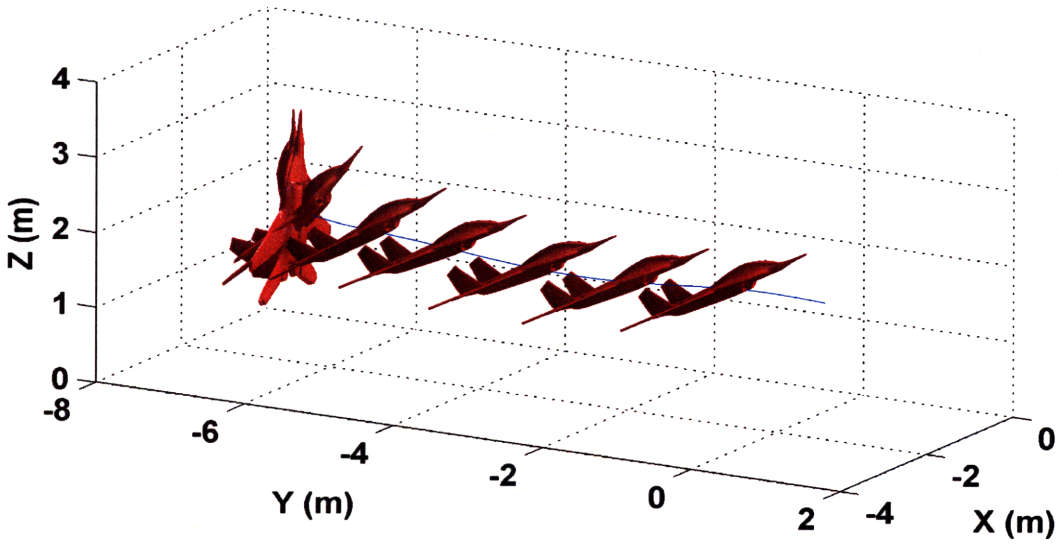


Figure 3-4: 3-D visualization of the quaternion controlled transition maneuver

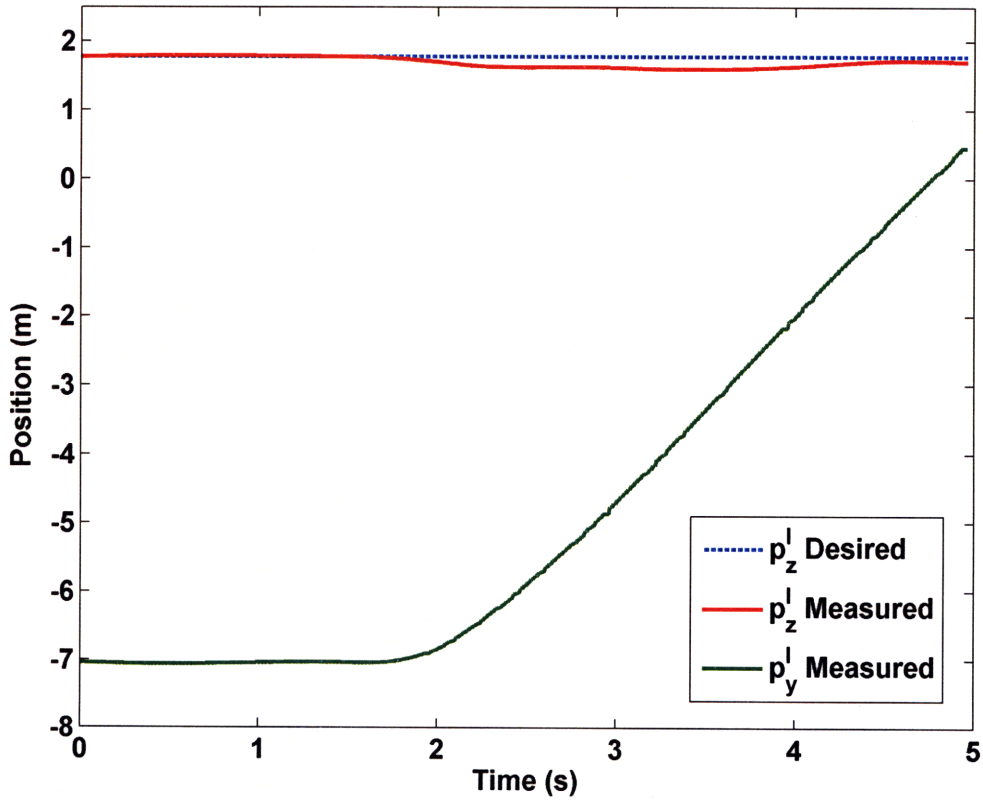
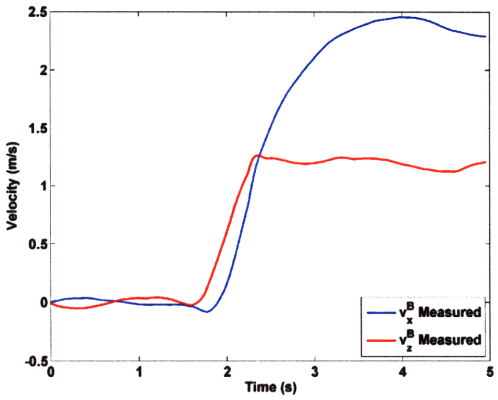
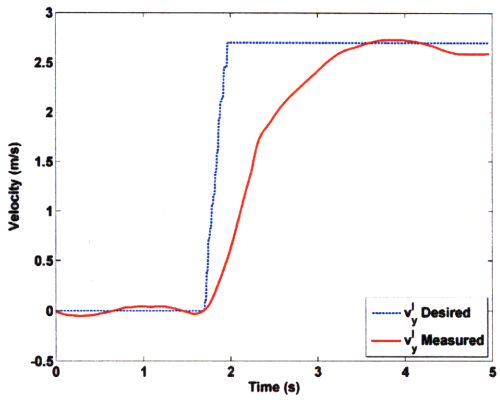


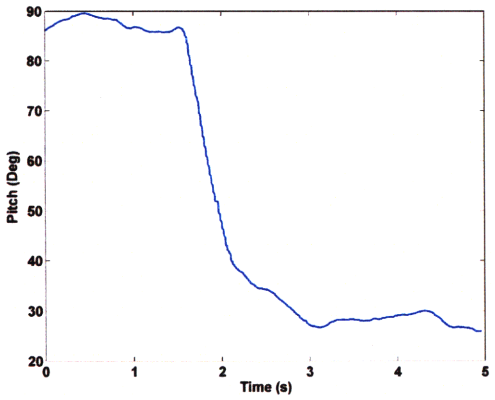
Figure 3-5: Position output for the quaternion-based control transition maneuver



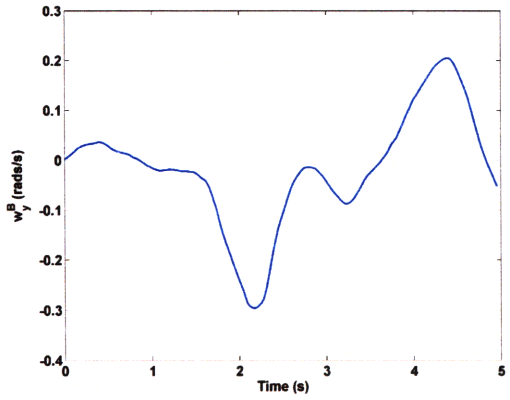
(a) Resultant output body velocities



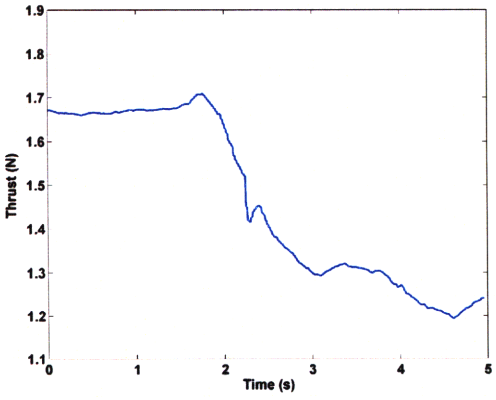
(b) Reference and measured inertial velocity signals



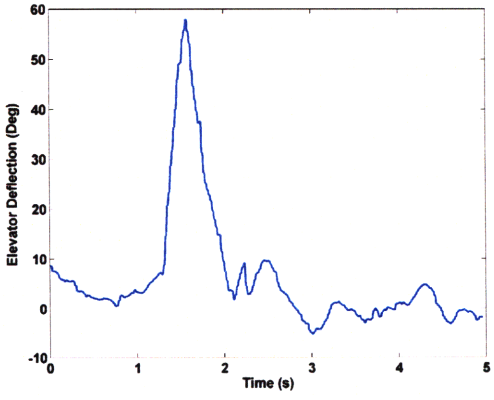
(c) Aircraft pitch angle over time



(d) Measured rotational y body rate



(e) Actuated aircraft thrust



(f) Elevator deflection over time

Figure 3-6: Various state results for the quaternion-based controlled transition to level-flight

Chapter 4

Nonlinear Lyapunov Backstepping Controller

4.1 Introduction

This chapter presents the design of a Lyapunov-based backstepping controller extending the work of [13] to track any feasible smooth time-varying position commands, including the case of constant position commands. Backstepping controllers are useful when states are controlled through other states as is the case with aircraft. The velocity vector is used to control the position, while the aircraft angular velocity and thrust are used to control the velocity vector. To close the loop on the controller, the aerodynamic control surfaces are used to control the angular velocity. The proposed control architecture is shown in Figure 4-1, showing how stepping through the dynamics of the system, the actuators can influence the response of the aircraft.

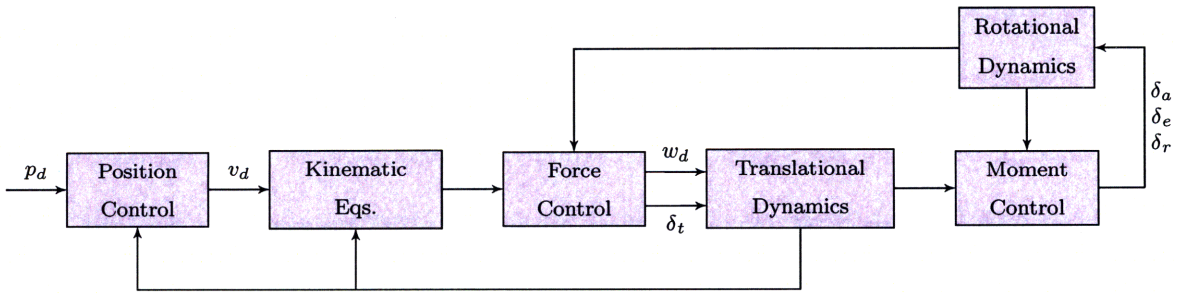


Figure 4-1: Lyapunov-based backstepping control architecture

4.2 Controller Outline

In this section, a brief overview of the Lyapunov back-stepping control scheme developed in [13] is presented. This controller is the core formulation for the control scheme presented in the following sections. For further understanding of the controller development and insight, please see the referenced source.

The position tracking error in the body-fixed frame is defined as

$$\mathbf{e} = R_I^B(\mathbf{p}_a - \mathbf{p}_d)^I. \tag{4.1}$$

By taking the derivative, the dynamic equation for the body-fixed tracking error $\mathbf{e}$ is

$$\begin{aligned}\dot{\mathbf{e}} &= \dot{R}_I^B(\mathbf{p}_a - \mathbf{p}_d)^I + R_I^B\dot{\mathbf{p}}_a^I - R_I^B\dot{\mathbf{p}}_d^I \\ &= -S(\mathbf{w}^B)\mathbf{e} + \mathbf{v}_a^B - R_I^B\dot{\mathbf{p}}_d\end{aligned} \tag{4.2}$$

which uses Equation 2.10 and properties of skew-symmetric matrices.

The first control Lyapunov function is defined as

$$V_1 = \frac{1}{2}\mathbf{e}^T\mathbf{e} \tag{4.3}$$

which is positive definite. Computing the time derivative of (4.3) and applying (4.2), yields

$$\begin{aligned}\dot{V}_1 &= \mathbf{e}^T\dot{\mathbf{e}} \\ &= \mathbf{e}^T(-S(\mathbf{w}^B)\mathbf{e} + \mathbf{v}_a^B - R_I^B\dot{\mathbf{p}}_d^I) \\ &= -\mathbf{e}^T S(\mathbf{w}^B)\mathbf{e} + \mathbf{e}^T(\mathbf{v}_a^B - R_I^B\dot{\mathbf{p}}_d^I) \\ &= \mathbf{e}^T(\mathbf{v}_a^B - R_I^B\dot{\mathbf{p}}_d^I).\end{aligned} \tag{4.4}$$

To achieve stabiltiy in the sense of Lyapunov, $\dot{V}_1$ must be negative definite, which

can be accomplished in (4.4) by setting the desired velocity as

$$\mathbf{v}_d^B = R_I^B \dot{\mathbf{p}}_d - K_e M^{-1} \mathbf{e}. \tag{4.5}$$

This cancels the dynamics and by choosing K_e as a diagonal positive definite matrix, since M is the diagonal mass matrix, the Lyapunov criteria is satisfied.

Since the desired velocity needs to follow Equation (4.5), the error between the desired and actual velocities must be forced to zero. Define the velocity error

$$\mathbf{z}_1 = \mathbf{v}_a^B - \mathbf{v}_d^B = \mathbf{v}_a^B - R_I^B \dot{\mathbf{p}}_d^I + K_e M^{-1} \mathbf{e}. \tag{4.6}$$

Equation (4.4) can then be re-written as

$$\dot{V}_1 = -K_e \mathbf{e}^T M^{-1} \mathbf{e} + \mathbf{e}^T \mathbf{z}_1 \tag{4.7}$$

and the time derivative of $\mathbf{z}_1$ can be calculated as

$$\begin{aligned}
M\dot{\mathbf{z}}_1 &= M\dot{\mathbf{v}}_a^B - M\dot{R}_I^B \dot{\mathbf{p}}_d^I - MR_I^B \ddot{\mathbf{p}}_d^I + K_e \dot{\mathbf{e}} \\
&= -S(\mathbf{w}^B) M\mathbf{v} + \mathbf{f}_v + \mathbf{g}_v \delta_t + MS(\mathbf{w}^B) R_I^B \dot{\mathbf{p}}_d - MR_I^B \ddot{\mathbf{p}}_d - K_e S(\mathbf{w}^B)\mathbf{e} + K_e \mathbf{v} - K_e R_I^B \dot{\mathbf{p}}_d \\
&= -S(\mathbf{w}^B) M\mathbf{v} + \mathbf{f}_v + \mathbf{g}_v \delta_t + MS(\mathbf{w}^B) R_I^B \dot{\mathbf{p}}_d - MR_I^B \ddot{\mathbf{p}}_d - K_e S(\mathbf{w}^B)\mathbf{e} + K_e \mathbf{z}_1 - K_e^2 M^{-1}\mathbf{e} \\
&= -S(\mathbf{w}^B) M[\mathbf{z}_1 + R_I^B \dot{\mathbf{p}}_d - K_e M^{-1}\mathbf{e}] + \mathbf{f}_v + \mathbf{g}_v \delta_t + MS(\mathbf{w}^B) R_I^B \dot{\mathbf{p}}_d - MR_I^B \ddot{\mathbf{p}}_d - K_e S(\mathbf{w}^B)\mathbf{e} \\
&\quad + K_e \mathbf{z}_1 - K_e^2 M^{-1}\mathbf{e} \\
&= S(M\mathbf{z}_1)\mathbf{w}^B + [S(MR_I^B \dot{\mathbf{p}}_d) - MS(R_I^B \dot{\mathbf{p}}_d)]\mathbf{w}^B + \mathbf{g}_v \delta_t + \mathbf{f}_v - MR_I^B \ddot{\mathbf{p}}_d + K_e \mathbf{z}_1 - K_e^2 M^{-1}\mathbf{e} \\
&= S(M\mathbf{z}_1)\mathbf{w}^B + \mathbf{g}_v \delta_t + \mathbf{h}
\end{aligned} \tag{4.8}$$

where,

$$\mathbf{h} = \mathbf{f}_v - MR_I^B \ddot{\mathbf{p}}_d + K_e \mathbf{z}_1 - K_e^2 M^{-1}\mathbf{e}. \tag{4.9}$$

Note that

$$\mathbf{v} = \mathbf{z}_1 + R_I^B \dot{\mathbf{p}}_d - K_e M^{-1}\mathbf{e}$$

and because of diagonal M

$$S(MR_I^B\dot{\mathbf{p}}_d) - MS(R_I^B\dot{\mathbf{p}}_d) = MS(R_I^B\dot{\mathbf{p}}_d) - MS(R_I^B\dot{\mathbf{p}}_d) = 0$$

In compact notation

$$M\dot{\mathbf{z}}_1 = S(M\mathbf{z_1})\mathbf{w}^B + \mathbf{g}_v\delta_t + \mathbf{h}. \tag{4.10}$$

The last term in Equation (4.7) works against the need for negative definiteness of $\dot{V}_1$ to meet the Lyapunov criteria. Unfortunately, it is not always possible to make $\mathbf{z}_1$ zero since it eliminates the direct coupling between the translational and the rotational dynamics. This can be seen by setting $\mathbf{z}_1$ to zero which also implies $\dot{\mathbf{z}}_1$ is zero, and in Equation (4.10), $\mathbf{g}_v\delta_t = \mathbf{h}$ which implies no coupling since $\mathbf{h}$ is a function of force terms. However, $\mathbf{z}_1$ can be made a constant vector $\delta \in \mathbb{R}^3$ such that the error

$$\varphi = \mathbf{z}_1 - \delta \tag{4.11}$$

is forced to zero. Based on Equation (4.11), a second control-Lyapunov function defined as

$$V_2 = V_1 + \frac{1}{2}\varphi^T M^2 \varphi \tag{4.12}$$

is defined. The time derivative of V_2 can be written as

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + \varphi^T M(M\dot{\mathbf{z}}_1) \\ &= -K_e\mathbf{e}^T M^{-1}\mathbf{e} + \mathbf{e}^T\delta + \varphi^T(M[\mathbf{g}_v \quad S(M\delta)][\delta_t \quad \mathbf{w}^B]^T + M\mathbf{h} + \mathbf{e}).\end{aligned} \tag{4.13}$$

which uses the fact that

$$\varphi^T MS(M\varphi) = 0 \quad \text{and} \quad \mathbf{e}^T\varphi = \varphi^T\mathbf{e}.$$

For simplicity, define

$$B = [\mathbf{g}_v \quad S(M\delta)] \quad \in \mathbb{R}^{3\times 4} \tag{4.14}$$

$$\zeta = [\delta_t \quad \mathbf{w}^B]^T \quad \in \mathbb{R}^4 \tag{4.15}$$

and thus Equation (4.13) in this compact notation becomes

$$\dot{V}_2 = -K_e \mathbf{e}^T M^{-1}\mathbf{e} + \mathbf{e}^T\delta + \varphi^T(MB\zeta + M\mathbf{h} + \mathbf{e}). \tag{4.16}$$

Equation (4.15) harbors the first control actuation ability in this control design. The next step extracts this control variable by satisfying the Lyapunov criteria for V_2. Note that one can choose $\delta = M^{-1}\mathbf{g}_v\bar{\epsilon}$, where $\bar{\epsilon}$ is a positive constant such that the matrix BB^T is invertible,

$$BB^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\bar{\epsilon} \\ 0 & 0 & \bar{\epsilon} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \bar{\epsilon} \\ 0 & -\bar{\epsilon} & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \bar{\epsilon}^2 & 0 \\ 0 & 0 & \bar{\epsilon}^2 \end{bmatrix}. \tag{4.17}$$

The vector ζ, which consists of the desired thrust and rotational rate, serves as a virtual control that forces $\dot{V}_2$ to satisfy the Lyapunov condition. This is achieved by setting ζ as

$$\zeta_d = B^T(BB^T)^{-1}(-\mathbf{h} - M^{-1}\mathbf{e} - M^{-1}K_\varphi\varphi) \tag{4.18}$$

where $K_\varphi \in \mathbb{R}^{3\times 3}$ is a diagonal symmetric positive definite matrix. To obtain the first control input δ_t, extract the first entry of ζ_d as

$$\delta_t = [1 \quad 0_{1\times 3}]\zeta_d. \tag{4.19}$$

Now, the final error variable that must be forced to zero is

$$\mathbf{z}_2 = \mathbf{w}^B - [0_{3\times1} \quad I_{3\times3}]\zeta_d. \tag{4.20}$$

By combining Equation (4.15) and (4.20) to form $\zeta = [0 \quad \mathbf{z}_2]^T + I_{4\times4}\zeta_d$ and noting that $\varphi^T MB[0 \quad \mathbf{z}_2]^T = \varphi^T MS(M\delta)\mathbf{z}_2$, we can rewrite Eq. 4.16 with δ_t given by Eq. 4.19 as

$$\dot{V}_2 = -K_e\mathbf{e}^T M^{-1}\mathbf{e} + \mathbf{e}^T\delta - \varphi^T K_\varphi\varphi + \varphi^T MS(M\delta)\mathbf{z}_2. \tag{4.21}$$

The third and final control-Lyapunov functions is

$$V_3 = V_2 + \frac{1}{2}\mathbf{z}_2^T J\mathbf{z}_2 = \frac{1}{2}\mathbf{e}^T\mathbf{e} + \frac{1}{2}\varphi^T M^2\varphi + \frac{1}{2}\mathbf{z}_2^T J\mathbf{z}_2. \tag{4.22}$$

The time derivative of V_3 is computed as

$$\begin{aligned}\dot{V}_3 = &-K_e\mathbf{e}^T M^{-1}\mathbf{e} + \mathbf{e}^T\delta - \varphi^T K_\varphi\varphi + \mathbf{z}_2^T(G_w\mathbf{u}_w - S(\mathbf{w}^B)J\mathbf{w}^B + \mathbf{f}_w \\ &- [0_{3\times1} \quad J]\dot{\zeta}_d - S(M\delta)M\varphi),\end{aligned} \tag{4.23}$$

where the last term is due to the fact $\varphi^T MS(M\delta)\mathbf{z}_2 = -\mathbf{z}_2^T S(M\delta)M\varphi$. In order to satisfy the Lyapunov criteria, $\mathbf{u}_w$ is chosen as

$$\mathbf{u}_w = G_w^{-1}(S(\mathbf{w}^B)J\mathbf{w}^B - \mathbf{f}_w + [\ 0_{3\times1} \quad J\ ]\dot{\zeta}_d + S(M\delta)M\varphi - K_{z_2}\mathbf{z}_2). \tag{4.24}$$

where $K_{z_2} \in \mathbb{R}^{3\times3}$ is a diagonal symmetric positive definite matrix. Note that G_w is never singular, even at hover due to the control authority produced by the propeller airflow over the control surfaces. Substituting in $\mathbf{u}_w$ from (4.24), the time derivatives of V_3 is then

$$\dot{V}_3 = -K_e\mathbf{e}^T M^{-1}\mathbf{e} + \mathbf{e}^T\delta - \varphi^T K_\varphi\varphi - \mathbf{z}_2^T K_{z_2}\mathbf{z}_2. \tag{4.25}$$

By selecting appropriate positive definite matrices K_e, K_ϕ, K_{z_2}, the user is able

to produce a control scheme with stable trajectory tracking capabilities. As can be seen from Equation (4.25), the Lyapunov function is not always negative definite because of the δ term introduced in Equation (4.11). Hence, maintaining this term to a sufficiently small value will aid in stability, with the trade-off of having decreased coupling between the rotational and translational dynamics. It is important to note that the above control design approach can not guarantee the stability results during implementation because it assumes no parametric uncertainty between the model and actual vehicle.

4.2.1 Simulation

This control design is tested with the aircraft model outlined in Chapter 2. Two types of tests are performed. The first test demonstrates the controller's capability to stabilize the aircraft in hover, while the second demonstrates its take-off to hover capabilities.

For the hover test, the aircraft was given a commanded vector position (in meters) of

$$\mathbf{p}_d = \begin{bmatrix} 10 \\ 0 \\ 6 \end{bmatrix}. \tag{4.26}$$

To test this controller's robustness to initial conditions, a small offset is placed in the x and z-axes, and its correction can be seen in Figure 4-2. There are a few things to note from this test. First, the controller corrects the aircraft's position in the x-direction. Second, there is a very noticeable error in the z-axis which will be explained further in Section 4.3. The controller, however, seems to be very dependent on the error constant that is chosen in Equation (4.17). It created a coupling between the rotational and translational dynamics and the larger the constant is made, the simulation converges to a result faster, but a larger steady state error is induced.

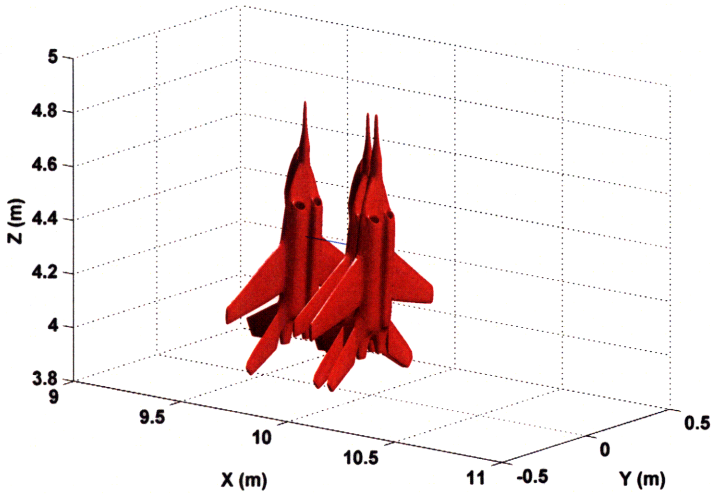


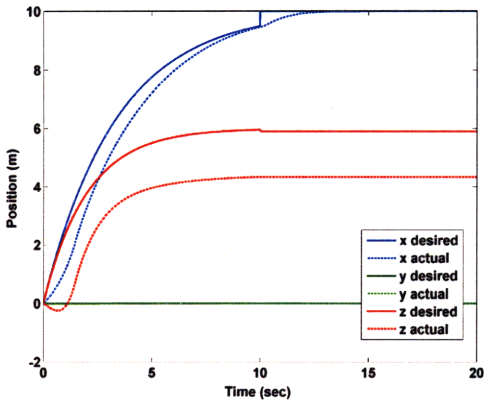
Figure 4-2: Simulated Lyapunov-based backstepping control in hover with initial condition offsets

For the take-off to hover test, the following trajectory is commanded:

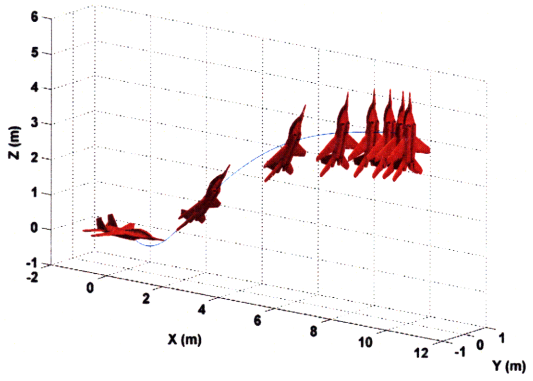
$$\mathbf{p}_d(t) = \begin{cases} \begin{bmatrix} 10(1-e^{-0.3t}) \\ 0 \\ 6(1-e^{-0.5t}) \end{bmatrix} & 0 \leq t < 10 \\ \begin{bmatrix} 10 \\ 0 \\ 6 \end{bmatrix} & 10 \leq t < 20. \end{cases} \tag{4.27}$$

Results from the simulation are shown in Figure 4-3. Note that a similar behavior can be seen here as in the hover test: there is a significant difference between the actual z and desired output, and the x-error converges to zero as $t \to \infty$. For this particular maneuver, the value of $\bar{\epsilon}$ from Equation (4.17) had to be increased compared to the hover test. This is expected since there is more interaction between the rotational and translational dynamics for such a desired trajectory.

When attempting to place this control design on the actual Clik aircraft, issues arise with the stability of the vehicle starting at hover. In Section 4.3, a linearization is performed in order to gain more insight and intuition on the control needed at



(a) Desired and actual parametric position output as a function of time



(b) 3-D visualization of the take-off to hover maneuver

Figure 4-3: Lyapunov-based backstepping take-off to hover simulation

hover. The key questions that are answered during this linearization process are the nature of steady-state errors and the controller's sensitivity to the $\bar{\epsilon}$ term.

4.3 Linearized Hover Controller

This section outlines a linearization of the nonlinear controller near hover to study the behavior in the linear domain about hover. By performing this linearization, a simplistic view of this controller concept can be understood and analyzed.

Near the hover orientation, $u_\infty \approx 0$, $u_p \approx \sqrt{\frac{mg}{2\rho A_d}}$ from Equation (2.3) and the direction cosine matrix (DCM) at a +90 degree pitch angle (the body x and inertial frame z axis are aligned) is

$$R_B^I = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

which transforms the equations of motion (2.9), (2.14) and (2.10) respectively into

$$\dot{\mathbf{p}} \overset{hover}{\approx} \begin{bmatrix} -v_z \\ v_y \\ v_x \end{bmatrix}, \qquad \mathbf{f}_v \overset{hover}{\approx} \begin{bmatrix} -mg \\ 0 \\ 0 \end{bmatrix},$$

$$\dot{R}_B^I \overset{hover}{\approx} \begin{bmatrix} w_y & -w_x & 0 \\ w_z & 0 & -w_x \\ 0 & -w_z & w_y \end{bmatrix}.$$

Starting with the nonlinear thrust command from Equation 4.19 and noting that

$$\begin{aligned} \zeta_d &= B^T(BB^T)^{-1}(-\mathbf{h} - M^{-1}\mathbf{e} - M^{-1}K_\varphi\varphi) \\ &\equiv B^T(BB^T)^{-1}\mathbf{W} \end{aligned}$$

with

$$B^T(BB^T)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\epsilon} \\ 0 & -\frac{1}{\epsilon} & 0 \end{bmatrix},$$

the thrust is the first entry of $\mathbf{W} \in \mathbb{R}^{3\times 1}$. Noting that the general vector form of $\mathbf{h}(1)$ denotes the first entry in the vector, and the thrust command can be reformulated as

$$\delta_t = -\mathbf{h}(1) - M^{-1}\mathbf{e}(1) - M^{-1}K_\varphi\varphi(1).$$

Applying the near hover approximations, the term-by-term linearization of $\mathbf{W}$ be-

comes

$$
\begin{aligned}
-\mathbf{h}(1) &= mg - k_e v_x, \\
-M^{-1}\mathbf{e}(1) &= \frac{p_z - p_z^d}{m}, \\
-M^{-1}K_\varphi \varphi(1) &= -k_\varphi \left(v_x + \frac{k_e(p_z - p_z^d)}{m} - \frac{\bar{\epsilon}}{m} \right),
\end{aligned}
$$

yielding the linearized thrust term

$$
\delta_t \overset{hover}{\approx} mg - (k_e + k_\varphi)v_x - \left(\frac{1 + k_e k_\varphi}{m} \right)(p_z - p_z^d) - \frac{k_\varphi \bar{\epsilon}}{m}, \tag{4.28}
$$

which is a Proportional/Derivative (PD) controller, given by the two inner terms, with a feed-forward weight term. Note that the last term in Equation (4.28) is both a function of k_φ and $\bar{\epsilon}$. This is the explanation to why, in the body x axis, the controller fails to eliminate the steady-state error. The larger ϵ is, the larger this steady-state error will be.

The aircraft control surface law is given in Equation 4.24 and rewritten here as $\mathbf{u}_w = G_w^{-1}\mathbf{T}$ where $\mathbf{T} \in \mathbb{R}^{3\times 1}$ is given as

$$
\mathbf{T} = S(\mathbf{w}^B)J\mathbf{w}^B - \mathbf{f}_w + \begin{bmatrix} 0_{3\times 1} & J \end{bmatrix} \dot{\zeta}_d + S(M\delta)M\varphi - K_{z_2}\mathbf{z}_2.
$$

With the hover assumptions, the moment decoupling matrix terms are

$$
\begin{bmatrix} g_{11} \\ g_{22} \\ g_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\rho u_{\mathrm{p_a}}^2 C_{L_{\delta_{aw}}} A_{ap} L_{ap} \\ \frac{1}{2}\rho u_{\mathrm{p_r}}^2 C_{L_{\delta_e}} A_e L_{ep} \\ \frac{1}{2}\rho u_{\mathrm{p_r}}^2 C_{L_{\delta_r}} A_r L_{rp} \end{bmatrix}.
$$

The following performs a term-by-term expansion on $\mathbf{T}$ using the hover conditions and assumptions:

$$S(\mathbf{w}^B)J\mathbf{w}^B = \begin{bmatrix} w_y w_z(I_{zz} - I_{yy}) \\ w_x w_z(I_{xx} - I_{zz}) \\ w_x w_y(I_{yy} - I_{xx} \end{bmatrix} \overset{hover}{\approx} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.29}$$

$$\mathbf{f}_w \overset{hover}{\approx} \begin{bmatrix} -M_{acc} - M_{drag} \\ -\frac{1}{2}\rho L_e L_{e,p} A_e u_{\mathrm{p_r}} w_y \\ -\frac{1}{2}\rho L_r L_{r,p} A_r u_{\mathrm{p_r}} w_z \end{bmatrix} \tag{4.30}$$

$$\begin{bmatrix} 0_{3\times 1} & J \end{bmatrix} \dot{\zeta}_d \overset{hover}{\approx} \begin{bmatrix} 0 \\ \frac{I_{yy}}{\epsilon}\left(-w_y mg - v_z \frac{1+k_\varphi k_e}{m}\right) \\ -\frac{I_{zz}}{\epsilon}\left(-w_z mg - v_y \frac{1+k_\varphi k_e}{m}\right) \end{bmatrix} \tag{4.31}$$

$$MS(\delta)M\varphi \overset{hover}{\approx} \begin{bmatrix} 0 \\ -\bar{\epsilon} m\left(v_z - \frac{k_e}{m}(p_x - p_x^d)\right) \\ \bar{\epsilon} m\left(v_y + \frac{k_e}{m}(p_y - p_y^d)\right) \end{bmatrix} \tag{4.32}$$

$$K_{z_2}\mathbf{z}_2 \overset{hover}{\approx} \frac{K_{z_2}}{\bar{\epsilon}} \begin{bmatrix} 0 \\ v_z(-k_e - \frac{k_\varphi}{m}) + \frac{m+k_e k_\varphi}{m^2}(p_x - p_x^d) \\ v_y(k_e + \frac{k_\varphi}{m}) + \frac{m+k_e k_\varphi}{m^2}(p_y - p_y^d) \end{bmatrix} \tag{4.33}$$

Combining terms Eq.(4.29-4.33) yields

$$\mathbf{T} \overset{hover}{\approx} \begin{bmatrix} M_{acc} + M_{drag} \\ w_y(A_{ep} u_{\mathrm{p_r}} + B_{ep}) + v_z(C_{ep}) + (p_x - p_x^d)D_{ep} \\ w_z(A_{ry} u_{\mathrm{p_r}} + B_{ry}) + v_y(C_{ry}) + (p_y - p_y^d)D_{ry} \end{bmatrix} \tag{4.34}$$

where,

$$A_{ep} = \frac{1}{2}\rho L_e L_{e,p} A_e \qquad A_{ry} = \frac{1}{2}\rho L_r L_{r,p} A_r$$
$$B_{ep} = -\frac{I_{yy}}{\bar{\epsilon}} mg \qquad B_{ry} = -\frac{I_{zz}}{\bar{\epsilon}} mg$$
$$D_{ep} = \frac{\bar{\epsilon}^2 m^2 k_e - k_e k_\varphi - m}{m^2 \bar{\epsilon}} \qquad D_{ry} = \frac{\bar{\epsilon}^2 m^2 k_e - k_e k_\varphi - m}{m^2 \bar{\epsilon}}$$

$$C_{ep} = \frac{k_{z_2}(m k_e + k_\varphi) - I_{yy}(1 + k_\varphi k_e) - \bar{\epsilon}^2 m^2}{\bar{\epsilon} m}$$
$$C_{ry} = \frac{m k_{z_2} k_e + k_\varphi k_{z_2} + I_{zz}(1 + k_\varphi k_e) - \bar{\epsilon}^2 m^2}{\bar{\epsilon} m}$$

Therefore the total deflection control in a linearized hover condition is

$$\mathbf{u}_w = \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} \overset{hover}{\approx} G_w \begin{bmatrix} M_{acc} + M_{drag} \\ w_y(A_{ep} u_p + B_{ep}) + v_z(C_{ep}) + (p_x - p_x^d) D_{ep} \\ w_z(A_{ry} u_p + B_{ry}) + v_y(C_{ry}) + (p_y - p_y^d) D_{ry} \end{bmatrix}. \tag{4.35}$$

Equation (4.35) shows that, when linearized about hover, the controller is similar to what might be expected. In particular, the deflection of the aileron is a function of the thrust and cancels the moment caused by the rotation of the propeller. The rudder and elevator control surfaces have similar feedback strategies that use a PD controller for position along with a rate attitude term.

4.3.1 Flight Test

This linearized version of the nonlinear controller has been implemented on the Clik indoor aerobatic aircraft. In this linearized controller test, the attitude controller presented in Chapter 3 is used prior to the linearized controller to attain a desired hover attitude and position in flight to ensure proper initial conditions. Once this attitude is obtained, the controllers are interchanged, and the last inertial position vector becomes the linearized controller's commanded position, shown in Figure 4-4. The switch becomes apparent when the reference command becomes constant.

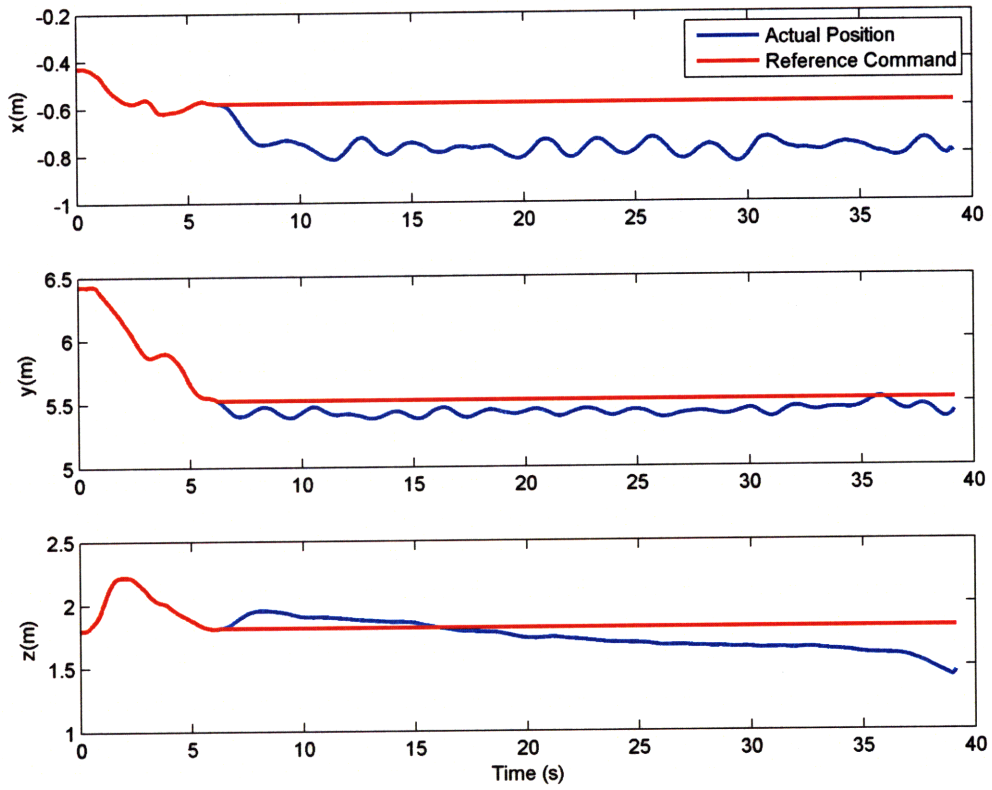


Figure 4-4: Experimental linearized Lyapunov-based controller position data about hover.

Since the controller is a PD controller, some steady-state error is seen. The z-axis behavior is due to the drop in battery voltage over time. In the Lyapunov sense of linearized stability at an equilibria, the equilibrium point is asymptotically stable for the actual nonlinear system [24]. Thus, the conclusion can be drawn that the aircraft will maintain stability with the nonlinear controller while at hover.

The above implementation of the linearized controller only takes the form of the controller found in Section 4.3. The reason for this is to acquire an envelope of gains that will stabilze the aircraft. Once this envelope is found, a system of equations is developed and a feasibility check is done to verify whether the form of Equations (4.28) and (4.35) can be satisfied. For instance, in Eq. (4.28), the gains that stabilized the

system take the form

$$\delta_t \overset{hover}{\approx} mg - 0.34v_x - 0.95(p_z - p_z^d). \tag{4.36}$$

To satisfy this criteria, the gains above would have to satisfy

$$0.34 = (k_e + k_\varphi) \tag{4.37}$$

$$0.95 = \left(\frac{1 + k_e k_\varphi}{m}\right) \tag{4.38}$$

along with the other gains used for the control surfaces as well as satisfying the Lyapunov criteria by remaining positive definite. Unfortunately, the criteria above could not be met. In order to satisfy the system of equations, the Lyapunov criteria had to be relaxed (meaning allow some values to take on negative values occasionally allowing $\dot{V} > 0$), which led to an unstable system in other flight regimes. In the next section, a controller is devised that utilizes part of the controller properties outlined in Section 4.2 to obtain the desired state values, but uses the quaternion controller to reach these states.

4.4 Lyapunov Quaternion Control

4.4.1 Introduction

In this section, a Lyapunov-based quaternion controller with state feed-back is presented. This control design combines the Lyapunov algorithm developed in [13] with the quaternion based controller from [15] and is shown in Figure 4-5. The user inputs a desired smoothly time-varying continuous trajectory (including constant commands), assumed to be twice differentiable, which will be used in both the attitude and thrust actuation.

Beginning with the attitude controller, the Lyapunov algorithm dictates the desired rotational velocities through Equation (4.20) as $\mathbf{w}_d^B = [0_{3\times1} \quad I_{3\times3}]\zeta_d$. To maintain the same notation from above, the rotational update from Equation (2.10) pro-

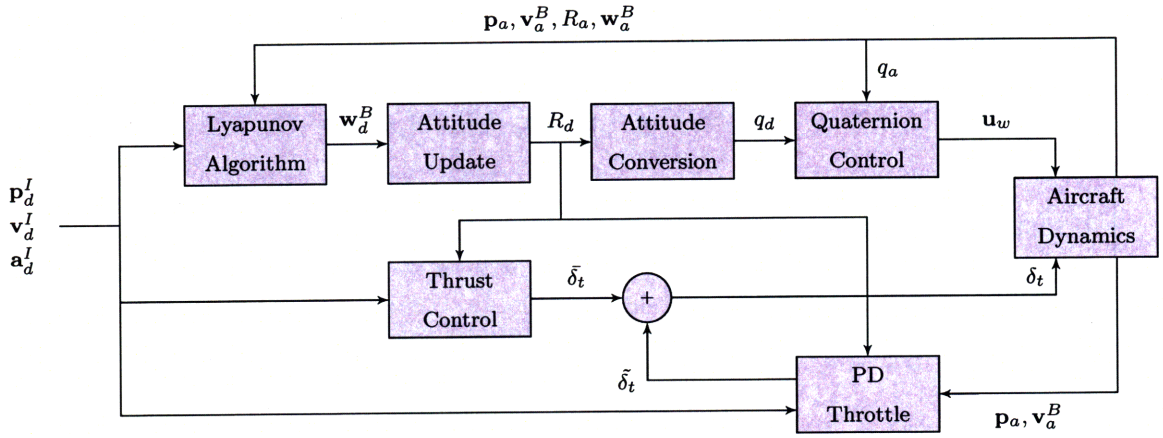


Figure 4-5: Lyapunov quaternion control architecture

duces the desired DCM which represents the desired attitude. This DCM then needs to be converted to a desired quaternion. The relationship between these two attitude descriptors is given in [25] as

$$R_I^B = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} q_o^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y + q_z q_o) & 2(q_x q_z - q_y q_o) \\ 2(q_x q_y - q_z q_o) & q_o^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z + q_x q_o) \\ 2(q_x q_z + q_y q_o) & 2(q_y q_z - q_x q_o) & q_o^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}. \tag{4.39}$$

By using the diagonal elements alone, one may determine the magnitudes of the individual quaternion elements. The other six off-diagonal elements determine the signs.

Many algorithms have been developed to convert from DCM to quaternion. For this design approach Shepperd's algorithm is used [26] and proceeds as outlined in Table 4.1. The reason Shepperd's algorithm is used is not only to provide the correct signs for the quaternion, but also to avoid dividing by a very small number, creating a singularity in the solution for q_d. This desired quaternion is used as the reference desired quaternion in the quaternion-based controller as given in Equation (3.4).

The thrust controller predicts a thrust feed-forward term, denoted as $\bar{\delta}_t$,based on the user defined desired state inputs which are inserted in the model. The primary goal for this design is to transition from hover to level flight by rotating about the

Table 4.1: Algorithm to convert from DCM to quaternion

	Algorithm 4.1. Shepperd's Algorithm
1:	Compute the trace of R_I^B
2:	Define r_{44}:
	$r_{44} = \text{trace}\{R_I^B\} = T$
3:	Define the elements p_i^2
	$p_i^2 = 1 + 2d_{ii} - T \qquad i = 1, 2, 3, 4$
4:	Based on Eq. 4.39 write the relations between p_i and elements in R_I^B
5:	Find the largest of the d_{ii}:
	$\lvert d_{ii,\max}\rvert = \text{Max}\{\lvert d_{11}\rvert, \lvert d_{22}\rvert, \lvert d_{33}\rvert, \lvert d_{44}\rvert\}$
6:	For that d_{ii}, compute the corresponding p_i using Step 3
	$p_i = \sqrt{1 + 2d_{ii} - T}$
7:	With this p_i, use Step 4 to obtain the rest of the $p_i's$
8:	Divide all four $p_i's$ by 2 to obtain the individual quaternion components

body-fixed y-axis. The main control actuation is performed by the thrust and elevator, and thus the $S(\mathbf{w}^B)M\mathbf{v}^B$ term may be omitted resulting in

$$\mathbf{g}_v\delta_t = M\dot{\mathbf{v}}^B - \mathbf{f}_v. \tag{4.40}$$

The first entry of (4.40) is the feed-forward thrust term. A feedback law is used to help correct the thrust controller if there is an error between the commanded and desired states. This feedback law uses the error in the state variables defined in Equation (4.6) for the velocity and Equation (4.1) for the position in the body frame but modified by the desired DCM in the following manner:

$$e_v^B = v_a^B - R_{I_d}^B v_d^I \tag{4.41}$$

$$e_p^B = R_{I_d}^B(p_a - p_d). \tag{4.42}$$

So for this thrust controller formulation, the errors are defined about the desired trajectory error through the use of the desired orientation. The final thrust controller is given as,

$$\delta_t = \bar{\delta}_t + K_p e_p^B + K_d e_v^B. \tag{4.43}$$

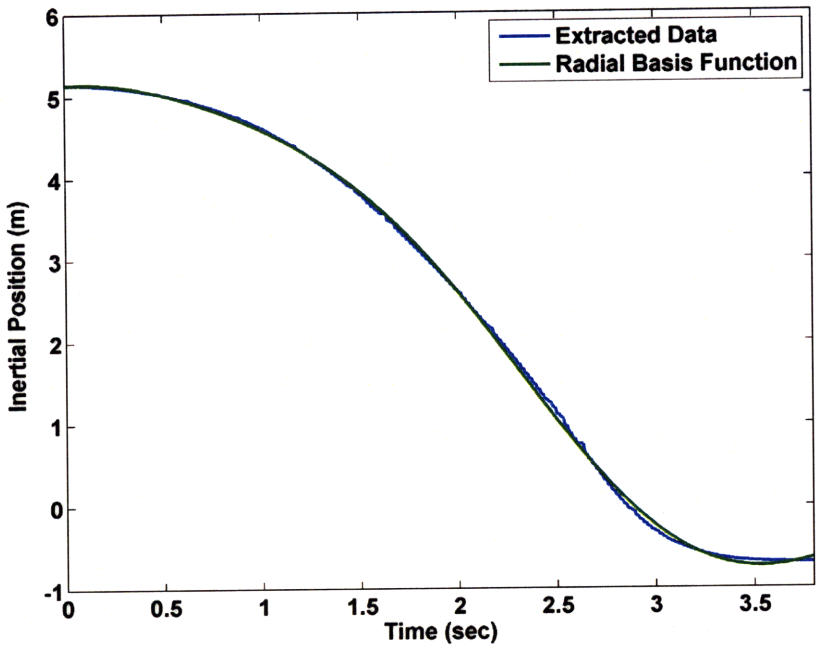


Figure 4-6: Sample transition trajectory data with radial basis least square fit

The major difference between this controller and the quaternion controller described in Chapter 3 is that the rotational and translational dynamics are cognisant of each other through the desired DCM. This is done by providing the thrust controller with the desired DCM and the control surfaces with the velocity and position errors in the body frame.

In the following sections, simulation and hardware implementation of this controller are presented. Since the quaternion-based controller has been successfully implemented, it serves as a guideline for the type of trajectory that is feasible for the aircraft to follow.

4.4.2 Simulation

Data taken from multiple trials of the transition maneuver are all quite similar and is shown in Figure 4-6. For the ease of computation when taking multiple derivatives, a normally distributed gaussian radial basis function is used to fit this curve and is

Table 4.2: Radial basis function values: $\sigma = 1.2$

i	t_i	α_i
1	0.0	7.7817
2	1.0	-5.2557
3	2.0	10.9059
4	3.0	-8.2935
5	4.0	2.5327

given by

$$p_{d_y} = \sum_{i=1}^{N} \alpha_i e^{\frac{-(t-t_i)^2}{2\sigma^2}}. \tag{4.44}$$

The t_i values represent the center of the normal gaussian distributed function and are placed discretely along the length of the maneuver time-line. The α_i's represent the weighting functions and are determined using the matrix methods of linear least squares [27]. The variances of the exponential functions is a function of the number of sample points; the denser the sample points, the smaller the variance. For this function, five basis functions are used (values shown in Table 4.2) and a comparison is shown in Figure 4-6. This function provides continuous both first and second order derivatives and is used as the desired inertial position function in the simulation and implementation of the Lyapunov quaternion controller.

In this simulation, the goal is to gain insight on the expected reaction of the vehicle with the Lyapunov quaternion controller implemented. It also provides a basis of how well the quaternion controller can track the type of desired rotational rates required to perform this maneuver. The desired position for the trajectory is denoted as given in Eq. (4.44), p_{d_y}, and the total desired position in $\mathbb{R}^3$ is given as

$$\mathbf{p}_d = \begin{bmatrix} 0.0 \\ p_{d_y} \\ 1.2 \end{bmatrix} \tag{4.45}$$

Since it was determined that the transition maneuver will be done about the body

y-axis, the above control design is only placed on the thrust and elevator control actuators so that the body x-z plane is parallel to the inertial y-z axis. The other surfaces are controlled through the quaternion based controller to maintain the above desired position.

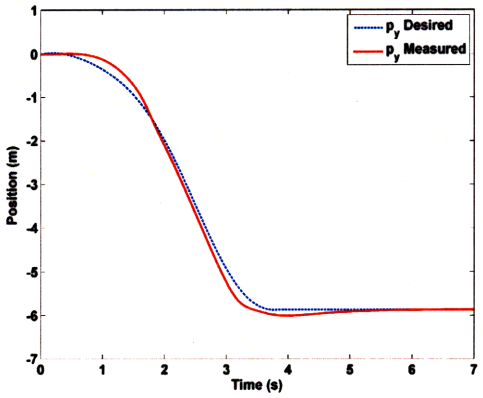
Some trade-offs are apparent between the "quickness" of the desired rotational rate and the overall performance of the vehicle. Figure 4-7(f) shows that there is some lag between the desired and measured y-rotational velocity. This is due to several reasons. Their is a modeled lag in the actuators in an attempt to bring more realism to the system, and the attitude update equation is actually running a time step (50Hz) behind the actual system. It is also apparent that the more aggressive the desired rotational rate is (a product of the selective gain choice being close to the point of instability), the more of a "wiggling" phenomenon takes place during the transition seen between 1-3 seconds.

This phenomenon is also captured in the body velocity output shown in Figure 4-7(c) and Figure 4-7(d) between the same time range. The most substantial advantage to this system design is the controllers ability to track the desired trajectory. Figure 4-7(a) and Figure 4-7(a) shows the output of both the y and z-positions which are followed very well. Position in the z-axis is more apparent but the error is less than 0.2 meters.

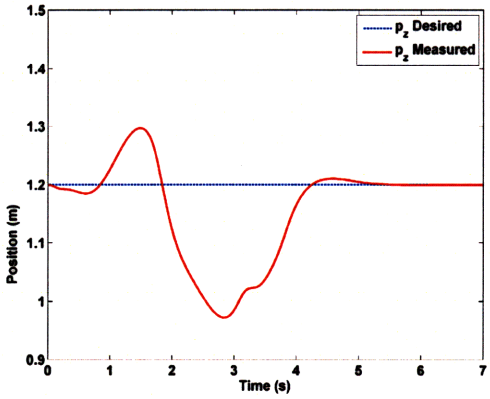
4.4.3 Hardware Implementation

Hover to Hover Maneuver

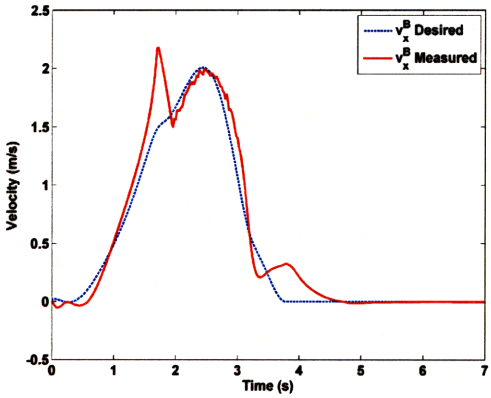
The same trajectory used in simulation also provided a reference command for hardware implementation. All the resultant state data for this experiment are shown in Figure 4-9. The first figure of merit of these plots is the similarities to the simulated data. The position tracking, in particular, in Figure 4-9(a) and Figure 4-9(b) are remarkably similar to that in the simulation. The body velocities in Figures 4-9(c) exhibit a bit of lag due to possible un-modeled delays in the motor and/or actuator dynamics.



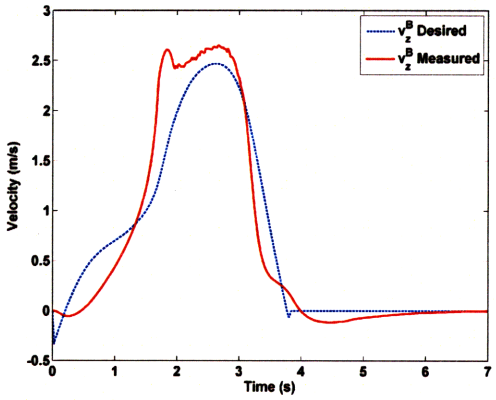
(a) Simulated y position output



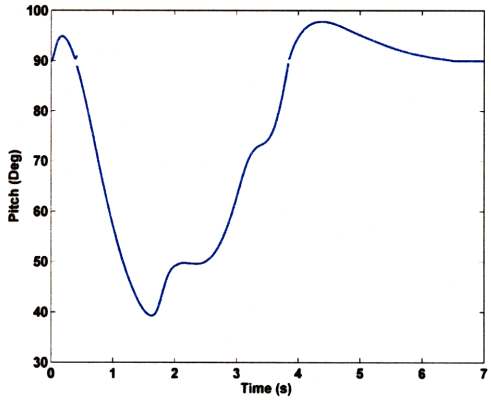
(b) Simulated z position output



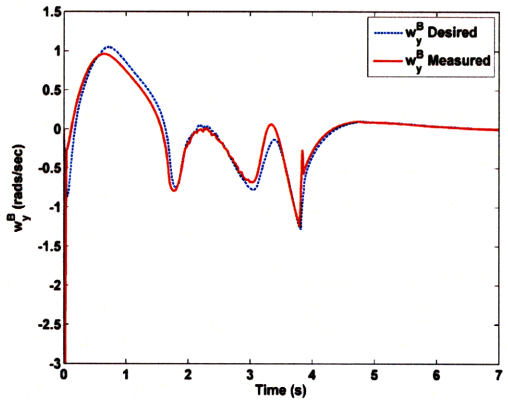
(c) Simulated v_x^B velocity output



(d) Simulated v_z^B velocity output



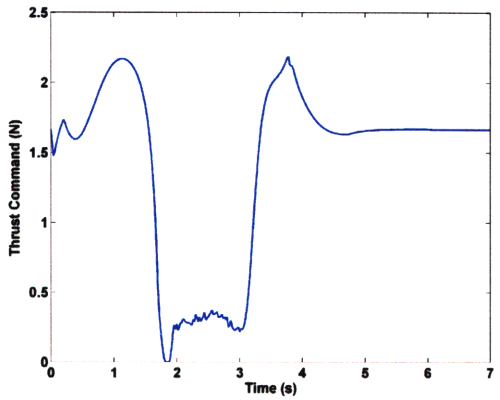
(e) Simulated pitch output



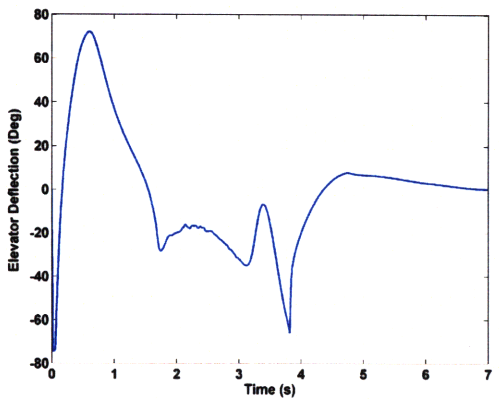
(f) Simulated w_y^B rotational rate output

Figure 4-7: Simulated Lyapunov quaternion controlled hover to hover state data

The pitch is similar to the simulation (Figure 4-9(e)), but the amount of control effort by the elevator (Figure 4-10(b)) and consequently the rotational rate (Figure 4-



(a) Simulated thrust command

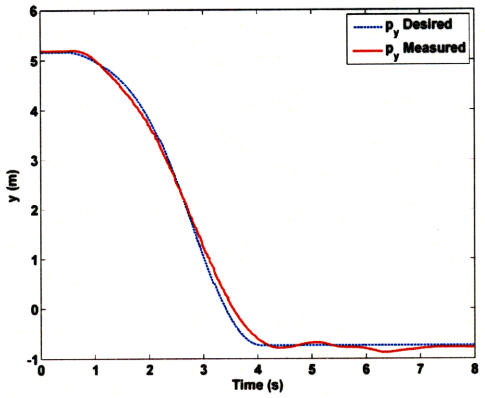


(b) Simulated elevator deflection control

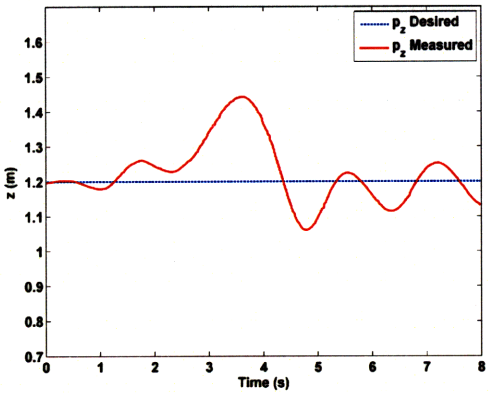
Figure 4-8: Simulated Lyapunov quaternion controlled hover to hover control effort

9(f)) differ. This is due to the difference in initial conditions and user-defined gain inputs. When tuning the gains for the desired rotational rate variable, the user is able to give a weighting factor to either error or modeled feed-forward output, similar to a linear quadratic estimator (LQE) design. This is also directly related to how anticipatory the desired rate will be. Since the trajectory is predefined, the desired control laws, if weighed heavily on the feed-forward term, will anticipate future movements and dampen the response. Therefore, the user is able to design the response to a given trajectory within the control limits. In this experiment, the measured error and the model based controls are weighed equally providing the authority necessary for tracking (Figure 4-10).

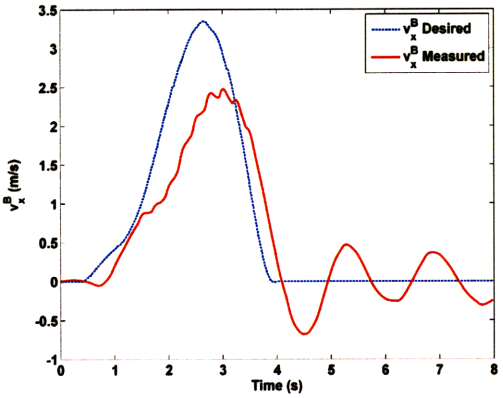
An additional experiment was done to determine the affects of weighing the models feed-forward control strategy for the elevator more heavily than the state error. As the model feed-forward control gains are increased, the system becomes more conservative yielding larger control deflections earlier in the maneuver which can be seen in Figure 4-12. As a result, the tracking about the y-position undershoots as seen in Figure 4-11. From this example, it is apparent that there is a trade-off in this control strategy between weighing the feed-forward controls calculated by the model and the error-based control signals.



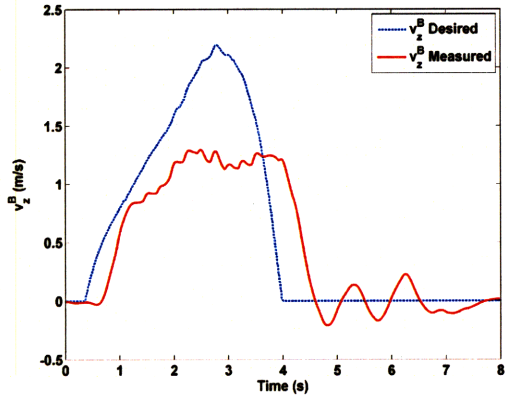
(a) y position outputs



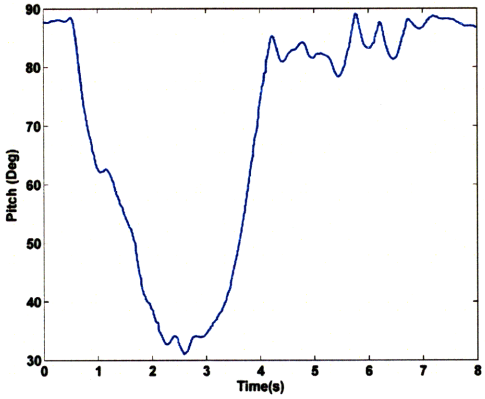
(b) z position outputs



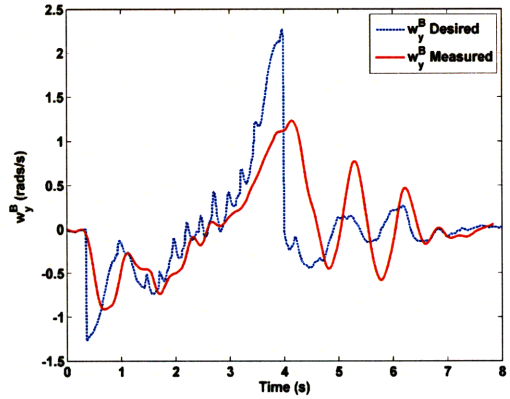
(c) v_x^B velocity outputs



(d) v_z^B velocity outputs

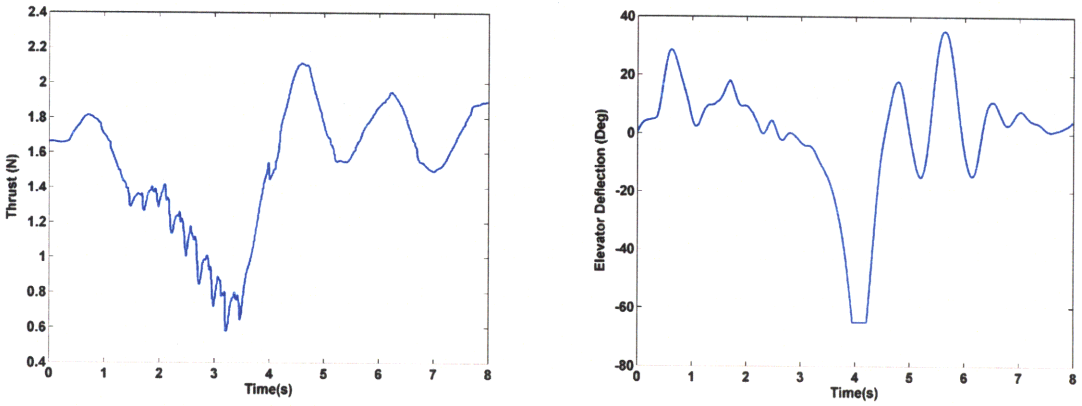


(e) Pitch output



(f) w_y^B rotational rate output

Figure 4-9: Hardware implemented Lyapunov quaternion controlled hover to hover state data



(a) Hardware implemented thrust command

(b) Hardware implemented elevator deflection control

Figure 4-10: Hardware Lyapunov quaternion controlled hover to hover control effort

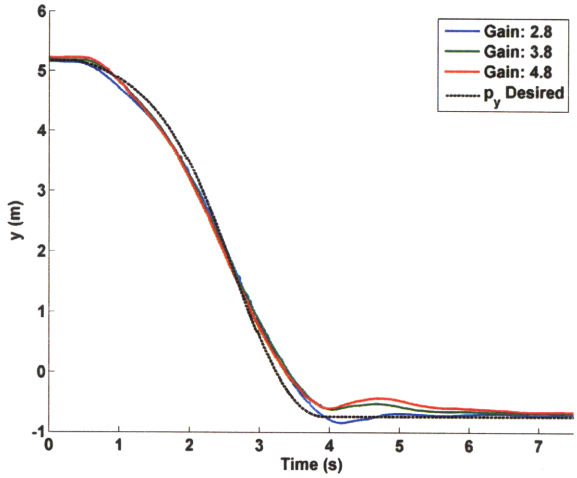


Figure 4-11: Measured y-position output for the hover to hover maneuver varying the feed-forward predictive gains

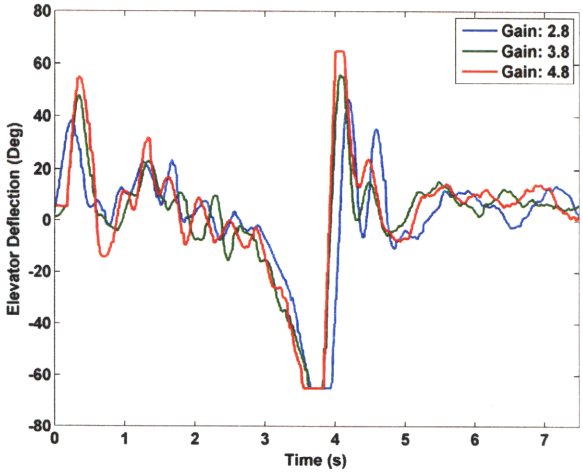


Figure 4-12: Elevator deflection output for the hover to hover maneuver varying the model feed-forward gains

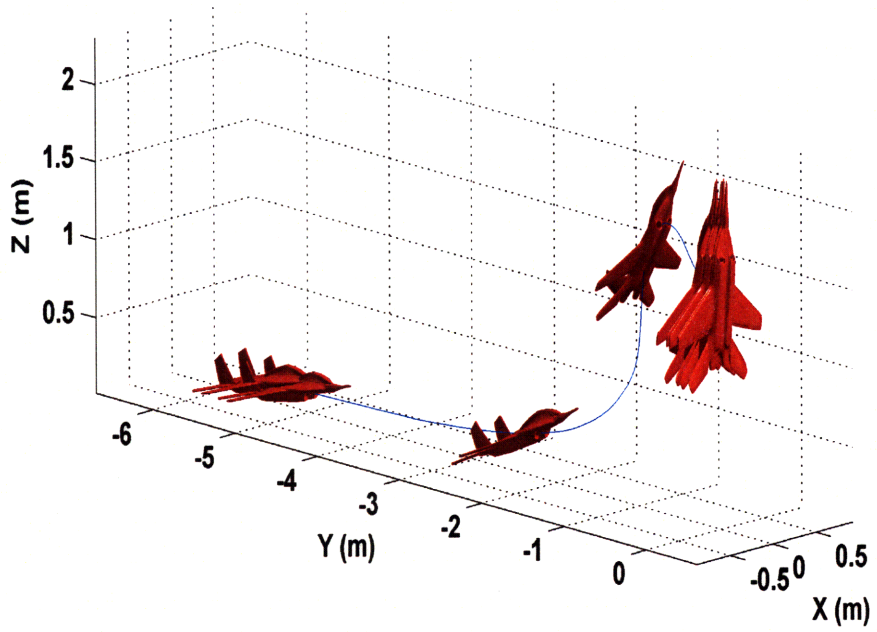
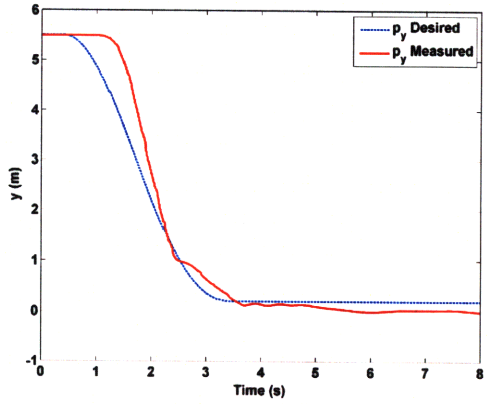


Figure 4-13: Hardware implemented Lyapunov quaternion controlled take-off to hover maneuver

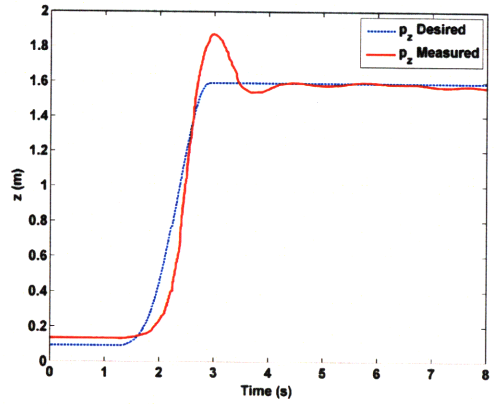
Take-off to Hover Maneuver

For the previous maneuver, the quaternion controller is used as insight to produce a feasible path. This type of trajectory insight is not available for a take-off to hover maneuver, so an educated *ad hoc* estimate was performed which involved manually directing the aircraft in the environment to develop a path, shown in Figure 4-14(a) and Figure 4-14(b). The same type of data fitting done in the previous section is done to provide the controller with a smooth, continuous path. The controller provides sufficient tracking given the uncertainty in the *ad hoc* trajectory planned. The elevator control inputs are intentionally made extra aggressive to ensure path following which can be seen in Figure 4-15(c) as the pitch angle dramatically decreases desperately attempting to track both the desired y and z-positions.

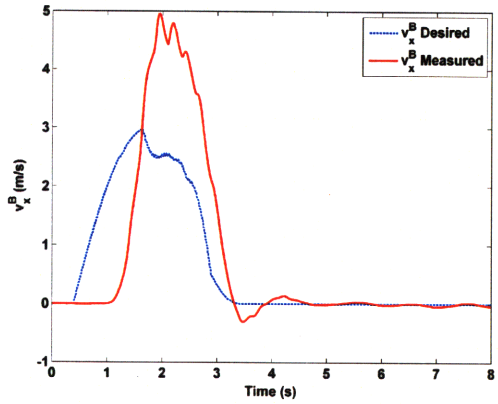
Since the aircraft is taking-off from the ground, the dynamics are considerably different than during flight because of drag from the floor and various other reasons including ground effect. These effects hinder the initial take-off portion of the maneuver as can been seen during the initial 2 seconds in Figure 4-14(a). The thrust term,



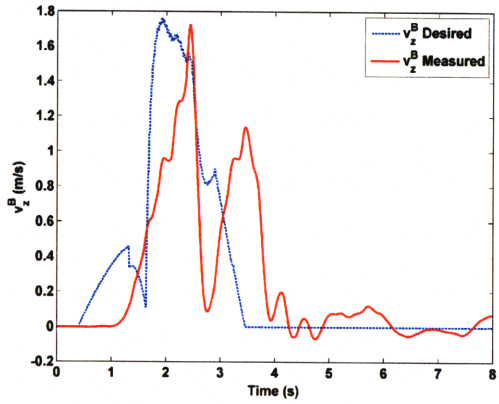
(a) y position outputs



(b) z position outputs



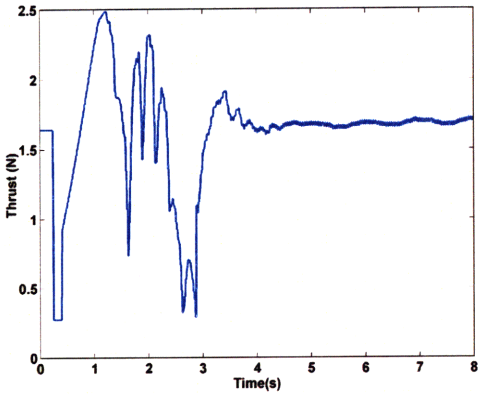
(c) v_x^B velocity outputs



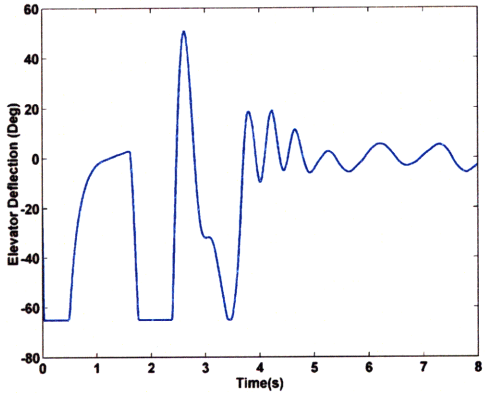
(d) v_z^B velocity outputs

Figure 4-14: Lyapunov quaternion controlled take-off to hover state data

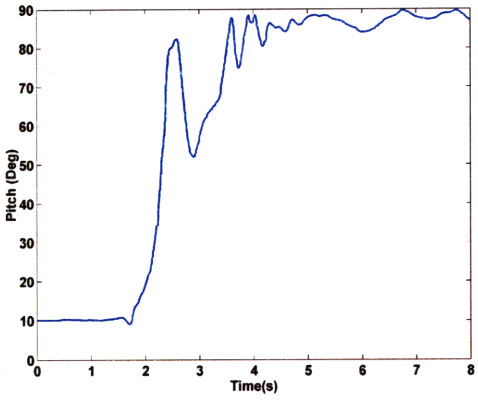
since anticipating aerial flight, only provides enough thrust to maintain the desired velocity and position for this instance in time. The error terms are building up, which can be seen in Figure 4-15(a) between 0.5-1.2 seconds as a linear increase in measured thrust output, and is providing the additional thrust necessary to compensate for the non-modeled effects. This lack of thrust can also be seen in Figure 4-14(c), by the measured velocity lagging behind the desired velocity then creating a massive over-shoot in attempt to decrease error. Given how arbitrary this trajectory was produced, the controller performs well showing great potential for high-bandwith capability.



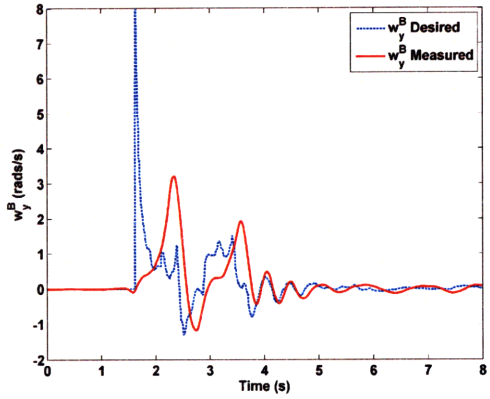
(a) Hardware implemented thrust command



(b) Hardware implemented elevator deflection control



(c) Pitch output



(d) w_y^B rotational rate output

Figure 4-15: Lyapunov quaternion controlled take-off to hover control effort

Chapter 5

Conclusion

This thesis presents multiple techniques for controlling a fixed-wing aircraft during aggressive, agile flight. Each controller offers its own unique approach to the reference command tracking problem. The quaternion-based attitude controller provides the ability to track a desired attitude sufficiently well using a linear control scheme. The Lyapunov controller uses derived equations of motion of the vehicle to accurately track feasible, sufficiently smooth, continuous position commands in a inertial-based frame. Each of these control designs are demonstrated, both in simulation and in actual hardware implementation using a small scale, indoor aerobatic fixed-wing aircraft.

Each chapter explored a specific component of the agile flight control problem which contribute to the overall goal of this thesis. In Chapter 2, a six-degree-of-freedom model of a fixed wing aircraft was derived and tested to ensure an accurate system identification. This was accomplished through the use of a load cell placed at the center of gravity of the aircraft which measured both forces and moments about the body fixed frame. Two main types of tests were performed to determine the aerodynamic properties of the vehicle, which included hover and multiple wind-tunnel tests. The hover tests were performed to determine the aircraft's response to propeller downwash, which solely provided the dynamic pressure used to control the aircraft. The wind tunnel tests were done at several different angles-of-attack, both with and without the propeller active. With these tests, the interaction of the propeller downwash and the free-stream velocity was determined as well as the

induced angle-of-attack in the region of the prop flow. The aircraft dynamics about the major lifting surface were split into two regions in order to differentiate this combined flow (or effective flow) from the pure free-stream flow. From these tests, a complete set of equations of motion were developed to form a high fidelity model. To ensure model accuracy, a system identification was done using the RAVEN motion capture system, which provided all the state data necessary for comparison. This model was then able to serve as the testbed for all simulation experiments and for the development of the nonlinear controller.

Chapter 3 presents a quaternion based attitude controller with velocity error based thrust correction. This controller has the ability to prescribe a desired attitude in the direction of decreasing error based on the error between a user-defined inertial velocity command and the actual measured velocity. The thrust command also uses the velocity command as a correction term for error minimization. The design was formulated to perform a desired transition from hover to level-flight mode. During this maneuver, the minimization of the loss of altitude as well as sufficient velocity tracking were priorities. The implementation of this controller was demonstrated on the testbed aircraft and the results confirmed desired performance.

Finally, Chapter 4 introduces a Lyapunov backstepping control technique that tracks reference input position commands. This design is highly model based and relies on the accuracy of the model presented in Chapter 2. Stability is proven under the Lyapunov criteria, and the linearization of the control law is provided to gain insight into the inner workings of the controller and to give aid to potential hazards of the design. Based on the research done in [13], a modified version of this controller has been created based on the lessons learned from the linearization technique, and combines the quaternion based control scheme outlined in Chapter 3. This controller is then implemented on actual hardware to obtain desired performance results.

In summary, the controllers developed in this thesis allow a fixed wing aircraft to perform some aggressive, agile flight maneuvers which show the potential of combining abilities of the desired flight regimes into one single design.

5.1 Future Work

During the course of this work, several key topics and issues have emerged which merit further research. These items are discussed below.

5.1.1 Improved Dynamic Model

The modeling done in this thesis, though extensive, was done primarily about the body y-axis focusing on a specific maneuver. The inclusion of a even higher fidelity model for the rest of the aircraft's axes would dramatically increase the capabilities of the control schemes mentioned. Given a three-dimensional, time-varying, parametric trajectory, the Lyapunov quaternion controller would be able to generate the necessary control commands for sufficient tracking. This additional modeling would include the motor dynamics with attached propeller and thus the effect they have on the thrust and moment in the aircraft's body x-direction.

5.1.2 Trajectory Linearized Control

Trajectory linearized control (TLC) is an advanced nonlinear control technique that linearizes the tracking error of a nonlinear system along a desired reference trajectory [28]. The TLC control configuration is shown in Figure 5-1.

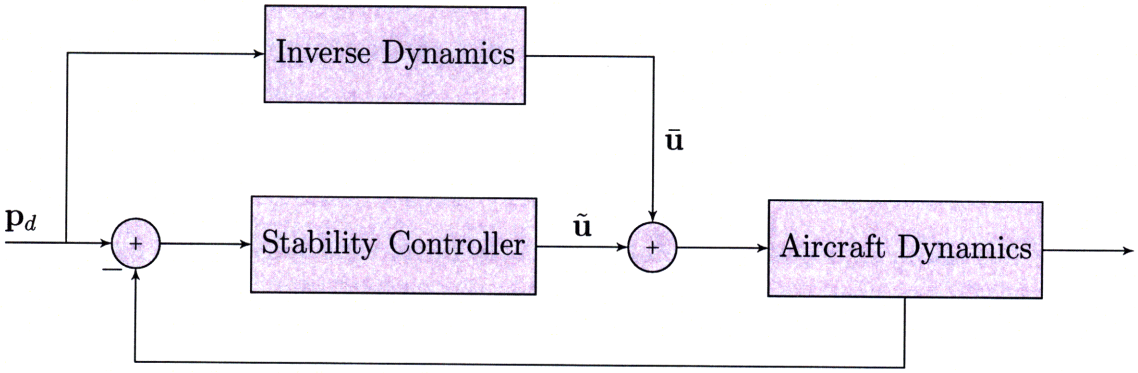


Figure 5-1: Trajectory linearized control architecture

This control technique is also a highly model-based control technique, but given an accurate model, would dramatically increase the tracking capability, thus increasing

the controllers bandwidth. The inverse dynamics are used to develop the desired (or nominal) control, $\bar{\mathbf{u}}$, necessary to perform the prescribed path. A linear time-varying, controller is then used to stabilize the nominal trajectory based on the state feedback. This design can be viewed as an ideal gain-scheduled controller at every point along the commanded trajectory and can be very robust given the accuracy and fidelity of the reference model. Computing the nominal control involves solving an optimal control problem. For a tracking problem, the definition of the cost function [29] is

$$\min_{u} J = \int_{t_o}^{t_f} \left[\left\| \mathbf{p}(t) - \mathbf{p_d}(t) \right\|^2_{\mathbf{Q}(t)} + \left\| \mathbf{u}(t) \right\|^2_{\mathbf{R}(t)} \right] dt \tag{5.1}$$

where $\mathbf{Q}$ and $\mathbf{R}$ represent symmetric $n \times n$ weighting matrices. The goal is to minimize the above cost function subject to the dynamics of the aircraft given as

$$s.t. \quad \dot{\mathbf{p}}^I = R\mathbf{v}^B \tag{5.2}$$

$$\dot{R} = RS(\mathbf{w}^B) \tag{5.3}$$

$$J\dot{\mathbf{w}}^B = -S(\mathbf{w}^B)J\mathbf{w}^B + \mathbf{f}_w + G_w\mathbf{u}_w \tag{5.4}$$

$$M\dot{\mathbf{v}}^B = -S(\mathbf{w}^B)M\mathbf{v}^B + \mathbf{f}_v + \mathbf{g}_v\delta_t \tag{5.5}$$

$$\mathbf{u}(t) \in \mathcal{U}. \tag{5.6}$$

These control commands can be computed either *a priori* on an off-board computation system, or during the flight, which would require quick computational power.

5.1.3 Path Feasibility Planner

One of the major issues when planning a desired path, especially within a constrained environment, is ensuring that the commanded path is feasible. The paths given in this thesis were defined using feedback from manually maneuvering the aircraft in an educated adhoc manner, or an actual flight path that was either flown manually or from another controller. With a path feasibility planner, the control techniques mentioned in this thesis would be able to track a path, eliminating this as one of the control variabilities.

Appendix A

Quaternion Based Method for the Determination of Body Rates using a Motion Capture System

A.1 Introduction

The Vicon motion capture system used for vehicle sensing provides position and orientation data about a local inertial reference frame. The reference frame is an East North Up (ENU) frame in which position is given in millimeters and orientation in a three vector axis-angle representation with an embedded angle. When performing aggressive, agile flight in a nonconventional manner, attitude representation without singularities is essential. Convenient and typical forms of attitude representation, such as the use of Euler angles, suffer from the loss of at least one or more degrees of freedom resulting in what is known as "gimbal lock". Since the sensor only provides orientation data, the extraction of rate data must be done by means of a derivative. Issues arise with the continuity of the orientation data due to the embedded angle information and precautionary steps are made to ensure smooth, continuous data extraction.

A.2 Extracting the Axis Angle

An axis angle representation parameterizes a single rotation of a rigid body through an angle θ and a reference axis,

$$(\text{angle}, \mathbf{axis}) = \left(\theta, \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \right) \tag{A.1}$$

commonly referred to as Euler axis or eigenaxis [25]. The three axis components and total rotational angle describe a vector and an object's pure rotation about this vector in a single fixed reference frame. The axis vector's magnitude produces an additional degree of freedom which is constrained to a unit magnitude given by

$$a_x^2 + a_y^2 + a_z^2 \equiv 1. \tag{A.2}$$

The motion capture system provides object orientation in the form (axis with embedded angle)

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \theta. \tag{A.3}$$

The angle θ can be extracted using Equations [A.2] and [A.3],

$$\begin{aligned} A_x^2 + A_y^2 + A_z^2 &= \left(a_x^2 + a_y^2 + a_z^2\right)\theta^2 = \theta^2 \\ \Rightarrow \theta &= \pm\sqrt{A_x^2 + A_y^2 + A_z^2} \end{aligned} \tag{A.4}$$

which can be substituted into Eq.[A.3] to find the axis vector components. Initially, the sign of θ can be arbitrarily chosen to be positive which leads to a bounded angle,

$$\theta \in [0, \pi]\,. \tag{A.5}$$

These axis angle components can then be transformed into their respective quaternion representations by

$$\begin{bmatrix} q_o \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ a_x \sin\left(\frac{\theta}{2}\right) \\ a_y \sin\left(\frac{\theta}{2}\right) \\ a_z \sin\left(\frac{\theta}{2}\right) \end{bmatrix}. \tag{A.6}$$

It is then possible to relate this quaternion and its respective component time rate of change to an object's rotational body rate, denoted by $\mathbf{w}$, using

$$\mathbf{w} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} 2(q_o\dot{q}_x + q_z\dot{q}_y - q_y\dot{q}_z - q_x\dot{q}_o) \\ 2(-q_z\dot{q}_x + q_o\dot{q}_y + q_x\dot{q}_z - q_y\dot{q}_o) \\ 2(q_y\dot{q}_x - q_x\dot{q}_y + q_o\dot{q}_z - q_z\dot{q}_o) \end{bmatrix}. \tag{A.7}$$

This is equivalent to the nominal aerospace nomenclature of p, q and r which are the rotational body rates. Cusps and discontinuities arise in the functions due to the ambiguous sign allocation of θ and is shown in Figure A-1. This figure shows the output

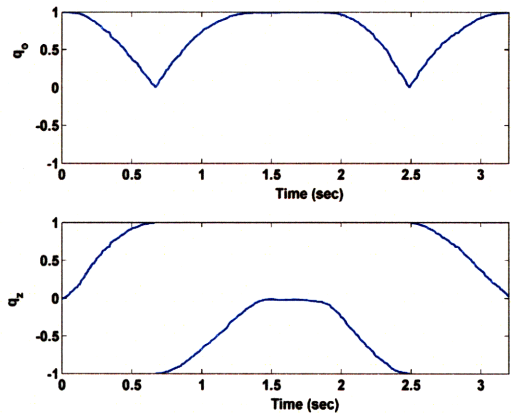


Figure A-1: Quaternion data due to a pure rotation about the reference z-axis.

quaternion of a pure rotation about the z-axis. Only the scalar and z components are shown because the rotation axis is strictly a unit z-axis rotation. The reason for these cusps and discontinuities can be explained by the following: since the rotation angle is chosen to be positive for all rotations, the axis must flip in order to agree with the

right-hand-rule, illustrated in Figure A-2. Taking the derivatives of these functions

(a) Positive rotation with axis pointing out of the page

(b) Positive rotation with axis pointing into the page

Figure A-2: Pure body z-rotation illustrating axis flip

leads to large errors at the point of the cusp or discontinuity. Therefore, a method for tracking when a flip occurs has been developed to maintain smooth continuous functions so that their derivatives are also continuous. This method uses the unit magnitude property of Equation A.2 by taking the dot product of an update axis angle $(k+1)$ with the previous update (k). This point can be shown by evaluating the dot product of an axis angle at each side of the discontinuity or cusp using the pure z rotation example as follows:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_{k+1} \approx \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \& \quad \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_{k} \approx \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{A.8}$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_{k} \cdot \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_{k+1} \approx -1 \tag{A.9}$$

The dot product of the these axis angle terms yields a -1 which is in violation of Equation A.2. Thus, the conclusion that a flip has occurred can be realized and the negative sign can be placed on the angle to maintain $\mathbf{q}$ as a continuous function of θ.

Table A.1: Smooth quaternion signal data algorithm

Algorithm A.1	
1:	**Initialize:** sign = 1.0
Loop	
2:	**if** $(\mathbf{a}_k \cdot \mathbf{a}_{k-1} < 0.0)$
	sign = -1.0·sign
3:	$q_o = \text{sign} \cdot \cos\left(\frac{\theta}{2}\right)$
	$q_x = \text{sign} \cdot a_x \sin\left(\frac{\theta}{2}\right)$
	$q_y = \text{sign} \cdot a_y \sin\left(\frac{\theta}{2}\right)$
	$q_z = \text{sign} \cdot a_z \sin\left(\frac{\theta}{2}\right)$

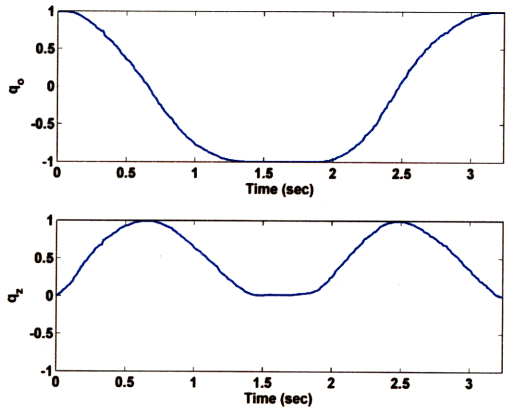


Figure A-3: Continuous quaternion data due to a pure rotation about the reference z-axis.

With this algorithm implemented (Table A.1), the output in Figure A-1 from before becomes a smooth time varying function, now shownn in Figure A-3. This data can now be differentiated and the quaternion rates are calculated. Then, by using Equation A.7, the body rates can be found.

Bibliography

[1] SPG Media. Airforce-technology.com. http://www.airforce-technology.com/projects/predator/, 2009. 13, 15

[2] AeroVironment, Inc. AeroVironment Unmanned Aircraft Systems. Available at http://www.avinc.com/uas/, 2009. 13

[3] Boeing. Boeing Integrated Defense Systems - V-22 Osprey. Available at http://www.boeing.com/rotorcraft/military/v22/index.htm, 2009. 14

[4] W. F. Chana and J. F. Coleman. World's first VTOL airplane Convair/Navy XFY-1 Pogo. In *Proceedings of the International Powered Lift Conference*, pages 99–104, Jupiter, FL, November 1996. 17

[5] W.E. Green and P.Y. Oh. A MAV That Flies Like an Airplane and Hovers Like a Helicopter. In *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey, CA, July 2005. 17

[6] E. N. Johnson, M. A. Turbe, A. D. Wu, S. K. Kannan, and J. C. Neidhoefer. Flight Test Results of Autonomous Fixed-Wing UAV Transitions to and from Stationary Hover. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, August 2006. 17

[7] A. Frank, J. McGrew, M. Valenti, D. Levine, and J. How. Hover, Transition, and Level Flight Control Design for a Single-Propeller Indoor Airplane. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Hilton Head, SC, August 2007. 17, 26, 35

[8] G. Meyer and R. Su and L. R. Hunt. Application of nonlinear transformations to automatic flight control. *Automatica*, 1(20), 1984. 18

[9] S. H. Lane and R. F. Stengle. Flight control using nonlinear inverse dynamics. *Automatica*, 24:471–483, 1988. 18

[10] S. N. Singh and M. Steinberg. Adaptive Control of Feedback Linearizable Nonlinear Systems with Application to Flight Control. *Journal of Guidance, Control, and Dynamics*, 19(4):871–877, 1996. 18

[11] D. Angeli and E. Mosca. Lyapunov-Based Switching Supervisory Control of Nonlinear Uncertain Systems. *IEEE Transactions on Automatic Control*, 47(3):500–505, 2002. 18

[12] Z.P. Jiang. Global Tracking Control of Underactuated Ships by Lyapunovs Direct Method. *Automatica*, 38:301–309, 2002. 18

[13] A. Pedro Aguiar and J. P. Hespanha. Trajectory-Tracking and Path-Following of Underactuated Autonomous Vehicles With Parametric Modeling Uncertainty. *IEEE Transactions on Automatic Control*, 52(8):1362–1379, August 2007. 18, 19, 53, 54, 67, 82

[14] S. Lee, H. Lee, D. Won and H. Bang. Backstepping Approach of Trajectory Tracking Control for the Mid-Altitude Unmanned Airship. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 20-23 August 2007. 18

[15] N. Knoebel. Adaptive Quaternion Control of a Miniature Tailsitter UAV. Master's thesis, Brigham Young University, 2007. 18, 19, 42, 67

[16] B.L. Stevens and F.L. Lewis. *Aircraft Control and Simulation.* John Wiley & Sons, Inc., 1992. 22, 31

[17] J.D. Anderson, Jr. *Fundamentals of Aerodynamics.* McGraw-Hill, 2001. 22, 27, 34

[18] RC Factory. High Performance Indoor Aerobatic Models. Available at http://www.rc-factory.eu/, 2008. 23

[19] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian. Real-Time Indoor Autonomous Vehicle Test Environment. *IEEE Control Systems Magazine*, April 2008. 24

[20] W.F. Phillips. *Mechanics of Flight.* John Wiley & Sons, Hoboken, NJ, 2004. 26

[21] R.C. Nelson. *Flight Stability and Automatic Control.* McGraw-Hill, 1998. 28

[22] J.L. Tangler and J.D. Kocurek. Wind Turbine Post-Stall Airfoil Performance Characteristics Guidelines for Blade-Element Momentum Methods. In *Proceedings of the 43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2005. 33

[23] A.J. Hanson. *Visualizing Quaternions.* Morgan Kaufmann Publishers, San Francisco, CA, 2006. 45

[24] J. E. Slotine and W. Li. *Applied Nonlinear Control.* Prentice Hall, 1991. 66

[25] W.F. Phillips, C.E. Hailey and G.A. Gebert. Review of Attitude Representations Used for Aircraft Kinematics. *AIAA Journal of Aircraft*, 38(4):718–737, July-August 2001. 68, 86

[26] I.Y. Bar-Itzhack. An Algorithm For Computing the Quaternion from the Rotation Matrix. *AIAA/AAS Astrodynamics Specialist Conference*, August 2000. 68

[27] G. Strang. *Computational Science and Engineering.* Wellesley-Cambridge Press, Wellesley, MA, 2007. 71

[28] T. Bevacqua, E. Best, A. Huizenga, D. Cooper, and J. Zhu. Improved Trajectory Linearization Flight Controller for Reusable Launch Vehicles. In *Proceedings of the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2004. 83

[29] D.E. Kirk. *Optimal Control Theory: An Introduction.* Dover Publications, Inc., Mineola, NY, 1998. 84