



All Theses and Dissertations

2007-08-30

Adaptive Quaternion Control for a Miniature Tailsitter UAV

Nathan B. Knoebel

Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>

 Part of the [Mechanical Engineering Commons](#)

BYU ScholarsArchive Citation

Knoebel, Nathan B., "Adaptive Quaternion Control for a Miniature Tailsitter UAV" (2007). *All Theses and Dissertations*. 1193.
<https://scholarsarchive.byu.edu/etd/1193>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

ADAPTIVE QUATERNION CONTROL OF A MINIATURE
TAILSITTER UAV

by

Nathan B. Knoebel

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

Brigham Young University

December 2007

Copyright © 2007 Nathan B. Knoebel

All Rights Reserved

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Nathan B. Knoebel

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date

Timothy W. McLain, Chair

Date

Randal W. Beard

Date

Mark B. Colton

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Nathan B. Knoebel in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Timothy W. McLain
Chair, Graduate Committee

Accepted for the Department

Matthew R. Jones
Graduate Coordinator

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of
Engineering and Technology

ABSTRACT

ADAPTIVE QUATERNION CONTROL OF A MINIATURE TAILSITTER UAV

Nathan B. Knoebel

Department of Mechanical Engineering

Master of Science

The miniature tailsitter is a unique aircraft with inherent advantages over typical unmanned aerial vehicles. With the capabilities of both hover and level flight, these small, portable systems can produce efficient maneuvers for enhanced surveillance and autonomy with little threat to surroundings and the system itself. Such vehicles are accompanied with control challenges due to the two different flight regimes. Problems with the conventional attitude representation arise in estimation and control as the system departs from level flight conditions. Furthermore, changing dynamics and limitations in modeling and sensing give rise to significant attitude control design challenges. Restrictions in computation also result from the limited size and weight capacity of the miniature airframe.

In this research, the inherent control challenges discussed above are addressed with a computationally efficient adaptive quaternion control algorithm. A backstepping method for model cancellation and consistent tracking of reference model attitude dynamics is derived. This is used in conjunction with two different algorithms

designed for the identification of system parameters. For a metric of baseline performance, gain-scheduled quaternion feedback control is developed. With a regularized data-weighting recursive least-squares parameter estimation algorithm, the adaptive quaternion controller is shown to be better than the baseline method in simulation and hardware results. This method is also shown to produce universal performance for all aircraft with the three conventional control surface actuators (aileron, elevator, and rudder) barring saturation and assuming accurate system identification.

Testing of attitude control algorithms requires development in quaternion-based navigational control and attitude estimation. A novel technique for hover north/east position control is derived. Also, altitude tracking in hover, given an inconsistent thrust system, is addressed with an original method of on-line throttle system identification. Means for quaternion-based level flight control are produced from adaptations made to existing techniques employed in the Brigham Young University Multi-Agent Coordination and Control Lab. Also generated are simple trajectories for transitions between flight modes. A method for the estimation of quaternion attitude is developed, which uses multiple sensors combined in a filtering technique similar to the fixed-gain Kalman filter. Simulation and hardware results of these methods are presented for concept validation. A discussion of the development and production of these testing means (a simulation environment and hardware flight test system) is provided. In culmination, a fully autonomous miniature tailsitter system is produced with results demonstrating its various capabilities.

ACKNOWLEDGMENTS

I would like to acknowledge the help of all those who have assisted me in my thesis work. The members of my committee, Dr. McLain, Dr. Beard, and Dr. Coltan, have all given me excellent guidance and advice throughout the project. I would also like to recognize Dr. Snyder a former member of my committee, who derived the propeller wash forces and torques model which was used in the research. Many members of the BYU MAGICC Lab assisted in the development of source code and the acquisition of data during static and in-flight testing. Foremost of these, Steve Osborne gave significant assistance in both flight experiments and the development of the tailsitter autopilot state machine. Other contributors include P. B. Sujit, Jeff Hogge, Andrew Eldredge, Neil Johnson, James Hall, Blake Barber, Brandon Call, and Travis Millet. Furthermore, all MAGICC Lab members have contributed to the wonderful atmosphere of learning and development, which I have appreciated immensely. I would like to thank my wife Amanda specifically. She has supported me in my research and helped considerably in the editing of my thesis.

Table of Contents

Acknowledgements	xiii
List of Tables	xix
List of Figures	xxiv
1 Introduction	1
1.1 Motivation	1
1.2 Background	4
1.3 Contributions	5
1.4 Document Organization	8
2 UAV Testbed	9
2.1 Simulation Environment	9
2.1.1 Propeller Forces and Torques	11
2.1.2 Propeller Wash Forces and Torques	12
2.1.3 Level Flight Forces and Torques	16
2.1.4 Equations of Motion	18
2.1.5 Graphical User Interface	19
2.2 Flight Test Setup	21
2.2.1 Airframe	21
2.2.2 Autopilot	24

2.2.3	Ground Station	25
2.3	Summary	26
3	Attitude Control	27
3.1	Gain-Scheduled Quaternion Feedback Attitude Control	27
3.2	Adaptive Attitude Control	30
3.2.1	Aircraft Attitude Dynamics	30
3.2.2	Reference Model Dynamics	33
3.2.3	Error Dynamics	34
3.2.4	Backstepping Attitude Controller Derivation	35
3.2.5	Parameter Estimation	38
3.3	Simulation Results	45
3.3.1	Gain-Scheduled Quaternion Feedback Attitude Control	47
3.3.2	Stabilized Recursive Least Squares With Data Forgetting	50
3.3.3	Lyapunov Method	56
3.4	Hardware Results	63
3.4.1	Gain-Scheduled Quaternion Feedback Attitude Control	65
3.4.2	Stabilized Recursive Least Squares With Data Forgetting	68
3.5	Summary	73
4	Navigational Control	75
4.1	Hover Position Control	75
4.1.1	Controller Derivation	75
4.1.2	Simulation Results	78
4.1.3	Hardware Results	81
4.2	Hover Altitude Control	83
4.2.1	Controller Derivation	83

4.2.2	Simulation Results	88
4.2.3	Hardware Results	92
4.3	Level Flight Navigational Control	95
4.3.1	Controller Derivation	95
4.3.2	Simulation Results	97
4.3.3	Hardware Results	101
4.4	Transitions	104
4.4.1	Controller Derivation	104
4.4.2	Simulation Results	105
4.4.3	Hardware Results	107
4.5	Summary	107
5	Attitude Estimation	109
5.1	Model Update	109
5.2	Measurement Update	110
5.2.1	Gravity Measurement	110
5.2.2	Bearing Measurement	112
5.3	Results	115
5.4	Summary	118
6	Conclusions and Recommendations	119
6.1	Conclusions	119
6.2	Recommendations	120
	Appendix A Quaternion Math	123
A.1	Quaternion Definition	123
A.2	Quaternion Derivative	124

A.3 Quaternion Composition	124
A.4 Quaternion Error	125
Bibliography	130

List of Tables

2.1	Aircraft physical parameters	10
2.2	Level flight coefficients	17

List of Figures

1.1	Brigham Young University Tailsitter in Hover Flight	2
1.2	Tailsitter Autopilot Block Diagram	4
2.1	Sketch of Tailsitter Geometry and Nomenclature	10
2.2	Graphical User Interface of Tailsitter Simulation Environment	20
2.3	Components of the Propulsion System	22
2.4	The Tailsitter Airframe Built for Flight Testing	23
2.5	Kestrel Autopilot Used in Flight Testing	24
2.6	The Ublox GPS Module and Sarentel Omnidirectional Antenna Used in Flight Testing	25
2.7	The Three-Axis PNI Magnetometer Used in Flight Testing	25
2.8	The Ground Station Employed in Flight Testing	26
3.1	Block Diagram of the Baseline Attitude Controller	29
3.2	Block Diagram of the Model Reference Adaptive Quaternion Controller	30
3.3	Contour Plot of Data Weighting as a Function of λ and $N - k$	40
3.4	Example Waypoint Path Selected for the Attitude Control Simulation Experiment	46
3.5	Attitude Results of the Simulated Baseline Attitude Controller	48
3.6	Error of the Simulated Baseline Attitude Controller	49
3.7	Attitude Results of the Simulated Adaptive Backstepping Controller with Least-Squares Parameter Estimation	51

3.8	Error of the Simulated Adaptive Backstepping Controller with Least-Squares Parameter Estimation	52
3.9	Angular Rates of the Simulated Adaptive Backstepping Controller with Least-Squares Parameter Estimation	53
3.10	Parameter Estimation of the Simulated Adaptive Backstepping Controller with Least-Squares Parameter Estimation	54
3.11	Pitch Acceleration of the Simulated Adaptive Backstepping Controller with Least-Squares Parameter Estimation	55
3.12	Attitude of the Simulated Adaptive Backstepping Controller with Lyapunov-Based Parameter Estimation	57
3.13	Attitude Error of the Simulated Adaptive Backstepping Controller with Lyapunov-Based Parameter Estimation	58
3.14	Angular Rates of the Simulated Adaptive Backstepping Controller with Lyapunov-Based Parameter Estimation	59
3.15	Parameter Estimation of the Simulated Adaptive Backstepping Controller with Lyapunov-Based Parameter Estimation	60
3.16	Parameters Estimated Directly with Lyapunov-Based Estimation During Attitude Experiment	61
3.17	Pitch Acceleration of the Simulated Adaptive Backstepping Controller with Lyapunov-Based Parameter Estimation	62
3.18	Example Waypoint Path Selected for the Attitude Control Hardware Experiment	64
3.19	Attitude Results of the Baseline Attitude Controller in Hardware	66
3.20	Attitude Error of the Baseline Attitude Controller in Hardware	67
3.21	Attitude Results of the Adaptive Backstepping Controller with Least-Squares Parameter Estimation in Hardware	69
3.22	Error of the Adaptive Backstepping Controller with Least-Squares Parameter Estimation in Hardware	70
3.23	Angular Rates of the Adaptive Backstepping Controller with Least-Squares Parameter Estimation in Hardware	71
3.24	Parameter Estimation of the Adaptive Backstepping Controller with Least-Squares Parameter Estimation in Hardware	72

3.25	Pitch Acceleration of the Adaptive Backstepping Controller with Least-Squares Parameter Estimation in Hardware	73
4.1	Diagram Showing the Vertical Quaternion Orientation	76
4.2	Position Tracking of the Simulated Hover Position Controller	79
4.3	Attitude Results of the Simulated Hover Position Controller	80
4.4	Position Tracking of the Hover Position Controller in Hardware	81
4.5	Attitude Results of the Hover Position Controller in Hardware	82
4.6	Propeller Angular Rate Relationship with Throttle and Voltage	84
4.7	Propeller Exit Airspeed Relationship with Propeller Angular Rate	85
4.8	Block Diagram of the Hover Altitude Controller	87
4.9	Altitude Results of the Simulated Hover Altitude Controller	89
4.10	Thrust Results of the Simulated Hover Altitude Controller	90
4.11	Thrust System Parameters of the Simulated Hover Altitude Controller	91
4.12	Altitude Results of Hover Altitude Controller in Hardware	92
4.13	Thrust Results of Hover Altitude Controller in Hardware	93
4.14	Thrust System Parameters of Hover Altitude Controller in Hardware	94
4.15	Vector Field Sliding Surface Example	95
4.16	Level Flight Altitude and Airspeed Controller logic	96
4.17	Level Flight Controller Path in Simulation Test	98
4.18	Course and Heading Results of the Level Flight Controller in Simulation	99
4.19	Altitude and Airspeed Results of the Level Flight Controller in Simulation	100
4.20	Level Flight Controller Path in Hardware Test	101
4.21	Course and Heading Results of the Level Flight Controller in Hardware	102
4.22	Altitude and Airspeed Results of the Level Flight Controller in Hardware	103
4.23	Attitude During Simulated Transitions	106

4.24	Attitude During Transitions in Hardware	108
5.1	Bearing Error Induced by Error in Roll with Magnetometers	113
5.2	Attitude in Simulated Attitude Estimation Experiment	116
5.3	Attitude Error in Simulated Attitude Estimation Experiment	117

Chapter 1

Introduction

The Vertical takeoff and land (VTOL) miniature tailsitter unmanned aerial vehicle (UAV) has many advantages over traditional aircraft. These advantages, however, come with control challenges unique to the miniature tailsitter. Motivation for research and autopilot development in this area is given. A discussion of past work performed to address similar challenges and contributions made during this research is also offered. Organization of the document is then presented.

1.1 Motivation

VTOL UAVs have inherent advantages due to hover capabilities. Such vehicles can fly in confined areas and effectively takeoff and land in designated regions without a runway. These capabilities greatly enlarge the autonomy of the UAV, limiting the need for human interaction in recovery and deployment, and also allowing for perch-and-stare maneuvers, in-flight persistent target imaging, and navigation of obstacle filled terrain. All of these tasks are difficult, if not infeasible, for typical fix-wing UAVs to accomplish.

A largely unexplored UAV concept is the miniature tailsitter (Figure 1.1). Tailsitter UAVs are fix-wing VTOL aircraft, thus having all of the advantages of hover flight, as well as the benefits of efficient fix-wing flight, which is significant for miniature UAVs due to energy limitations. Moreover, the miniature tailsitter, because of size, is more portable and can navigate difficult terrain effectively and with little threat to immediate surroundings and the vehicle itself. This UAV concept, however, with its many advantages, poses control challenges.



Figure 1.1: The Brigham Young University tailsitter is shown in hover flight.

Conventionally the aeronautics community uses the $3 - 2 - 1$ body-referenced Euler angles (ψ , θ , and ϕ) to relate the inertial and body reference frames of an aircraft. This representation works well for attitude estimation and control if pitch (θ) and roll (ϕ) angles are relatively small. From investigation it can be seen that the described set of Euler angles contains a singularity in the kinematic equations when θ approaches positive or negative $\pi/2$ known as gimbal lock (ψ and ϕ are not unique). These equations are integrated over time for attitude estimation, resulting in a divide by zero in the described conditions. Moreover, elevator control based upon error in θ and rudder control based upon error in ψ degrades as ϕ and θ become larger than $\pi/4$. Hence it is clear that a tailsitter, which is intended to fly both vertically and horizontally, is incompatible with the conventional attitude representation.

Noting these limitations, the quaternion attitude representation for control and estimation is attractive. In the quaternion control formulation, attitude error can be represented conveniently in the aircraft body reference frame about principal x , y , and z axes, which correspond to available actuator input torques, produced by deflections in the aircraft's ailerons, elevators, and rudders. Also, the quaternion attitude representation lacks singularities, which are inherent with any set of Euler angles chosen to represent attitude.

Another control challenge is the nonlinear dynamics of the tailsitter that are difficult to model. Traditional autopilot design for attitude control involves the linearizing of dynamic equations about an operating condition. A controller is then

typically derived from linear control methodology. This technique requires that a good model of the aircraft's dynamics be available, all relevant states be measured or estimated, and the aircraft remains in the design flight conditions. Due to the nature of miniature tailsitter UAVs, flight conditions can change drastically through typical maneuvers. Furthermore, aerodynamics in some flight regimes (particularly in stall conditions) are difficult to model and states such as aircraft angle of attack and sideslip angle are infeasible for miniature UAVs to measure or estimate. Thus, traditional methods for attitude control design are not viable for miniature VTOL tailsitters.

Also, small hover aircraft require lightweight hardware. This constraint extends to the autopilot processor. Currently, small autopilot hardware for miniature UAVs is available. However, computation on such small systems is limited. Thus, the tailsitter autopilot algorithms developed must be computationally efficient for hardware implementation.

To address these attitude control challenges, investigation and development of several computationally efficient model reference adaptive quaternion control methods were performed. This research will show that through adaptive attitude control, a system is able to effectively identify plant dynamics and apply appropriate control for stable attitude tracking and good performance, despite drastic changes in dynamics and limitations in state estimation and aircraft modeling. The algorithm used to display such behavior surprisingly requires little computation.

Gain-scheduled quaternion feedback proportional, integral, and derivative (PID) control has also been designed as a baseline method to be used as a metric for determining the effectiveness of adaptive attitude control. Due to the challenges that have been described, gains for this controller were tuned empirically from flight test observations and performance analysis. In this method, two sets of gains were scheduled based upon propeller wash velocity and aircraft airspeed.

Miniature VTOL tailsitter attitude control is the focus of this research, however, development in other major areas of the tailsitter autopilot were necessary for flight testing of attitude control. As shown in Figure 1.2, the basic structure

of the quaternion autopilot has three main elements. To test attitude control, the development of quaternion attitude estimation, as well as quaternion and throttle navigational control for hover, level flight, and transitions were required. Consequently, VTOL autopilot advancement was accomplished in the necessary estimation and navigational control areas.

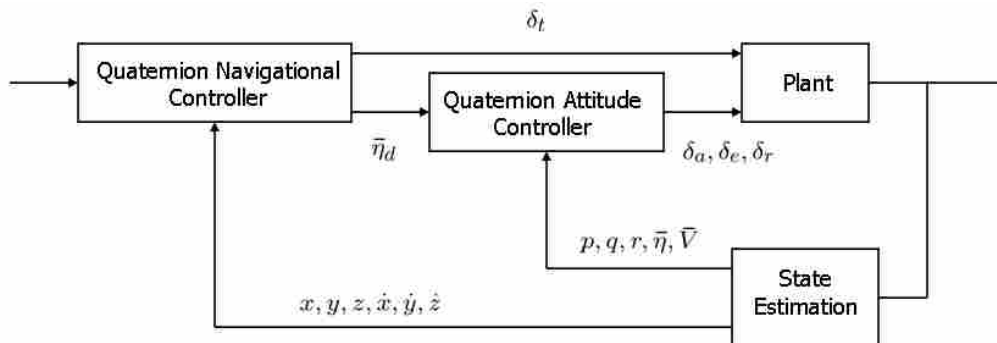


Figure 1.2: The tailsitter autopilot is shown in a block diagram.

1.2 Background

Despite being a relatively unexplored topic, VTOL tailsitter UAVs have received considerable attention in recent research. Wernicke [1] surveys a variety of VTOL designs and shows the efficiency of the tailsitter concept. Ailon [2] demonstrates state-to-state motion control and stable trajectory tracking with a VTOL UAV. Martin, et al. [3] devise a state tracking system for a VTOL aircraft utilizing the concept of differential flatness. Much work has been done on the design, control, and hardware implementation of the T-Wing tailsitter UAV by Stone and Clarke [4, 5, 6, 7]. Finally, Taylor, et al. [8] have published a preliminary design process and lessons learned for a military autonomous VTOL system.

The quaternion attitude control structure has also been explored by numerous researchers [4, 9, 10, 11], although, mainly for spacecraft. Despite this, considering the

capabilities of the tailsitter, quaternion feedback control is quite applicable because such aircraft can effectively produce torques about all three body frame axes.

For flight during aggressive maneuvers and in poorly modeled conditions, a variety of different adaptive control techniques have been proposed, including neural networks, least squares estimation, and Lyapunov based methods. The neural network approach typically entails training a network off-line for model inversion. An on-line adaptive network is then employed to compensate for modeling errors [12, 13, 14, 15, 16]. Recursive least squares techniques identify the airframe parameters on-line and use these parameters to adjust the controller. Such controllers have the ability to quickly converge on the airframe parameters [17], and therefore, adapt rapidly to changes in flight conditions. Other approaches to aircraft autopilot design involving least squares estimation include Refs. [15, 18, 19]. In Lyapunov based approaches, the parameter update law is selected to ensure stability of the tracking error. These methods generally cannot guarantee parameter convergence [20, 21, 22, 23, 24, 25]. The \mathcal{L}_1 adaptive controller is a unique Lyapunov-based method that has also been used in flight control of UAVs [26]. The method differs in that a unique companion model is tracked. The companion model is fundamentally similar to a reference model, however, specific changes are made to enable the filtering of system parameters. As a result, fast, yet stable adaptation can be achieved [27, 28]. This method has been shown to be effective for attitude control, but requires exact knowledge of how the inputs affect the system. Structured adaptive model inversion (SAMI) has also been shown to be effective at Lyapunov based flight control. The SAMI controller, with the understanding that aircraft kinematics are known exactly, only adapts unknown momentum level parameters using Lyapunov theory and then employs principles of dynamic model inversion for control [29, 30].

1.3 Contributions

Throughout this research project many contributions have been made, much of which is original work. These contributions include:

- **Adaptive quaternion backstepping control** A novel adaptive quaternion backstepping controller was developed for model cancellation and stable tracking of reference model attitude dynamics. The method was derived to achieve consistent performance for any given set of identified system parameters. As an additional feature, this results in a universal controller for all three-actuator (aileron, elevator, and rudder) airframes.
- **Least-squares system parameter identification** Two computationally efficient methods for estimating system parameters were developed, namely least-squares and Lyapunov-based methods. The regularized least-squares algorithm with data forgetting was shown to be able to accurately learn system parameters despite drastic changes in dynamics and poor sensing capabilities.
- **Baseline gain-scheduled quaternion feedback controller** A baseline quaternion feedback controller with scheduled gains was designed for a metric of attitude control performance. This method also displayed effectiveness at attitude tracking amidst the various control challenges.
- **Hover north/east quaternion-based position controller** Unique quaternion-based position control for north/east tracking in hover flight was derived. This, with altitude and attitude control, enabled the tracking of hover waypoints in autonomous flight.
- **Adaptive hover altitude controller with on-line parameter estimation** A novel method for altitude control with thrust system identification was designed. The technique was shown to be able to track altitude regardless of an inconsistent thrust system with on-line learning.
- **Level flight quaternion-based path following and altitude/airspeed controllers** Quaternion-based level flight controllers for path following and altitude/airspeed tracking were developed based upon methods previously used at Brigham Young University.

- **Simple trajectories for transitions between level and hover flight** Simple quaternion-based trajectories were produced to allow the system to transition between hover and level flight. These methods, although not designed for optimality, were successful at navigating the tailsitter through transitions in both simulation and hardware.
- **Quaternion attitude estimation with improved sensors** A scheme similar to the fixed-gain Kalman filter was developed for the estimation of quaternion attitude. For an improved estimate of bearing in hover, three-axis magnetometers were integrated. Also, a 4 Hz GPS with an omnidirectional antenna was incorporated for faster updates and satellite tracking despite large changes in orientation.
- **Tailsitter simulation** To expedite the progress of the tailsitter autopilot, a graphical simulation environment was developed. This allowed for algorithm testing and assessment with little time commitment. It also gave insightful information of algorithm performance.
- **Tailsitter hardware system** For full verification of the techniques derived, a tailsitter hardware system was produced. An airframe was built and integrated with an autopilot system. Existing autopilot and ground station source code was modified to execute the tailsitter algorithms.
- **Simulation and Hardware Results** Results from both simulation and hardware testing of all techniques derived in this research were obtained for concept verification.

In culmination of the project, a functional autonomous miniature VTOL tailsitter UAV system was produced with hardware results demonstrating its various capabilities.

1.4 Document Organization

In this document a discussion of developments made throughout this research will be given. The experimental testbed including both the simulation environment and the hardware flight test platform and autopilot system is presented first in Chapter 2. Basic modeling and the graphical user interface developed for simulation of the tailsitter are discussed. Also described are the airframe design and components of both the thrust and autopilot systems employed for autonomous tailsitter flight. In Chapter 3, attitude control is discussed. A baseline gain-scheduled quaternion feedback control method is put forth first. Derivation of a backstepping model reference adaptive quaternion controller is then given, followed by two methods of on-line parameter estimation. Simulation and hardware results of the attitude controllers are discussed and compared at the end of the chapter. Chapter 4 contains a discussion of developments made in navigational control. Hover flight position and altitude methods are presented first. Level flight navigational control is then discussed for path following and altitude and airspeed tracking. Simple trajectories for transitions between flight modes are then put forth. Simulation and hardware results of all of the navigational techniques derived are given in each individual section for concept verification. A scheme for quaternion attitude estimation in hover and level flight is discussed in Chapter 5. Results are presented for validation. Conclusions and recommendations, found in Chapter 6, are given as a final synopsis of the research performed and possible future work.

Chapter 2

UAV Testbed

For verification of the algorithms developed for miniature VTOL tailsitter control, a testbed has been developed. The technique employed involved both simulation and hardware flight testing. Initially simulation was utilized to find what algorithms were most likely to succeed. The algorithms were then tested on the autopilot hardware in flight. This procedure allowed for rapid algorithm development. Ideas could be studied and assessed easily in simulation with little time commitment before full hardware verification.

2.1 Simulation Environment

The simulation environment used in the experiment was created for quick implementation and facilitation of understanding in algorithm development and assessment. The modeling technique used was not necessarily intended for accuracy, but to adequately portray the behavior of the algorithms being tested.

The basic simulation method of integrating the standard quaternion based non-linear translational and rotational kinematic and dynamic equations for a six degree-of-freedom rigid-body aircraft was applied. Forces and torques used by the equations of motion were generated from simple models of the motor/propeller system, level flight aerodynamics, and propeller wash aerodynamics.

Physical aircraft parameters used in simulation can be seen in Figure 2.1. The actual values used in calculations with brief descriptions are shown in Table 2.1. Note that sensors used in the simulated algorithms are also emulated in the described simulation environment. Appropriate update rates, noise, and biases typical of the sensors were applied.

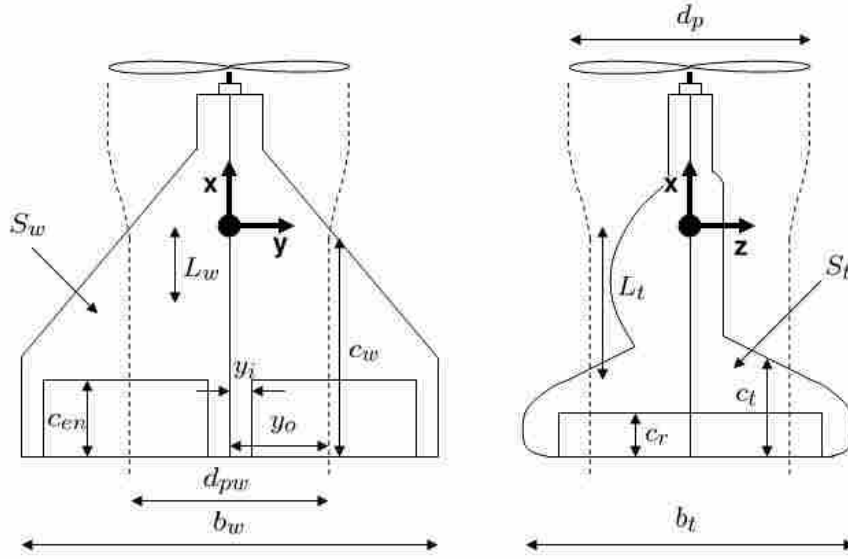


Figure 2.1: Tailsitter geometry is shown with nomenclature.

Table 2.1: Table of physical parameters used in simulation

Variable	Value	Description
S_w	0.259	Wing area, m^2
S_t	0.151	Tail area, m^2
c_w	0.380	Wing cord, m
c_t	0.225	Tail cord, m
c_{en}	0.100	Elevon cord, m
c_r	0.0700	Rudder cord, m
b_w	0.580	Wing span, m
b_t	0.580	Tail span, m
L_w	0.0380	x distance from center of gravity to wing aerodynamic center, m
L_t	0.181	x distance from center of gravity to wing aerodynamic center, m
y_o	0.140	y distance from center of gravity to outer edge of elevon, m
y_i	0.0250	y distance from center of gravity to inner edge of elevon, m
d_p	0.280	Propeller diameter, m
d_{pw}	0.252	Propeller wash diameter, m
g	9.81	Gravity, m/s^2
μ	0.746	Mass, kg
ρ	1.27	Air density, kg/m^3
J_{xx}	0.115	x -axis moment of inertia, $kg \cdot m^2$
J_{yy}	0.0776	y -axis moment of inertia, $kg \cdot m^2$
J_{zz}	0.171	z -axis moment of inertia, $kg \cdot m^2$
J_{xz}	0.00150	cross-product of inertia, $kg \cdot m^4$

2.1.1 Propeller Forces and Torques

Modeling of the propulsion system will be discussed in this section. Assumptions about the system are made based upon propulsion test results performed on the ground (static tests). These assumptions are then used to produce a simple model of propeller wash airflow, and ultimately of thrust and propeller induced torque.

From static testing (see Chapter 4), it has been found that there is a linear relationship between propeller angular rate (ω_p) and both the percent throttle setting (δ_t) and battery voltage (E):

$$\omega_p = C_{\omega_p 0} + C_{\omega_p, E} E + C_{\omega_p, \delta_t} \delta_t. \quad (2.1)$$

Thrust system parameter values used in simulation were similar to those found in hardware testing, namely $C_{\omega_p 0} = -356$, $C_{\omega_p, E} = 46.6$, and $C_{\omega_p, \delta_t} = 7.28$.

The flow of air exiting the propeller has also been found to have a linear relationship with propeller angular rate in similar propulsion tests (see Chapter 4), given as

$$V_p = C_{V_p, \omega_p} \omega_p, \quad (2.2)$$

where $C_{V_p, \omega_p} = 0.02310$, as found from static testing. For simplification we assume that these two relationships remain consistent in all flight conditions. Under these assumptions the propeller wash airspeed is defined as the difference between the propeller and aircraft airspeed:

$$V_{pw} = V_p - V. \quad (2.3)$$

Using momentum theory and noting that V and V_p are the entrance and exit velocities, the thrust is found to be

$$F_{x-p} = \frac{1}{2} \rho A_p (V_p^2 - V^2). \quad (2.4)$$

Propeller induced torque is modeled simply with the non-dimensional torque coefficient

$$l_p = C_{l,p} \rho \left(\frac{\omega_p}{2\pi} \right)^2 d_p^5, \quad (2.5)$$

with $C_{l,p} = 0.000390$.

2.1.2 Propeller Wash Forces and Torques

For propeller wash forces and torques, a simplified aerodynamic model was derived. This model, briefly discussed in this section, is based on the following assumptions:

- Aerodynamic propeller wash lift/drag forces due to the body are negligible.
- Aerodynamic propeller wash control forces are generated only by deflections of control surfaces located within the propeller wash region.
- Flow in the propeller wash region remains parallel to the body x -axis.
- The only significant aerodynamic propeller wash forces/moments are lifting surface normal forces (which may induce moments) and lifting surface moments.

Propeller wash forces (F_{x-pw} , F_{y-pw} , and F_{z-pw}) and moments (l_{pw} , m_{pw} , and n_{pw}) in the body reference frame are functions of propeller wash velocity (V_{pw}) and the angular deflection of the aileron (δ_a), elevator (δ_e), and rudder (δ_r) control surfaces:

$$F_{x-pw} = 0.0,$$

$$F_{y-pw} = -F_r(V_{pw}, \delta_r),$$

$$F_{z-pw} = -F_e(V_{pw}, \delta_e),$$

$$l_{pw} = M_a(V_{pw}, \delta_a),$$

$$m_{pw} = -F_e(V_{pw}, \delta_e)L_{en} + M_e(V_{pw}, \delta_e), \text{ and}$$

$$n_{pw} = F_r(V_{pw}, \delta_r)L_r + M_r(V_{pw}, \delta_r).$$

Note that the actuator sign convention was chosen, such that positive deflections of ailerons, elevators, and rudders resulted in positive torques about the x -axis, y -axis, and z -axis respectively.

Aileron

Although the actual control surfaces used to produce pitching and rolling moments will be elevons, the aerodynamic forces can be separated into aileron and elevator contributions. The ailerons produce a moment about the x -axis only. A simple method for estimating the rolling moment induced by the ailerons, called strip theory, multiplies the local section lift increment due to the aileron deflection by the local moment arm and integrates over the wing.

The change in lift coefficient of the wing for a two dimension (2D) wing section as a function of aileron deflection is

$$(\Delta\tilde{C}_{Lw})_{\delta_a} = \pm\tilde{C}_{Lw,\alpha}\tilde{\epsilon}_{en}\delta_a, \quad (2.6)$$

where $\tilde{}$ is used to denote 2D section properties, ϵ_{en} the elevon efficiency, and α the angle of attack. For a symmetric deflection of the ailerons, the moment coefficient about the x -axis is given by strip theory as

$$C_{\ell,\delta_a} = -\frac{2\tilde{C}_{Lw,\alpha}}{S_w b_w} \int_{y_i}^{y_o} y \tilde{\epsilon}_{en} c_w dy. \quad (2.7)$$

This can be simplified even further by assuming that the wing chord length (c_w) is constant with span, as is the aileron size. This yields

$$C_{\ell,\delta_a} = -\frac{\tilde{C}_{Lw,\alpha}}{S_w b_w} \tilde{\epsilon}_{en} c_w (y_o^2 - y_i^2). \quad (2.8)$$

Finally, assuming no rolling moment is produced at zero aileron deflection, the moment produced by an aileron deflection is

$$M_a = \frac{1}{2}\rho V_{pw}^2 \left(\delta_a \tilde{C}_{Lw,\alpha} \tilde{\epsilon}_{en} c_w (y_o^2 - y_i^2) \right), \quad (2.9)$$

where y_i is the body y -coordinate of the inner most edge of the elevon, and y_o the outer most edge of the elevon or the coordinate of the outer most edge of the slipstream. The lift slope for the 2D wing sections can be approximated as

$$\tilde{C}_{Lw,\alpha} \approx 2\pi, \quad (2.10)$$

and from thin airfoil theory (see for instance Ref. [31]) the elevon efficiency is

$$\tilde{\epsilon}_{en} \approx 1 - \frac{\sigma_{en} - \sin(\sigma_{en})}{\pi} \quad (2.11)$$

with

$$\sigma_{en} = \cos^{-1} \left(2 \frac{c_{en}}{c_w} - 1 \right). \quad (2.12)$$

Elevator

The elevators produce a force in the body z -direction and a corresponding moment about the body y -axis. Again using strip theory, we integrate the section change in lift coefficient over the spanwise extent of the control surface

$$C_{z,\delta_e} = \frac{2\tilde{C}_{Lw,\alpha}}{S_w} \int_{y_i}^{y_o} \tilde{\epsilon}_{en} c_w dy. \quad (2.13)$$

Assuming symmetry about the y -axis and that the wing chord and elevon geometry are constant along the span, yields

$$C_{z,\delta_e} = \frac{2\tilde{C}_{Lw,\alpha}}{S_w} \tilde{\epsilon}_{en} c_w (y_o - y_i) \quad (2.14)$$

or

$$F_e = \frac{1}{2} \rho V_{pw}^2 \left(\delta_e \tilde{C}_{Lw,\alpha} \tilde{\epsilon}_{en} \right) c_w (y_o - y_i) \quad (2.15)$$

with the wing lift slope and elevon efficiency defined in equations (2.10) and (2.11), respectively.

Using a very similar analysis, the elevator pitching moment about the lifting surface quarter chord is

$$M_e = \frac{1}{2}\rho V_{pw}^2 \tilde{C}_{m,\delta_e} \delta_e c_w (y_o - y_i) \quad (2.16)$$

where, from thin airfoil theory, we can approximate the moment coefficient as

$$\tilde{C}_{m,\delta_e} = \frac{\sin(2\sigma_{en}) - 2\sin(\sigma_{en})}{4} \quad (2.17)$$

with σ_{en} defined in equation (2.12).

Rudder

The rudders produces a force in the body y -direction and a corresponding moment about the body z -axis. Using an approach similar to that employed for the elevator and referring to the sketch in Figure 2.1, we find the force produced by the rudders is

$$F_r = \frac{1}{2}\rho V_{pw}^2 \left(\delta_r \tilde{C}_{L_t,\alpha} \epsilon_r \right) c_t d_s. \quad (2.18)$$

The section lift slope is again approximated as $C_{L_t,\alpha} \approx 2\pi$ and the rudder efficiency is

$$\tilde{\epsilon}_r \approx 1 - \frac{\sigma_r - \sin(\sigma_r)}{\pi} \quad (2.19)$$

with

$$\sigma_r = \cos^{-1} \left(2 \frac{c_r}{c_t} - 1 \right). \quad (2.20)$$

Finally, the moment produced about the lifting surface quarter-chord due to rudder deflection is

$$M_r = \frac{1}{2}\rho V_{pw}^2 \tilde{C}_{m,\delta_r} \delta_r d_s c_t, \quad (2.21)$$

where

$$\tilde{C}_{m,\delta_r} = \frac{\sin(2\sigma_r) - 2\sin(\sigma_r)}{4}. \quad (2.22)$$

2.1.3 Level Flight Forces and Torques

Level flight forces and torques are produced from typical aerodynamic modeling methods. Aerodynamic coefficients are utilized to describe approximate linear relationships of system inputs and states to forces and torques. These coefficients include lift (C_L), drag (C_D), lateral force (C_Y), rolling moment (C_l), pitching moment (C_m), and yawing moment (C_n). The inputs into the system involved with aerodynamic forces and torques modeling are actuator angular deflections from trim (δ_a , δ_e , and δ_r), while the states include airspeed (V), angle of attack (α), sideslip angle (β), and angular rates (p , q , and r) about principle body reference frame axes. Actual coefficients used in simulation, as well as brief descriptions, are shown in Table 2.2.

Given the aerodynamic coefficients described earlier, the forces can be modeled as

$$\begin{aligned}
 F_{x-lev} &= \frac{1}{2}\rho V^2 S_w \left(-\cos \alpha C_{D0} - \sin \alpha C_{L0} - \cos \alpha C_{D,\alpha} |\alpha| - \sin \alpha C_{L,\alpha} \alpha \right. \\
 &\quad \left. - \cos \alpha C_{D,q} \frac{c_w}{V} |q| - \sin \alpha C_{L,q} \frac{c_w}{V} q - \cos \alpha C_{D,\delta_e} |\delta_e| - \sin \alpha C_{L,\delta_e} \delta_e \right), \\
 F_{y-lev} &= \frac{1}{2}\rho V^2 S_w \left(C_{Y0} + C_{Y,\beta} \beta + C_{Y,p} \frac{b_w}{2V} p + C_{Y,r} \frac{b_w}{2V} r + C_{Y,\delta_a} \delta_a + C_{Y,\delta_r} \delta_r \right), \text{ and} \\
 F_{z-lev} &= \frac{1}{2}\rho V^2 S_w \left(-\sin \alpha C_{D0} - \cos \alpha C_{L0} - \sin \alpha C_{D,\alpha} |\alpha| - \cos \alpha C_{L,\alpha} \alpha \right. \\
 &\quad \left. - \sin \alpha C_{D,q} \frac{c_w}{V} |q| - \cos \alpha C_{L,q} \frac{c_w}{V} q - \sin \alpha C_{D,\delta_e} |\delta_e| - \cos \alpha C_{L,\delta_e} \delta_e \right), \quad (2.23)
 \end{aligned}$$

and the torques similarly as

$$\begin{aligned}
 l_{lev} &= \frac{1}{2}\rho V^2 S_w \frac{b_w}{2} \left(C_{l0} + C_{l,\beta} \beta + C_{l,p} \frac{b_w}{2V} p + C_{l,r} \frac{b_w}{2V} r + C_{l,\delta_a} \delta_a + C_{l,\delta_r} \delta_r \right), \\
 m_{lev} &= \frac{1}{2}\rho V^2 S_w c_w \left(C_{m0} + C_{m,\alpha} \alpha + C_{m,q} \frac{c_w}{V} q + C_{m,\delta_e} \delta_e \right), \text{ and} \\
 n_{lev} &= \frac{1}{2}\rho V^2 S_w \frac{b_w}{2} \left(C_{n0} + C_{n,\beta} \beta + C_{n,p} \frac{b_w}{2V} p + C_{n,r} \frac{b_w}{2V} r + C_{n,\delta_a} \delta_a + C_{n,\delta_r} \delta_r \right). \quad (2.24)
 \end{aligned}$$

Table 2.2: Table of level flight coefficients

Variable	Value	Description
C_{L0}	0.00	Nominal lift coefficient
$C_{L,\alpha}$	3.45	Lift coefficient with respect to angle of attack
$C_{L,q}$	0.00	Lift coefficient with respect to pitch rate
C_{L,δ_e}	-0.360	Lift coefficient with respect to elevator deflection
C_{D0}	0.0300	Nominal drag coefficient
$C_{D,\alpha}$	0.300	Drag coefficient with respect to angle of attack
$C_{D,q}$	0.00	Drag coefficient with respect to pitch rate
C_{D,δ_e}	0.00	Drag coefficient with respect to elevator deflection
C_{Y0}	0.00	Nominal lateral force coefficient
$C_{Y,\beta}$	-0.980	Lateral force coefficient with respect to sideslip angle
$C_{Y,p}$	0.00	Lateral force coefficient with respect to roll rate
$C_{Y,r}$	0.00	Lateral force coefficient with respect to yaw rate
C_{Y,δ_a}	0.00	Lateral force coefficient with respect to aileron deflection
C_{Y,δ_r}	0.170	Lateral force coefficient with respect to rudder deflection
C_{l0}	-0.100	Nominal roll moment coefficient
$C_{l,\beta}$	0.0200	Roll moment coefficient with respect to sideslip angle
$C_{l,p}$	-0.200	Roll moment coefficient with respect to roll rate
$C_{l,r}$	0.00	Roll moment coefficient with respect to yaw rate
C_{l,δ_a}	4.40	Roll moment coefficient with respect to aileron deflection
C_{l,δ_r}	0.00	Roll moment coefficient with respect to rudder deflection
C_{m0}	0.0150	Nominal pitch moment coefficient
$C_{m,\alpha}$	-0.380	Pitch moment coefficient with respect to angle of attack
$C_{m,q}$	-3.60	Pitch moment coefficient with respect to pitch rate
C_{m,δ_e}	3.40	Pitch moment coefficient with respect to elevator deflection
C_{n0}	0.0100	Nominal yaw moment coefficient
$C_{n,\beta}$	0.250	Yaw moment coefficient with respect to sideslip angle
$C_{n,p}$	0.0220	Yaw moment coefficient with respect to roll rate
$C_{n,r}$	-0.350	Yaw moment coefficient with respect to yaw rate
C_{n,δ_a}	0.00	Yaw moment coefficient with respect to aileron deflection
C_{n,δ_r}	3.60	Yaw moment coefficient with respect to rudder deflection

This method is a reasonable approximation in non-stall conditions, and is therefore used only if the angle of attack is reasonable:

$$-\alpha_{stall} \leq \alpha \leq \alpha_{stall}. \quad (2.25)$$

If this condition does fail (indicating stall) only drag is applied. The force model can then be written as

$$\begin{aligned}
F_{x-lev} &= \frac{1}{2}\rho V^2 S_w \left(-\cos \alpha C_{D0} - \cos \alpha C_{D,\alpha} |\alpha| \right), \\
F_{y-lev} &= \frac{1}{2}\rho V^2 S_w \left(C_{Y0} + C_{Y,\beta} \beta \right), \text{ and} \\
F_{z-lev} &= \frac{1}{2}\rho V^2 S_w \left(-\sin \alpha C_{D0} - \sin \alpha C_{D,\alpha} |\alpha| \right),
\end{aligned} \tag{2.26}$$

and the torque model is simply

$$\begin{aligned}
l_{lev} &= 0.0, \\
m_{lev} &= 0.0, \text{ and} \\
n_{lev} &= 0.0.
\end{aligned} \tag{2.27}$$

2.1.4 Equations of Motion

The tailsitter simulation relies on the time integration of equations of motion in order to propagate aircraft states. The conventional quaternion-based 13-state model is applied. These states include body reference frame velocities (u , v , and w) and angular rates (p , q , and r). Also included are the quaternion attitude elements (η_1 , η_2 , η_3 , and η_4) and the inertial position (x , y , and z).

Using Newton's second law, aircraft linear and angular dynamics can be derived as

$$\begin{aligned}
\dot{u} &= rv - qw + g(2\eta_1\eta_3 - 2\eta_4\eta_2) + \frac{F_x}{\mu}, \\
\dot{v} &= -ru + pw + g(2\eta_2\eta_3 + 2\eta_4\eta_1) + \frac{F_y}{\mu}, \text{ and} \\
\dot{w} &= qu - pv + g(\eta_4\eta_4 - \eta_1\eta_1 - \eta_2\eta_2 + \eta_3\eta_3) + \frac{F_z}{\mu},
\end{aligned} \tag{2.28}$$

and

$$\begin{aligned}
\dot{p} &= \frac{J_{xz}(J_{xx} - J_{yy} + J_{zz})}{J_{xx}J_{zz} - J_{xz}^2}pq - \frac{J_{zz}(J_{zz} - J_{yy}) + J_{xz}^2}{J_{xx}J_{zz} - J_{xz}^2}qr \\
&\quad + \frac{J_{zz}}{J_{xx}J_{zz} - J_{xz}^2}l + \frac{J_{xz}}{J_{xx}J_{zz} - J_{xz}^2}n, \\
\dot{q} &= \frac{J_{zz} - J_{xx}}{J_{yy}}pr - \frac{J_{xz}}{J_{yy}}(p^2 - r^2) + \frac{1}{J_{yy}}m, \text{ and} \\
\dot{r} &= \frac{J_{zz}(J_{zz} - J_{yy}) + J_{xz}^2}{J_{xx}J_{zz} - J_{xz}^2}pq - \frac{J_{xz}(J_{xx} - J_{yy} + J_{zz})}{J_{xx}J_{zz} - J_{xz}^2}qr \\
&\quad + \frac{J_{xz}}{J_{xx}J_{zz} - J_{xz}^2}l + \frac{J_{zz}}{J_{xx}J_{zz} - J_{xz}^2}n.
\end{aligned} \tag{2.29}$$

Observe that F_x , F_y , and F_z represent the resulting sum of forces in the associated body frame directions and l , m , and n are the resulting torques about the associated axes. The aircraft attitude kinematics are described by the quaternion derivative (see Appendix A.2)

$$\begin{aligned}
\dot{\eta}_1 &= \frac{1}{2}(\eta_4p - \eta_3q + \eta_2r), \\
\dot{\eta}_2 &= \frac{1}{2}(\eta_3p + \eta_4q - \eta_1r), \\
\dot{\eta}_3 &= \frac{1}{2}(-\eta_2p + \eta_1q + \eta_4r), \text{ and} \\
\dot{\eta}_4 &= \frac{1}{2}(-\eta_1p - \eta_2q - \eta_3r),
\end{aligned} \tag{2.30}$$

while the inertial position time derivative is given simply from rotating the body-frame linear velocities into the inertial reference frame:

$$\begin{aligned}
\dot{x} &= u(\eta_4\eta_4 + \eta_1\eta_1 - \eta_2\eta_2 - \eta_3\eta_3) + v(2\eta_2\eta_1 - 2\eta_4\eta_3) + w(2\eta_3\eta_1 + 2\eta_4\eta_2), \\
\dot{y} &= u(2\eta_1\eta_2 + 2\eta_4\eta_3) + v(\eta_4\eta_4 - \eta_1\eta_1 + \eta_2\eta_2 - \eta_3\eta_3) + w(2\eta_3\eta_2 - 2\eta_4\eta_1), \text{ and} \\
\dot{z} &= u(2\eta_1\eta_3 - 2\eta_4\eta_2) + v(2\eta_2\eta_3 + 2\eta_4\eta_1) + w(\eta_4\eta_4 - \eta_1\eta_1 - \eta_2\eta_2 + \eta_3\eta_3).
\end{aligned} \tag{2.31}$$

2.1.5 Graphical User Interface

To enhance user interaction with the simulation environment, a graphical user interface (GUI) was developed. This provided to the user visual feedback of the

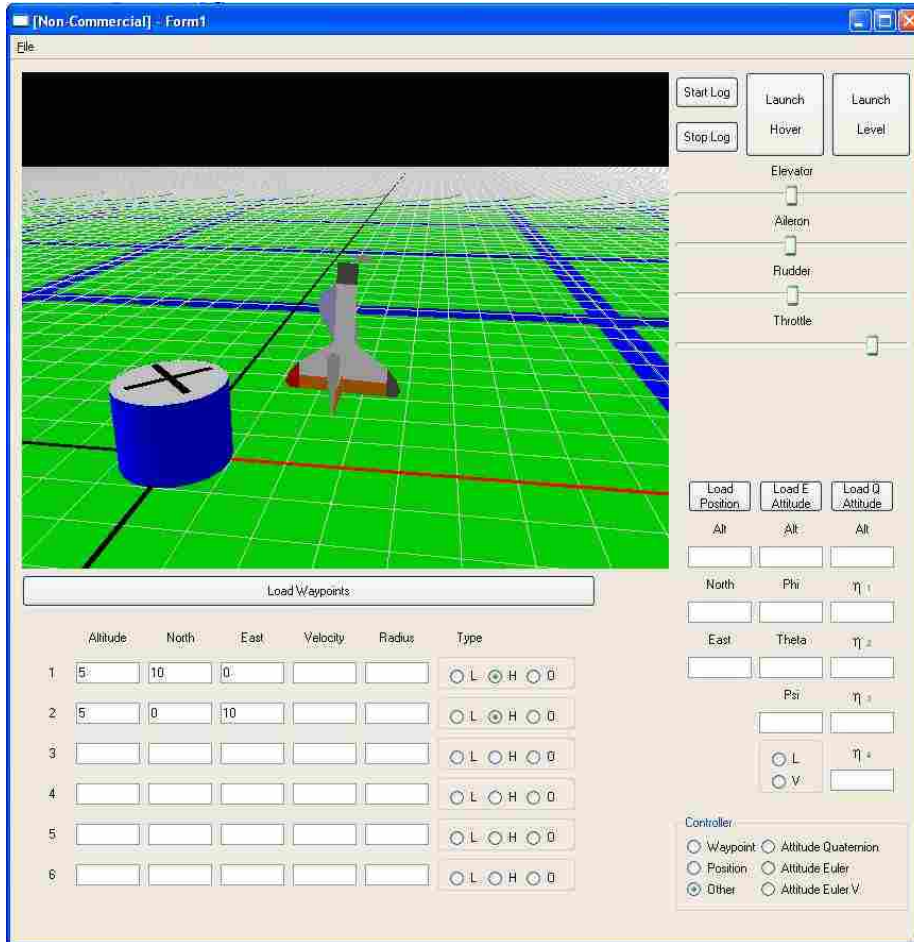


Figure 2.2: The graphical user interface of the simulation environment is displayed above.

simulated aircraft in flight and allowed for effortless input into the simulation. A screenshot of the GUI is provided in Figure 2.2. The GUI allowed the user to load commands, select controllers, and start and stop data logging. OpenGL was used for graphics rendering and the user interface was designed with QT version 3.2.1 noncommercial.

2.2 Flight Test Setup

The flight test setup included all of the hardware and software required to implement the tailsitter autopilot algorithms that have been developed. Major components of this system were an airframe, autopilot system, and ground station. Development of these three elements was necessary to perform the experiment, however, autopilot algorithms remained the focus of the research.

2.2.1 Airframe

To perform tailsitter flight tests, an airframe platform was produced. Even though the research goal was not to develop an airframe, an adequate platform was required. Consequently, a simple design of known performance was selected. This design was then improved upon for robustness and altered to carry the autopilot and necessary sensors.

The airframe design chosen was the commercially available RC tailsitter model kit known as the Pogo [32]. The straightforward design is manufactured from two pieces of flat depron foam simply cut into a main wing and tail section. This simple design proved to be advantageous in minimizing construction time, as several vehicles were required to accomplish all of the flight testing for the research. To improve durability, which is a nontrivial quality for flight testing, corrugated plastic was substituted for the foam. This, along with all of the extra hardware needed onboard, significantly increased the weight of the vehicle. Other material, such as fiberglass plate, expanded polypropylene (EPP) foam, velcro, epoxy, cyanoacrylate glue, and fiber tape were employed in the manufacturing of the tailsitter.

To accommodate the weight increase, a powerful yet efficient thrust system was necessary to achieve hover flight. An efficient, high-torque brushless outrunner motor (Hacker A20-20L [33]) with a 10×4.7 propeller, a programmable 25 amp electronic speed control (Castle Creations Phoenix-25 [34]), and a light weight, high discharge 3-cell 2000 mAh lithium polymer battery (Thunder Power Pro Lite [35]) were found in flight testing to be a superior and adequate propulsion system. Components of the propulsion system are shown in Figure 2.3.

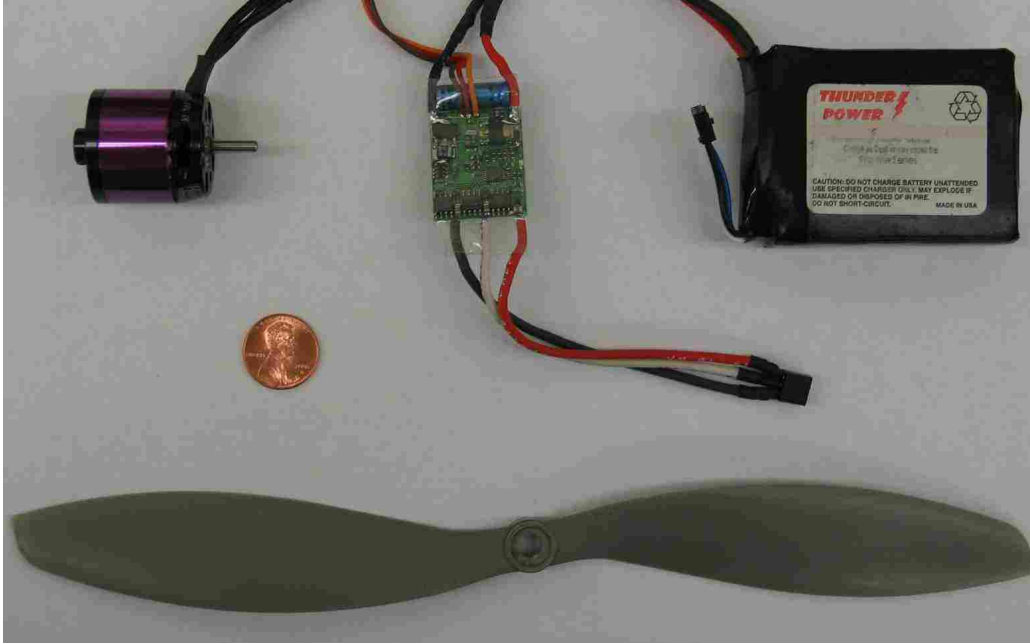


Figure 2.3: Components of the propulsion system are exhibited above: Hacker A20-20L brushless outrunner motor (top left), Castle Creations Phoenix-25 electronic speed control (top center), Thunder Power Pro Lite lithium polymer battery (top right), 10 × 4.7 propeller (bottom).

Lightweight construction, a powerful propulsion system, and large control surfaces in the propeller wash region allowed the miniature UAV to takeoff and land vertically, hover, and fly level, meeting the needs of a miniature VTOL tailsitter UAV hardware testbed. A picture of the tailsitter built for flight testing can be seen in Figure 2.4.



Figure 2.4: The tailsitter that was built to perform flight testing is shown.



Figure 2.5: Above is the Kestrel autopilot that was used in flight testing.

2.2.2 Autopilot

For algorithm execution the airframe was equipped with an autopilot system. A major component of this system was the Kestrel Autopilot [36] in Figure 2.5, originally developed at Brigham Young University. The Kestrel Autopilot includes a main processor, onboard sensors, and means of driving actuators, ground station communication, and external sensor integration. Existing functional source code was modified to run the tailsitter algorithms.

The Kestrel autopilot is equipped with a 29 MHz Rabbit microcontroller with 512K Flash and 512K RAM. The sensors on the autopilot include rate gyros, accelerometers, an absolute pressure sensor for measuring altitude, and a differential pressure sensor for measuring airspeed.

To improve upon position estimation, a small, yet accurate and fast GPS with 4 Hz update (Ublox LEA-4H SuperSense GPS Module [37]) was selected along with a reliable active omnidirectional antenna (Sarentel GeoHelix-S [38]) that allowed for satellite tracking amidst large changes in aircraft attitude. The GPS module and antenna can be seen in Figure 2.6. An external three-axis magnetometer (PNI MicroMag3 [39]) in Figure 2.7 was also integrated for improved bearing estimation in hover flight. The autopilot source code was altered as well to effectively communicate with these external sensors.

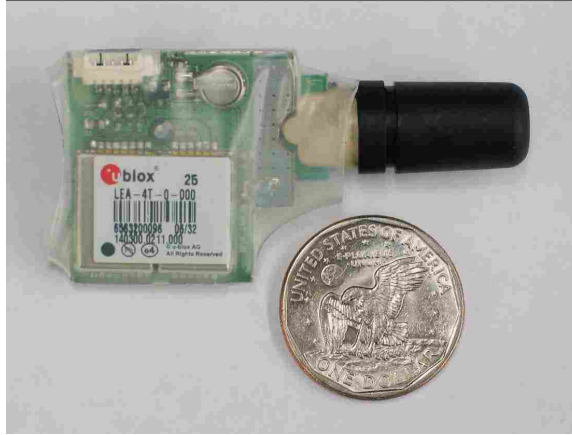


Figure 2.6: This figure shows the Ublox GPS module and Sarentel omnidirectional antenna that were used in flight testing.



Figure 2.7: Above is the three-axis PNI magnetometer used in flight testing

2.2.3 Ground Station

A user operated ground station (Figure 2.8) was employed for user interface with the tailsitter autopilot. This system permitted user inputs to the autopilot in many forms and gave efficient feedback of important information. The ground station hardware consisted of a laptop for running user interface software, a comm box and antenna for modem communication with the autopilot, and an RC transmitter for manual override of the autopilot actuator outputs. Virtual Cockpit user interface software, which also was developed at BYU, displayed telemetry acquired by the comm box. High level commands, such as waypoint paths and control loop configurations, could be uploaded, and many variables could be set from the laptop via the user interface.

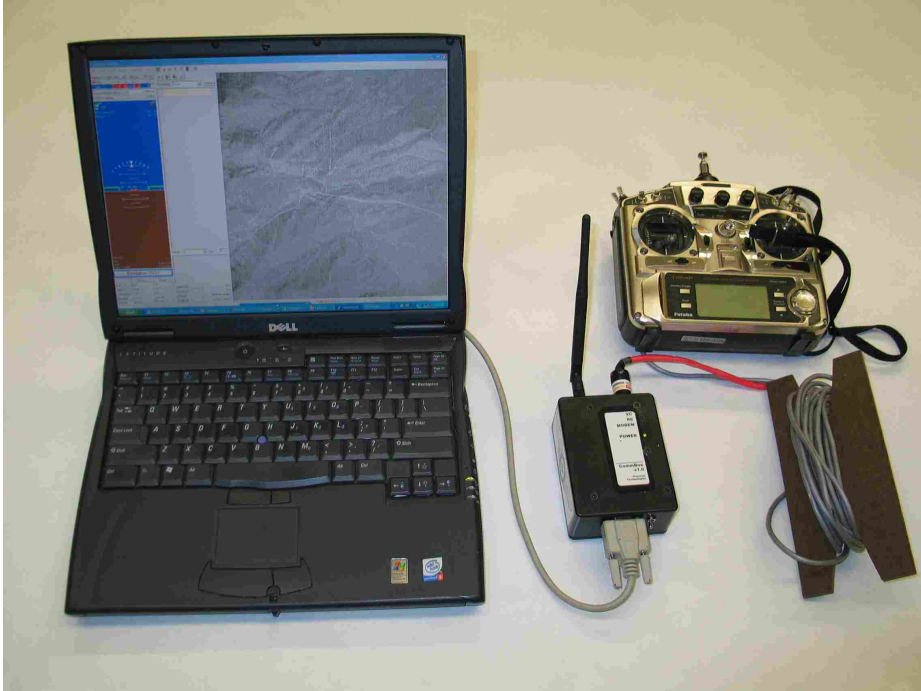


Figure 2.8: The ground station employed in flight testing is shown.

2.3 Summary

In this chapter, the research testbed was presented. This included a discussion of the tailsitter simulation environment developed for simplistic testing and assessment of autopilot algorithms. The testbed also consisted of a hardware flight test system. For this, an actual miniature tailsitter platform was produced and integrated with an autopilot system. Available source code for the autopilot and ground station was altered to execute the tailsitter algorithms derived in this research.

In the next chapter (Chapter 3), attitude control, the main objective of the research, will be discussed. Adaptive quaternion control with two methods of system identification will be derived. A conventional technique will be used to design a baseline controller. These methods will be compared in simulation and hardware results.

Chapter 3

Attitude Control

In this chapter, tailsitter attitude control will be discussed. A quaternion feedback method based upon the established technique of gain-scheduled PID control will be put forth as a means of determining the baseline performance. Adaptive quaternion attitude control, with two methods of system identification, will be derived to improve aircraft performance amid changing dynamics and despite poor modeling and sensing capabilities. These methods are designed with computational efficiency in mind, due to processor limitations. Simulation results for all of the controllers will be presented and discussed, as well as hardware results, for the most effective methods as determined from simulation. In this chapter, the term \bar{V} is introduced to represent the dominant source of airflow over the control surfaces, which is either aircraft velocity in level flight or propeller wash velocity in hover. In these algorithms, because only one measurement of airspeed is available, propeller wash airspeed is estimated from measured thrust, while aircraft airspeed is measured from the onboard differential pressure sensor.

3.1 Gain-Scheduled Quaternion Feedback Attitude Control

As a metric for determining the effectiveness of adaptive attitude techniques, gain-scheduled quaternion feedback control has been developed for level and hover flight. Due to the lack of an accurate tailsitter model and the limited sensing capabilities, feedback control gains have been tuned empirically from flight test observations and performance analysis.

For quaternion attitude error, consider the following equation,

$$\bar{\eta}_d = \bar{\eta}_e \otimes \bar{\eta}, \quad (3.1)$$

where $\bar{\eta}_d$ and $\bar{\eta}$ represent the aircraft's desired and actual attitude, respectively. $\bar{\eta}_e$ is thus the error between the two expressed in the aircraft's body reference frame. The error quaternion (from Appendix A.4) is then

$$\bar{\eta}_e = \{\bar{\eta}\}_R^T \bar{\eta}_d = \begin{pmatrix} -\eta_{4d}\eta + \eta_4\eta_d - \eta \times \eta_d \\ \eta_4\eta_{4d} + \eta^T \eta_d \end{pmatrix}. \quad (3.2)$$

Since the first three elements of the error quaternion (η_e) define the unit norm axis of rotation from the actual to desired attitude, scaled by $\sin(\Theta_e/2)$, the three aircraft actuators (aileron, elevator, and rudder) can be employed directly to drive Θ_e to zero, given η_e . Thus, provided a desired quaternion and existing angular rates (p , q , and r), scheduled PID gains can be applied for stable feedback attitude control. Figure 3.1 shows the structure of the gain scheduled controller. k_p , k_i , and k_d are diagonal PID gain matrices. Two sets of attitude gains have been tuned: one for hover and one for level flight. These gains are scheduled by either the aircraft airspeed in level flight or the propeller wash airspeed in hover. Logic for switching between gains and the type of airspeed measurement is employed for applying the proper set based upon the current flight regime. Note that the controller is designed to track $\bar{\eta}_m$ which is the reference model system. This acts as a pre-filter for smoothing large changes in the desired attitude trajectory. It also represents the desired obtainable tailsitter performance (see Section 3.2.2).

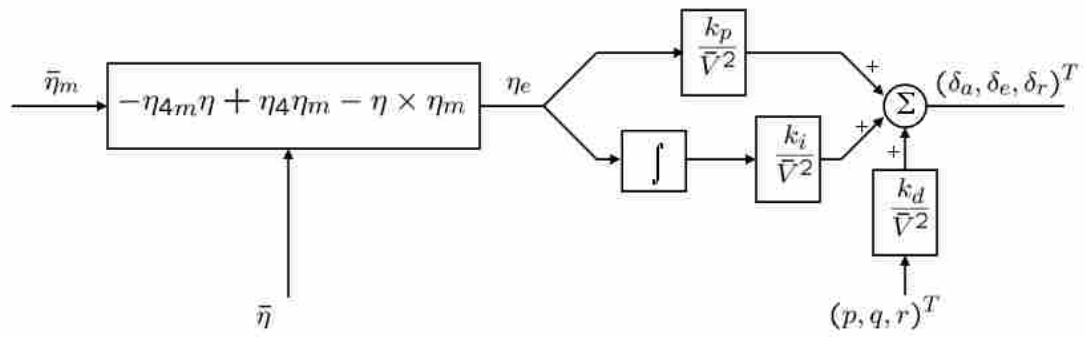


Figure 3.1: This figure shows a block diagram of the baseline PID gain-scheduled quaternion feedback controller.

3.2 Adaptive Attitude Control

Adaptive quaternion control has been developed to address challenges of miniature tailsitters due to drastic changes in system dynamics and limitations in both modeling and sensing. The structure chosen for adaptive control is an indirect model reference adaptive method. System parameters are identified on-line and Lyapunov-based backstepping control is utilized to cancel estimated dynamics and produce desirable reference model performance. The basic structure of the method can be seen in Figure 3.2. Two different techniques of on-line parameter estimation have been derived, namely least-squares and Lyapunov-based methods.

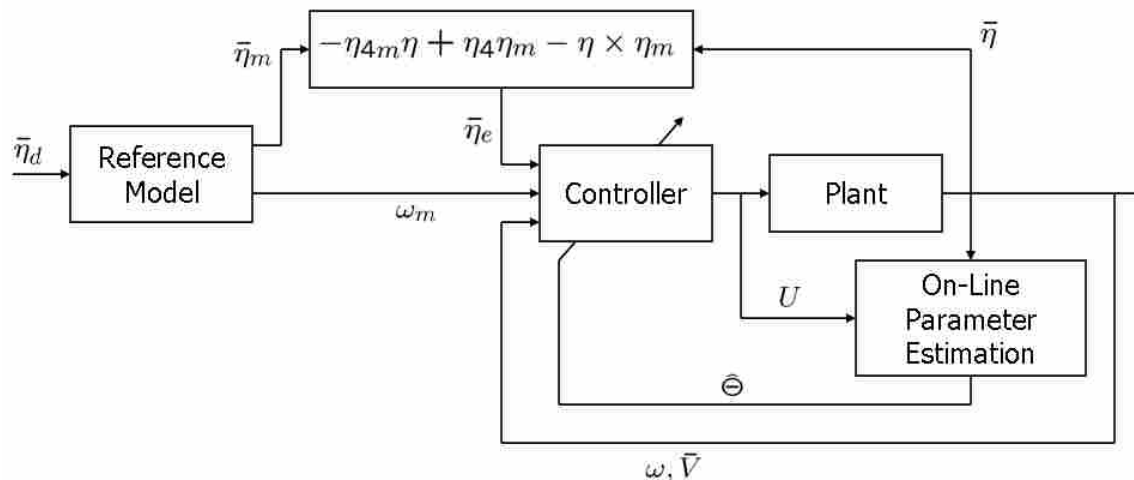


Figure 3.2: A block diagram of the model reference adaptive quaternion controller is presented.

3.2.1 Aircraft Attitude Dynamics

In this section, attitude dynamics are modeled for controller derivation. The complex angular acceleration dynamics are simplified to facilitate on-line parameter estimation that is computationally easy, yet effective. Based upon knowledge about the system, these simplifications are chosen with design considerations in mind, in

order to reduce computation and still capture the major contributions of angular acceleration given the information available to the autopilot.

For rigid-body aircraft, such as the tailsitter, the rotational acceleration dynamics are described as

$$\dot{\omega} = -J^{-1}\omega \times J\omega + J^{-1}M, \quad (3.3)$$

where $\omega = (p, q, r)^T$ and $M = (l, m, n)^T$ are the vectors of body frame angular velocities and moments and J is the constant positive-definite symmetric inertia matrix:

$$J = \begin{pmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{yz} & J_{zz} \end{pmatrix}. \quad (3.4)$$

Assuming symmetry in the aircraft's xz -plane, equation (3.3) can be written explicitly as

$$\begin{aligned} \dot{p} &= \frac{J_{xz}(J_{xx} - J_{yy} + J_{zz})}{J_{xx}J_{zz} - J_{xz}^2}pq - \frac{J_{zz}(J_{zz} - J_{yy}) + J_{xz}^2}{J_{xx}J_{zz} - J_{xz}^2}qr \\ &\quad + \frac{J_{zz}}{J_{xx}J_{zz} - J_{xz}^2}l(V, V_{pw}, \beta, p, \delta_a) + \frac{J_{xz}}{J_{xx}J_{zz} - J_{xz}^2}n(V, V_{pw}, \beta, r, \delta_r), \\ \dot{q} &= \frac{J_{zz} - J_{xx}}{J_{yy}}pr - \frac{J_{xz}}{J_{yy}}(p^2 - r^2) + \frac{1}{J_{yy}}m(V, V_{pw}, \alpha, q, \delta_e), \text{ and} \\ \dot{r} &= \frac{J_{zz}(J_{zz} - J_{yy}) + J_{xz}^2}{J_{xx}J_{zz} - J_{xz}^2}pq - \frac{J_{xz}(J_{xx} - J_{yy} + J_{zz})}{J_{xx}J_{zz} - J_{xz}^2}qr \\ &\quad + \frac{J_{xz}}{J_{xx}J_{zz} - J_{xz}^2}l(V, V_{pw}, \beta, p, \delta_a) + \frac{J_{zz}}{J_{xx}J_{zz} - J_{xz}^2}n(V, V_{pw}, \beta, r, \delta_r). \end{aligned} \quad (3.5)$$

Note that for the proposed on-line estimation methods, linearity in estimated parameters is required. Also, computation increases for every additional term estimated, significantly in the case of least squares. As a result, angular acceleration is modeled with the fewest number of linear terms that adequately describe the system. From simple modeling of miniature UAVs, one can note that the major components of angular acceleration are due to incorrect trim (constant bias acceleration) and control surface input. Other elements, such as gyroscopic effects, are significantly smaller and prominent components such as accelerations from angle of attack and sideslip angle

are undetectable because of sensing limitations. Aerodynamic damping is a source of considerable acceleration, and associated information is available to the autopilot in the form of angular rates. However, these accelerations are not as significant as bias and actuator input terms. Therefore, for model simplification, angular acceleration about each body referenced axis is modeled with one bias acceleration term and one actuator-based input term:

$$\begin{aligned}\dot{p} &= \theta_1 + \bar{V}^2\theta_2\delta_a, \\ \dot{q} &= \theta_3 + \bar{V}^2\theta_4\delta_e, \text{ and} \\ \dot{r} &= \theta_5 + \bar{V}^2\theta_6\delta_r.\end{aligned}\tag{3.6}$$

This simplified model reduces computation significantly for parameter estimation and control. With only two parameters to identify for each axis, recursive least squares can be performed with the simple inversion of three 2×2 matrices. Considering Chapter 2, it can be seen that torques in the level flight and propeller wash models are linear in one main actuator scaled by several constants (which denote actuator effectiveness) and either V^2 or V_{pw}^2 , as modeled above. All other contributions to angular acceleration are lumped into the bias terms. Note that the addition of aerodynamic damping terms to the model would result in the inversion of three 3×3 matrices for least-squares estimation, causing a drastic increase in computation. In view of design tradeoffs, only two terms were chosen to represent acceleration, under the assumption that the method will be able to track dynamics with the reduced model.

The angular acceleration model equation (3.6) can now be rewritten in vector form as

$$\dot{\omega} = C_1 + \bar{V}^2C_2U,\tag{3.7}$$

where $C_1 = (\theta_1, \theta_3, \theta_5)^T$ and $U = (\delta_a, \delta_e, \delta_r)^T$ are vectors of unknown bias parameters and actuator settings, and C_2 is a diagonal matrix of unknown actuator scaling

parameters:

$$C_2 = \begin{pmatrix} \theta_2 & 0 & 0 \\ 0 & \theta_4 & 0 \\ 0 & 0 & \theta_6 \end{pmatrix}.$$

From the quaternion time derivative (Appendix A.2), the attitude kinematics are

$$\dot{\eta} = \frac{1}{2}(\eta^\times + \eta_4 I_3)\omega \quad (3.8)$$

and

$$\dot{\eta}_4 = -\frac{1}{2}\eta^T \omega, \quad (3.9)$$

where the notation ξ^\times denotes the skew-symmetric matrix given by:

$$\xi^\times = \begin{pmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{pmatrix}.$$

The tailsitter rotational dynamics are therefore represented with equations (3.7), (3.8), and (3.9).

3.2.2 Reference Model Dynamics

Reference model dynamics are derived to smooth desired attitude trajectories. The reference model is chosen to have desired performance that is attainable by the actual system. Because a second-order model is used in the derivation of aircraft dynamics the reference model is also designed with second-order dynamics. These basic guidelines are pertinent to Lyapunov-based control and parameter estimation.

The reference model quaternion ($\bar{\eta}_m$) has the same kinematic relationship with the reference model angular velocities (ω_m) as given in equations (3.8) and (3.9), namely

$$\dot{\eta}_m = \frac{1}{2}(\eta_m^\times + \eta_{4m} I_3)\omega_m \quad (3.10)$$

and

$$\dot{\eta}_{4m} = -\frac{1}{2}\eta_m^T \omega_m. \quad (3.11)$$

For second order dynamics, the reference model angular accelerations are defined as

$$\dot{\omega}_m = -2\zeta\omega_n\omega_m + \omega_n^2\eta_{em}, \quad (3.12)$$

where ζ and ω_n are the damping ratio and natural frequency of the reference model and $\bar{\eta}_{em}$ is the error quaternion from the reference model to the commanded attitude.

3.2.3 Error Dynamics

In this section, attitude error dynamics are discussed. Error dynamics are derived to simplify the development of the backstepping controller. Furthermore, later in the chapter the discussion will return to this subject to show that with the developed adaptive control method, system performance, in the form of error dynamics, is not influenced by the vehicle dynamics, due to cancellation. This is an attractive feature for control systems.

Quaternion attitude error ($\bar{\eta}_e$) is defined as the rotation between the actual attitude and the controller reference model ($\bar{\eta}_m = \bar{\eta}_e \otimes \bar{\eta}$). The tailsitter attitude error (see Appendix A.4) can then be written as

$$\eta_e = -\eta_{4m}\eta + \eta_4\eta_m - \eta^\times\eta_m \quad (3.13)$$

and

$$\eta_{4e} = \eta_4\eta_{4m} + \eta^T\eta_m. \quad (3.14)$$

Rotation matrices (R and R_m) which transform from the inertial frame into the actual body reference frame and the reference model reference frame are defined as

$$R = (\eta_4^2 - \eta^T\eta)I_3 + 2\eta\eta^T - 2\eta_4\eta^\times \quad (3.15)$$

and

$$R_m = (\eta_{4m}^2 - \eta_m^T \eta_m) I_3 + 2\eta_m \eta_m^T - 2\eta_{4m} \eta_m^\times. \quad (3.16)$$

To rotate from the reference model reference frame to the actual body reference frame, the following rotation matrix (\tilde{R}) is defined as

$$\tilde{R} = R R_m^T = (\eta_{4e}^2 - \eta_e^T \eta_e) I_3 + 2\eta_e \eta_e^T + 2\eta_{4e} \eta_e^\times. \quad (3.17)$$

Given equations (3.8), (3.9), (3.10), (3.11), (3.13), and (3.14) the attitude error dynamics can be defined as

$$\dot{\eta}_e = \frac{1}{2}(\eta_e^\times + \eta_{4e} I_3)(\tilde{R}\omega_m - \omega) \quad (3.18)$$

and

$$\dot{\eta}_{4e} = -\frac{1}{2}\eta_e^T(\tilde{R}\omega_m - \omega). \quad (3.19)$$

3.2.4 Backstepping Attitude Controller Derivation

Derivation of the adaptive quaternion controller is based upon Lyapunov theory. Because the system has a relative degree of two, the backstepping method is utilized to drive attitude error between the actual and reference model to zero by tracking desired angular rates with actuator inputs. As a result, the system is designed to asymptotically converge on desired angular rates and reference model attitude.

Let

$$\mathcal{V}' = \frac{1}{2}\eta_e^T \eta_e \quad (3.20)$$

be the initial candidate Lyapunov equation. Differentiating equation (3.20),

$$\dot{\mathcal{V}}' = \eta_e^T \dot{\eta}_e = \eta_e^T \frac{1}{2}(\eta_e^\times + \eta_{4e} I_3)(\tilde{R}\omega_m - \omega)$$

is obtained from equation (3.18). The derivation goal is for $\dot{\mathcal{V}}' = -k_1\eta_e^T\eta_e$, where k_1 is a scalar gain. Therefore, the desired angular velocity (ω_d) can be defined as

$$\omega_d = 2k_1(\eta_e^\times + \eta_{4e}I_3)^{-1}\eta_e + \tilde{R}\omega_m. \quad (3.21)$$

Thus,

$$\dot{\mathcal{V}}' = \eta_e^T \frac{1}{2}(\eta_e^\times + \eta_{4e}I_3)(\tilde{R}\omega_m - \omega_d + \omega_d - \omega)$$

can be rewritten with equation (3.21) as

$$\dot{\mathcal{V}}' = -k_1\eta_e^T\eta_e + \eta_e^T \frac{1}{2}(\eta_e^\times + \eta_{4e}I_3)\tilde{\omega}, \quad (3.22)$$

where $\tilde{\omega}$ is the backstepping variable:

$$\tilde{\omega} = \omega_d - \omega. \quad (3.23)$$

Let

$$\mathcal{V} = \frac{1}{2}\eta_e^T\eta_e + \frac{1}{2}\tilde{\omega}^T\tilde{\omega} \quad (3.24)$$

be our final candidate Lyapunov equation. Equation (3.7) combined with equation (3.24) differentiated yields

$$\dot{\mathcal{V}} = \eta_e^T \dot{\eta}_e + (\dot{\omega}_d - C_1 - \bar{V}^2 C_2 U)^T \tilde{\omega}.$$

Noting equation (3.22),

$$\dot{\mathcal{V}} = -k_1\eta_e^T\eta_e + \eta_e^T \frac{1}{2}(\eta_e^\times + \eta_{4e}I_3)\tilde{\omega} + (\dot{\omega}_d - C_1 - \bar{V}^2 C_2 U)^T \tilde{\omega}. \quad (3.25)$$

Consequently, the chosen control for model cancellation and asymptotic convergence is

$$U = \frac{1}{V^2} C_2^{-1} \left(k_2 \tilde{\omega} + \frac{1}{2} (-\eta_e^\times + \eta_{4e} I_3) \eta_e + \dot{\omega}_d \right) - \frac{1}{V^2} C_2^{-1} C_1, \quad (3.26)$$

where k_2 is a scalar gain. Thus,

$$\dot{\mathcal{V}} = -k_1 \eta_e^T \eta_e - k_2 \tilde{\omega}^T \tilde{\omega} \quad (3.27)$$

and with Lyapunov arguments it can be shown that η_e and $\tilde{\omega} \rightarrow 0$ as $t \rightarrow \infty$.

Consider the error dynamics shown in equation (3.18). With the addition and subtraction of desired angular rates,

$$\dot{\eta}_e = \frac{1}{2} (\eta_e^\times + \eta_{4e} I_3) (\tilde{R} \omega_m - \omega_d + \omega_d - \omega),$$

which combined with equation (3.21) can be written as

$$\dot{\eta}_e = -k_1 \eta_e + \frac{1}{2} (\eta_e^\times + \eta_{4e} I_3) \tilde{\omega}. \quad (3.28)$$

Also,

$$\dot{\tilde{\omega}} = \dot{\omega}_d - \dot{\omega}, \quad (3.29)$$

with the equations of angular acceleration (3.7) and control (3.26), produce

$$\dot{\tilde{\omega}} = -k_2 \tilde{\omega} - \frac{1}{2} (-\eta_e^\times + \eta_{4e} I_3) \eta_e. \quad (3.30)$$

Noting equations (3.28) and (3.30), one can see that because of cancellation, the error dynamics are not influenced by the system parameters. As a direct consequence, one set of control gains (k_1 and k_2) will produce the same performance universally for any aircraft, if system parameters are identified correctly and saturation does not occur. As this implies, the algorithm gains are required to be tuned only once for the

consistent performance of any airframe design (with all three control surface inputs available). This is an attractive feature of the algorithm.

3.2.5 Parameter Estimation

The backstepping adaptive controller described in the previous section requires real time accurate estimation of system parameters. Computational limitations of the autopilot processor chosen for the experiment require the estimation algorithms to be efficient. Two simple techniques, namely least-squares and Lyapunov-based methods developed for on-line parameter estimation, will be discussed in this section.

Stabilized Recursive Least Squares With Data Forgetting

An effective technique of parameter estimation is optimization. System parameters can be chosen to minimize the error between estimated and observed or measured dynamics. Solving for the least-squares solution is a proven method, which lends itself well to a recursive algorithm. For adaptation, data forgetting, where recent data is weighted higher than old data, can be applied easily. Consequently, if system parameters change, weighting on new performance data allows the algorithm to learn the new parameters. Also, regularization, a stabilizing technique, can be employed to address instabilities introduced from noisy data combined with data forgetting.

From equation (3.6) the angular acceleration dynamics can be written as

$$\begin{aligned} \dot{p} &= \Phi_1^T \Theta_1, \\ \dot{q} &= \Phi_2^T \Theta_2, \text{ and} \\ \dot{r} &= \Phi_3^T \Theta_3, \end{aligned} \tag{3.31}$$

where

$$\Phi_1 = \begin{pmatrix} 1 \\ \bar{V}^2 \delta_a \end{pmatrix}, \quad \Phi_2 = \begin{pmatrix} 1 \\ \bar{V}^2 \delta_e \end{pmatrix}, \text{ and} \quad \Phi_3 = \begin{pmatrix} 1 \\ \bar{V}^2 \delta_r \end{pmatrix},$$

and

$$\Theta_1 = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, \quad \Theta_2 = \begin{pmatrix} \theta_3 \\ \theta_4 \end{pmatrix}, \quad \text{and} \quad \Theta_3 = \begin{pmatrix} \theta_5 \\ \theta_6 \end{pmatrix}.$$

Considering only the angular rate dynamics about the body reference frame x -axis, the following convex function is chosen to be minimized:

$$J = \sum_{k=1}^N \frac{1}{2} (\Phi_1[k]^T \hat{\Theta}_1[N] - \dot{p}[k])^2 \lambda^{N-k} + \sum_{k=1}^2 \frac{1}{2} \alpha_k (\hat{\theta}_k[N] - \hat{\theta}_k[N-1])^2. \quad (3.32)$$

The term λ weights the data according to the iteration difference $N - k$. A value between 0 and 1 is chosen for λ , where a value of one weights all of the past data the same; the closer the value gets to zero the less it weights old data (see Figure 3.3). The terms α_k penalize large changes in the estimated parameters from one time step to the next. This is a form of regularization that stabilizes the algorithm in the presence of noisy data when weighting is applied.

The gradient of J with respect to $\hat{\Theta}_1$ is

$$\nabla J = \sum_{k=1}^N \Phi_1[k] (\Phi_1[k]^T \hat{\Theta}_1[N] - \dot{p}[k]) \lambda^{N-k} + \alpha (\hat{\Theta}_1[N] - \hat{\Theta}_1[N-1]), \quad (3.33)$$

where α denotes the diagonal matrix

$$\alpha = \begin{pmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{pmatrix}.$$

The least-squares solution is then found by setting $\nabla J = 0$ and solving for $\hat{\Theta}_1[N]$ as follows:

$$\hat{\Theta}_1[N] = \left(\sum_{k=1}^N \Phi_1[k] \Phi_1[k]^T \lambda^{N-k} + \alpha \right)^{-1} \sum_{k=1}^N \Phi_1[k] \dot{p}[k] \lambda^{N-k} + \alpha \hat{\Theta}_1[N-1]. \quad (3.34)$$

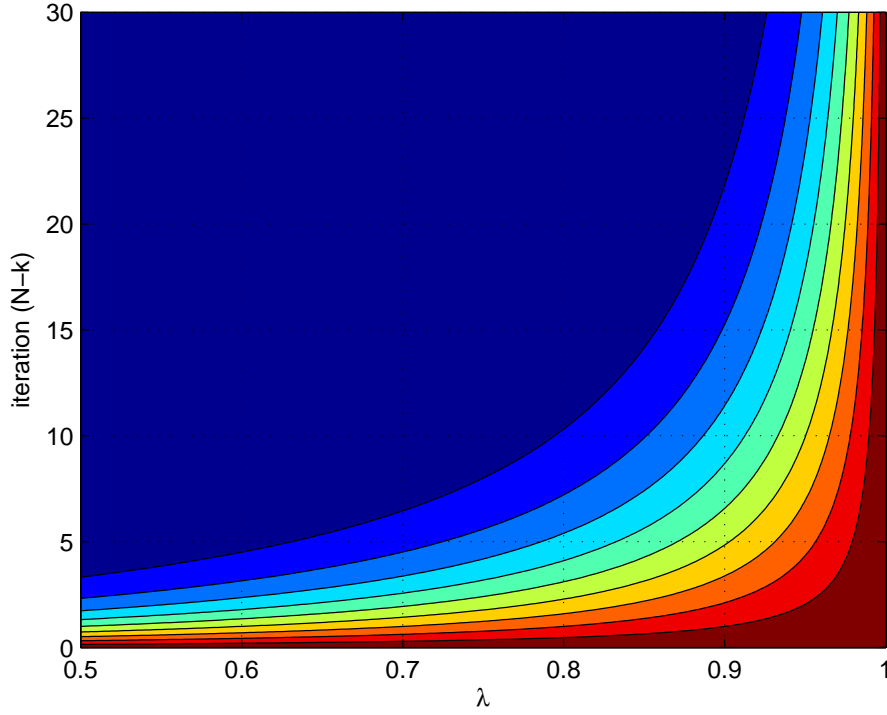


Figure 3.3: This figure is a contour plot of data weighting as a function of λ and $N - k$. All contour lines occur at a weight increment of 0.1. Thus dark blue represents a weight of 0 to 0.1 and Red a weight of 0.9 to 1.0.

Because equation (3.32) is convex and the problem is unconstrained, equation (3.34) is a global minimum. Note that the inverse term is the covariance matrix

$$P[N] = \left(\sum_{k=1}^N \Phi_1[k] \Phi_1[k]^T \lambda^{N-k} + \alpha \right)^{-1}. \quad (3.35)$$

Equation (3.34) can be written as

$$\left(\sum_{k=1}^N \Phi_1[k] \Phi_1[k]^T \lambda^{N-k} + \alpha \right) \hat{\Theta}_1[N] = \sum_{k=1}^N \Phi_1[k] \dot{p}[k] \lambda^{N-k} + \alpha \hat{\Theta}_1[N-1]. \quad (3.36)$$

Setting $\hat{\Theta}_1[N] = \hat{\Theta}_1[N-1] + \delta\hat{\Theta}_1$ and pulling out the last iteration of the summations results in

$$\begin{aligned} & \left(\sum_{k=1}^{N-1} \Phi_1[k]\Phi_1[k]^T \lambda^{N-k} + \Phi_1[N]\Phi_1[N]^T + \alpha \right) (\hat{\Theta}_1[N-1] + \delta\hat{\Theta}_1) \\ &= \sum_{k=1}^{N-1} \Phi_1[k]\dot{p}[k]\lambda^{N-k} + \Phi_1[N]\dot{p}[N] + \alpha\hat{\Theta}_1[N-1]. \end{aligned} \quad (3.37)$$

Noting that

$$\sum_{k=1}^{N-1} \Phi_1[k]\Phi_1[k]^T \lambda^{N-k} \hat{\Theta}_1[N-1] = \sum_{k=1}^{N-1} \Phi_1[k]\dot{p}[k]\lambda^{N-k},$$

we solve for $\delta\hat{\Theta}_1$ as

$$\delta\hat{\Theta}_1 = P[N]\Phi_1[N](\dot{p} - \Phi_1[N]^T\Theta_1[N-1]). \quad (3.38)$$

The recursive algorithm for the least-squares solution is then

$$\hat{\Theta}_1[N] = \hat{\Theta}_1[N-1] + P[N]\Phi_1[N](\dot{p} - \Phi_1[N]^T\Theta_1[N-1]). \quad (3.39)$$

Considering equation (3.35), setting $P^{-1}[N] = \lambda P^{-1}[N-1] + \delta P^{-1}$ and pulling out the last iteration of the summation gives

$$\sum_{k=1}^{N-1} \Phi_1[k]\Phi_1[k]^T \lambda^{N-k} + \Phi_1[N]\Phi_1[N]^T + \alpha = \lambda P^{-1}[N-1] + \delta P^{-1}.$$

Substituting in $P^{-1}[N-1]$ yields

$$\sum_{k=1}^{N-1} \Phi_1[k]\Phi_1[k]^T \lambda^{N-k} + \Phi_1[N]\Phi_1[N]^T + \alpha = \lambda \sum_{k=1}^{N-1} \Phi_1[k]\Phi_1[k]^T \lambda^{N-k-1} + \alpha\lambda + \delta P^{-1}$$

and thus,

$$\delta P^{-1} = \Phi_1[N]\Phi_1[N]^T + \alpha(1 - \lambda). \quad (3.40)$$

As a result, the recursive algorithm for $P^{-1}[N]$ is

$$P^{-1}[N] = \lambda P^{-1}[N - 1] + \Phi_1[N]\Phi_1[N]^T + \alpha(1 - \lambda). \quad (3.41)$$

This procedure can be easily reproduced for estimating angular acceleration parameters associated with the other two body-referenced axes. Also note that this algorithm is computationally simple. Three matrix inverses are required. However, since only two parameters are being identified, the matrices are only 2×2 , which is computationally easy, resulting in little burden on the autopilot processor.

The described algorithm requires a measurement of angular acceleration. Because this is not available, angular rates from gyros are differentiated with the simple backwards difference derivative. Data mismatching caused by actuator dynamics and the backwards difference derivative of angular rates is overcome by delaying recorded actuator settings and velocity terms from buffered data.

Lyapunov Method

For the second method of system parameter estimation developed, Barbalat's lemma and Lyapunov theory are used for algorithm derivation. This method does not ensure parameter convergence. The update law, however, is chosen to produce asymptotic tracking in controller performance.

Let

$$\mathcal{V} = \frac{1}{2}\eta_e^T \eta_e + \frac{1}{2}\tilde{\omega}^T \tilde{\omega} \quad (3.42)$$

be the initial candidate Lyapunov equation. From equation (3.25),

$$\dot{\mathcal{V}} = -k_1 \eta_e^T \eta_e + \eta_e^T \frac{1}{2}(\eta_e^\times + \eta_{4e} I_3) \tilde{\omega} + (\dot{\omega}_d - C_1 - \bar{V}^2 C_2 U)^T \tilde{\omega}.$$

Therefore, the desired control (U_d) is defined as

$$U_d = C_2^{-1} \frac{1}{\bar{V}^2} \left(k_2 \tilde{\omega} + \frac{1}{2}(-\eta_e^\times + \eta_{4e} I_3) \eta_e + \dot{\omega}_d \right) - C_2^{-1} \frac{1}{\bar{V}^2} C_1. \quad (3.43)$$

Equation (3.43) can be rewritten as follows:

$$U_d = \Psi^T \Xi, \quad (3.44)$$

where Ψ is a matrix of known values

$$\Psi = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ n_1 & 0 & 0 \\ 0 & n_2 & 0 \\ 0 & 0 & n_3 \end{pmatrix}, \quad (3.45)$$

n_1 , n_2 , and n_3 being the elements of the vector

$$N = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} = \frac{1}{\bar{V}^2} \left(k_2 \tilde{\omega} + \frac{1}{2} (-\eta_e^\times + \eta_{4e} I_3) \eta_e + \dot{\omega}_d \right). \quad (3.46)$$

Ξ can be defined as a vector of unknown parameters

$$\Xi = \begin{pmatrix} \theta_1/\theta_2 \\ \theta_3/\theta_4 \\ \theta_5/\theta_6 \\ 1/\theta_2 \\ 1/\theta_4 \\ 1/\theta_6 \end{pmatrix}. \quad (3.47)$$

The actual controller input is then

$$U = \Psi^T \hat{\Xi}, \quad (3.48)$$

where $\hat{\Xi}$ is composed of estimated parameters. Thus,

$$\dot{\mathcal{V}}' = -k_1 \eta_e^T \eta_e + \frac{1}{2} \eta_e^T (\eta_e^\times + \eta_{4e} I_3) \tilde{\omega} + (\dot{\omega}_d - C_1 - \bar{V}^2 C_2 U_d + \bar{V}^2 C_2 U_d - \bar{V}^2 C_2 U)^T \tilde{\omega}$$

reduces to

$$\dot{\mathcal{V}}' = -k_1 \eta_e^T \eta_e - k_2 \tilde{\omega}^T \tilde{\omega} + \bar{V}^2 \tilde{\omega}^T C_2 \Psi^T \tilde{\Xi},$$

where

$$\tilde{\Xi} = \Xi - \hat{\Xi}. \quad (3.49)$$

Let

$$\mathcal{V} = \frac{1}{2} \eta_e^T \eta_e + \frac{1}{2} \tilde{\omega}^T \tilde{\omega} + \frac{1}{2} \tilde{\Xi}^T \Gamma^{-1} C_3 \tilde{\Xi} \quad (3.50)$$

be the final candidate Lyapunov equation, where $\Gamma \in R^{6 \times 6}$ is a diagonal positive definite gain matrix and C_3 is a diagonal matrix of unknown parameters defined as follows:

$$C_3 = \begin{pmatrix} \theta_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \theta_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & \theta_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & \theta_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \theta_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & \theta_6 \end{pmatrix}. \quad (3.51)$$

Assuming that the parameters are slowly varying, C_3 is approximately constant and $\dot{\tilde{\Xi}} \approx -\dot{\hat{\Xi}}$. Under this assumption, differentiating equation (3.50) and noting that $C_2 \Psi^T = \Psi^T C_3$, we obtain

$$\dot{\mathcal{V}} = -k_1 \eta_e^T \eta_e - k_2 \tilde{\omega}^T \tilde{\omega} + \bar{V}^2 \tilde{\omega}^T \Psi^T C_3 \tilde{\Xi} - \dot{\tilde{\Xi}}^T \Gamma^{-1} C_3 \tilde{\Xi},$$

which factors to

$$\dot{\mathcal{V}} = -k_1 \eta_e^T \eta_e - k_2 \tilde{\omega}^T \tilde{\omega} + (\bar{V}^2 \Psi \tilde{\omega} - \Gamma^{-1} \dot{\hat{\Xi}})^T C_3 \tilde{\Xi}.$$

For cancellation, the estimated parameter update law is

$$\dot{\hat{\Xi}} = \bar{V}^2 \Gamma \Psi \tilde{\omega}. \quad (3.52)$$

As a result,

$$\dot{\mathcal{V}} = -k_1 \eta_e^T \eta_e - k_2 \tilde{\omega}^T \tilde{\omega}. \quad (3.53)$$

From Lyapunov theory it follows that

$$\lim_{t \rightarrow \infty} \int_0^t \dot{\mathcal{V}}(\tau) d\tau \leq \mathcal{V}(t_0).$$

Also noting that equation (3.53) is continuous and that

$$\ddot{\mathcal{V}} = -2k_1 \eta_e^T \dot{\eta}_e - 2k_2 \tilde{\omega}^T \dot{\tilde{\omega}} \quad (3.54)$$

is bounded, it can be shown that $\dot{\mathcal{V}}$ is uniformly continuous. In consequence, from Barbalat's lemma, $\dot{\mathcal{V}} \rightarrow 0$ as $t \rightarrow \infty$, which implies $\eta_e = \tilde{\omega} = 0$. From equations (3.21) and (3.23),

$$\tilde{\omega} = 2k_1 (\eta_e^\times + \eta_{4e} I_3)^{-1} \eta_e + \tilde{R} \omega_m - \omega, \quad (3.55)$$

implying that $\tilde{R} \omega_m = \omega$. This shows that based upon the assumptions made previously, the system asymptotically approaches the reference model.

3.3 Simulation Results

Simulation results of the quaternion-based attitude control algorithms are presented in this section. Simulation was utilized as a tool to expedite the algorithms' investigation and development process. Information about performance was easily obtained, enabling the assessment of a method's effectiveness. For attitude control in simulation, trajectories were generated from navigational controllers described in Chapter 4; the actual quaternion attitude was estimated with the method described in Chapter 5. The simulated controllers flew the same desired flight path, which was chosen to represent all of the flight conditions a tailsitter might encounter. This

path was comprised of a hover takeoff, followed by a hover waypoint, a transition to level flight, two level waypoints, a transition to hover, a hover waypoint, and finally a hover land. Figure 3.4 gives an example of the desired path just described. The adaptive methods are compared to the baseline gain-scheduled PID controller. The backstepping adaptive controller was flown with both methods of parameter estimation that were derived earlier. Because the reference model pre-filters the attitude commands for all controllers, attitude error will be defined as the difference between the measured attitude and the reference model.

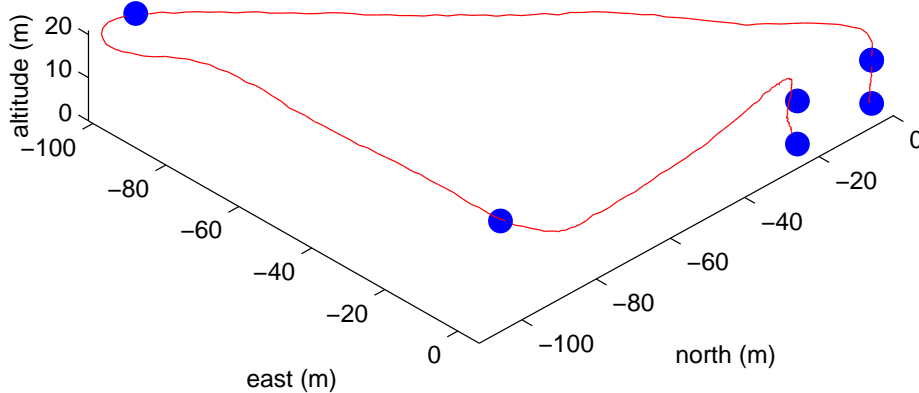


Figure 3.4: The waypoint path selected for the attitude control simulation experiment is shown above. A vertical takeoff was performed at 0 north and 0 east. This was followed by a hover waypoint, two level waypoints, another hover waypoint, and finally a hover land. This flight was executed with the adaptive backstepping controller and recursive least-squares parameter estimation.

3.3.1 Gain-Scheduled Quaternion Feedback Attitude Control

Simulation results of the baseline gain-scheduled quaternion feedback controller are presented in this section. These results will be compared to the other methods for a metric of performance.

Attitude tracking of the controller is shown in Figure 3.5. Performance as expected is reasonably good. It can be observed that level and hover flight PID gains, scheduled with airspeed and propeller wash airspeed squared, are effective at quaternion feedback control. No undesired behavior occurs during transitions, where PID gains switch from level and hover values. These transitions occur at around 5 and 35 seconds.

Quaternion error throughout the flight is shown in Figure 3.6. The first three terms of the error quaternion are used to indicate which axes mainly contributed to the attitude error. The quaternion error angle is also shown in the same figure. This term indicates the magnitude of attitude error. It can be seen that two initial spikes of error occur during the hover roll-to-heading and pitch-to-level flight maneuvers. Once in level flight, the major contributions of error occur in the yaw-to-turn, climb, and descent segments of the level flight, where error is mainly about the body frame y and z axes. It can also be observed that a rolling moment induced error in level flight from poor aileron trim is compensated with integrator control after about 10 seconds of flight. During the pitch-up rotation to hover, a spike in error about the body frame y -axis is shown as well. Attitude is well maintained for the rest of the flight.

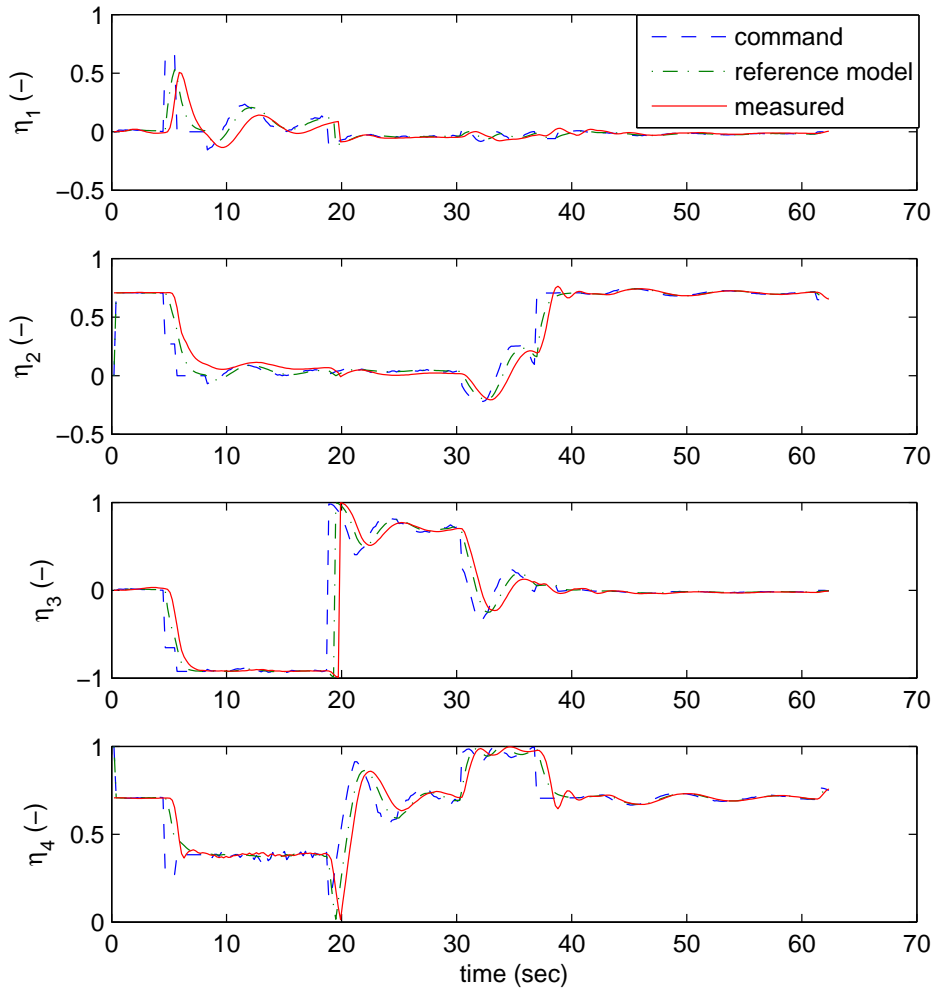


Figure 3.5: Attitude results of the simulated gain-scheduled quaternion feedback controller during the attitude experiment are shown.

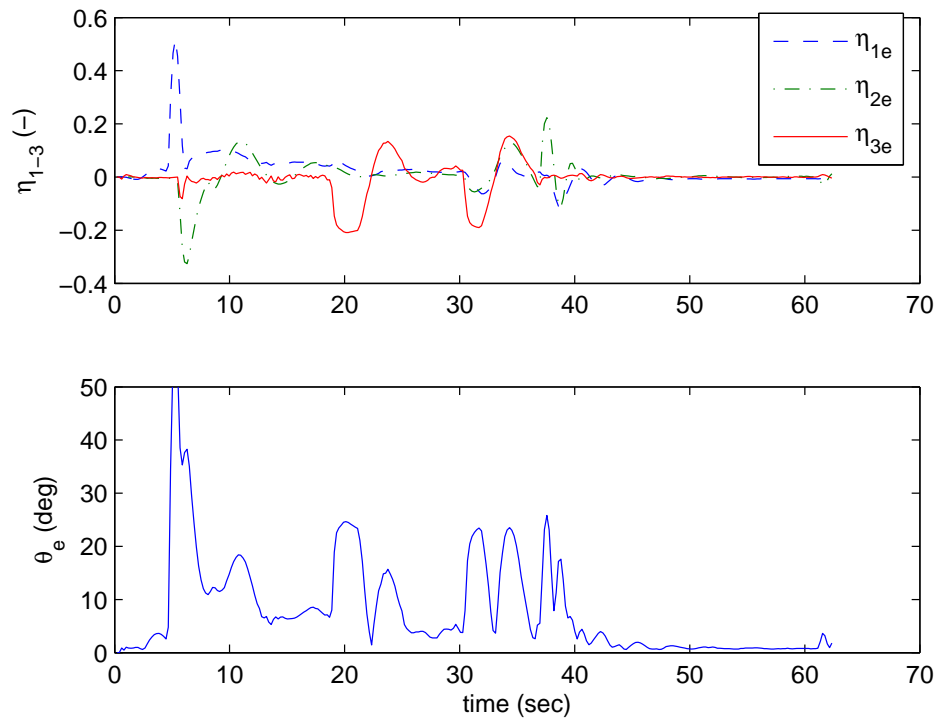


Figure 3.6: In this figure quaternion error measured between the actual and reference model during the simulation test of the gain-scheduled quaternion feedback controller is presented.

3.3.2 Stabilized Recursive Least Squares With Data Forgetting

The same simulation experiment was performed with the backstepping adaptive controller with least-squares parameter estimation. Simulated flight test results will be presented in this section. Quaternion attitude and error are given in Figure 3.7 and Figure 3.8. Significant reduction in error from the gain-scheduled control method is evident. Total error for the system never spikes above 13 degrees throughout the whole flight, even in transition maneuvers, which occur at 5 and 35 seconds. The spikes in error that do occur are from the same transitional, skid-to-turn, climb, and descent maneuvers that were discussed in the earlier section. Note that the rolling moment due to poor aileron trim causes no noticeable error. This indicates that the system has quickly learned and cancelled the bias roll angular acceleration directly after the hover-to-level transition was performed.

The basic concept of the backstepping method derived is that attitude tracking is performed through maintaining desired angular rates with accelerations produced by control surface deflections. Because of this, for good attitude performance angular rate tracking is required. Angular rates throughout the attitude experiment are shown to follow desired rates considerably well in Figure 3.9, facilitating the impressive performance of the attitude controller.

System identification performance in the form of estimated parameters can be seen in Figure 3.10. Note that parameters with odd subscripts represent angular accelerations lumped together as bias terms, while parameters with even subscripts indicate the associated control surface effectiveness, because these terms, along with \bar{V}^2 , linearly relate the actuator settings to angular acceleration. For example, the roll acceleration model is

$$\dot{p} = \theta_1 + \bar{V}^2 \theta_2 \delta_a.$$

In Figure 3.10 discussed above, it can be seen that despite drastic changes, the true parameters are identified accurately and with little delay. Consider for example, the parameter representing aileron effectiveness (θ_2) changes from 1.8 to 0.2 instantly during a transition from level to hover flight at 35 seconds. However, within a tenth

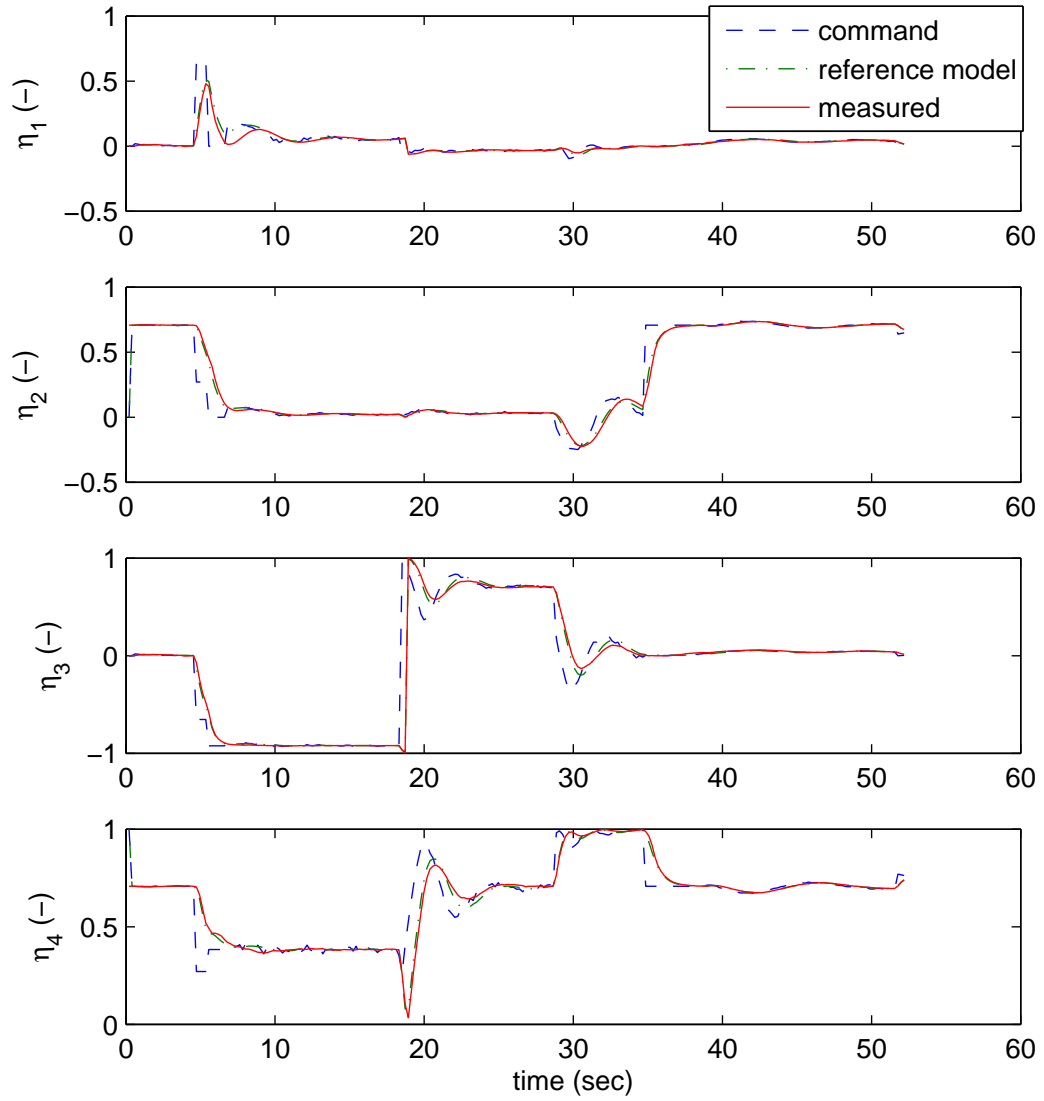


Figure 3.7: Attitude results during simulation testing of the backstepping adaptive controller with least-squares parameter estimation are given above.

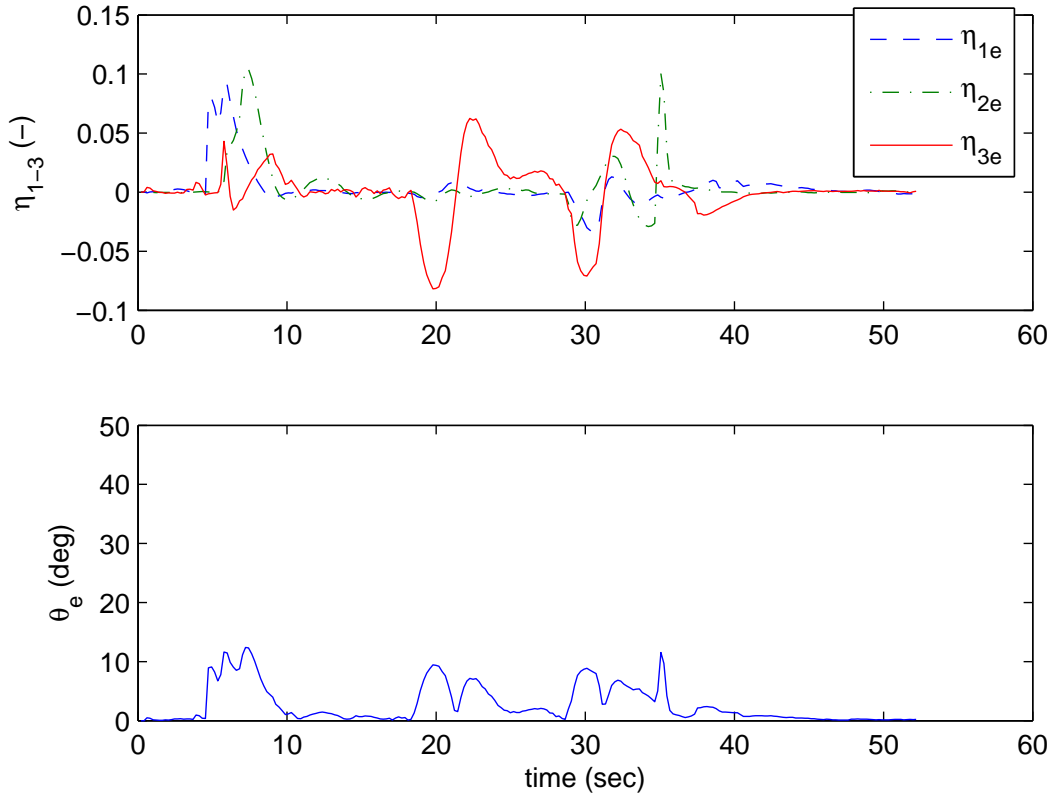


Figure 3.8: Quaternion error measured between the actual and reference model during the simulation test of the backstepping adaptive controller with least-squares parameter estimation is presented.

of a second the true parameter is identified. Similar behavior is exhibited for all other parameters. The level-flight roll moment due to poor trim discussed earlier, can be seen in θ_1 , which changes from roughly 0 to 5 in the transition to level flight. This bias acceleration is also instantly identified. In addition, note the pitch and yaw bias terms during level flight (θ_3 and θ_5). Large transients occur during rotations about the associated axes. These are due to two significant sources of angular acceleration that oppose rotation, namely rotational damping and torque from sideslip in yaw or angle of attack in pitch. These torques, due to limited computational and sensing capabilities, were all lumped into the bias terms. Even these large transients, however, with no information on sideslip angle, angle of attack, or angular rates are tracked exceptionally well.

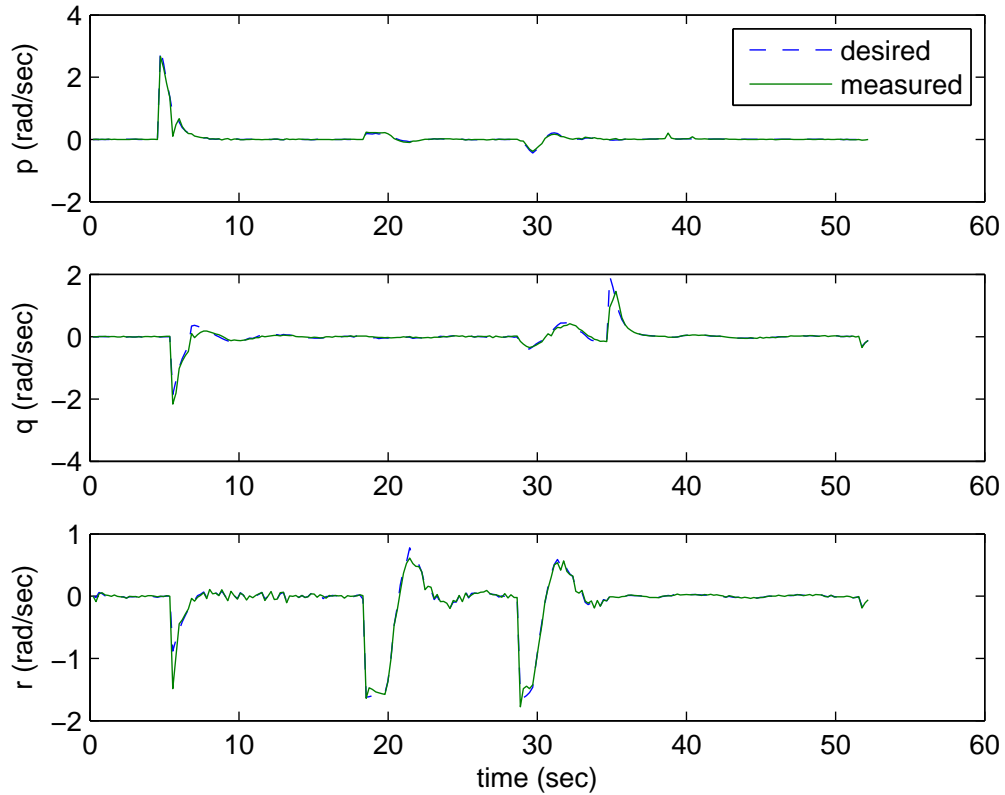


Figure 3.9: This figure gives the angular rates during the attitude simulation test of the backstepping adaptive controller with least-squares parameter estimation.

As a natural result of accurate parameter estimation, good modeling of angular acceleration is achieved. An example of model tracking can be seen in Figure 3.11, which shows a plot of measured and estimated pitch angular accelerations for a 20 second segment of the flight. Note that the transition from level to hover occurs in the middle of this segment to show tracking and typical accelerations during both flight regimes. Error between the two signals is also given in Figure 3.11. Observe that the plot of angular acceleration and angular acceleration error are a good indication of how well the estimated model matches the true dynamics of the system. Acceleration results about the other two axes are comparable. The average absolute error for all three axes is less than 0.5 radians per second squared. This is phenomenal considering that the airspeed measurement squared, with considerable noise, is used

in the estimate of angular acceleration. Also, noise in the measured angular rates is amplified when differentiated for the measurement of angular acceleration.

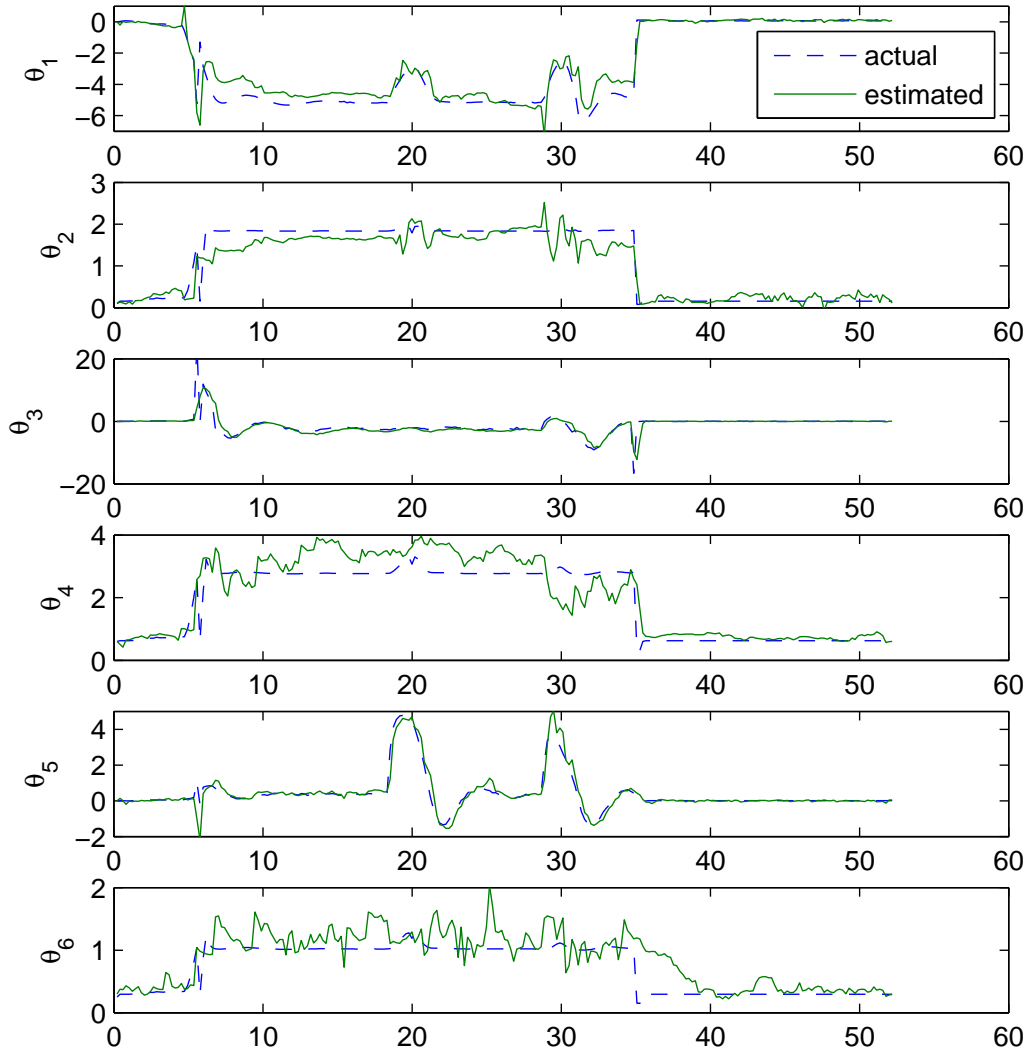


Figure 3.10: Parameter estimation throughout the simulation test of the adaptive backstepping controller with recursive least-squares parameter estimation is shown. Note that θ_1 , θ_2 , and θ_3 are the roll, pitch, and yaw acceleration biases. Also, θ_2 , θ_4 , and θ_6 are aileron, elevator, and rudder effectiveness terms.

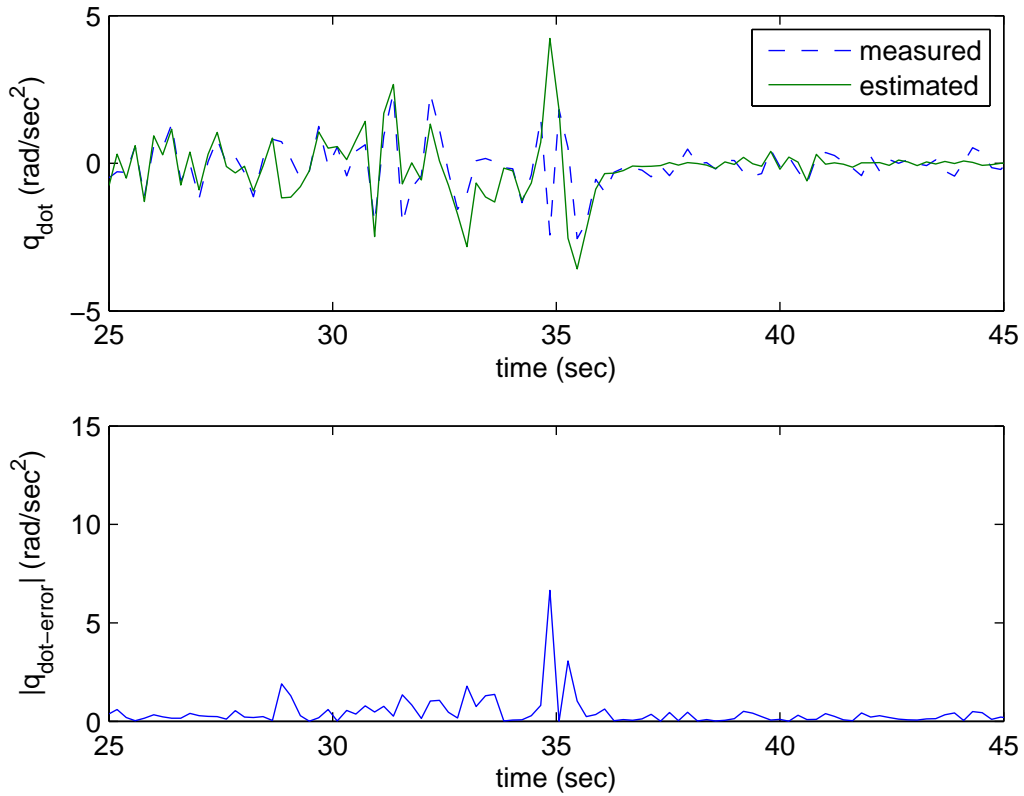


Figure 3.11: Pitch angular acceleration and acceleration error throughout the simulation test of the adaptive backstepping controller with recursive least-squares parameter estimation is shown. Note that the pitch acceleration is shown during only a 20 second segment of the flight. The transition from level to hover flight occurs at 35 seconds in the middle of this plot.

3.3.3 Lyapunov Method

The Lyapunov-based method for parameter estimation was also simulated on the same flight path with the backstepping controller. It was noted during simulation that the system was somewhat susceptible to large gains. Also, instability in the attitude controller would occur if the estimated terms with even subscripts initially were small in magnitude.

Attitude during the simulated flight is shown in Figure 3.12, with error given in Figure 3.13. It can be observed that attitude tracking is considerably better when compared to the gain-scheduled method, yet slightly worse than the results obtained from the least-squares parameter estimation adaptive controller. Angular rates of this method, shown in Figure 3.14, also exhibit good performance.

The parameters estimated with the Lyapunov-based method are shown in Figure 3.15. Interestingly, the reasonable angular rate and attitude performance of the controller is accompanied with poor parameter estimation. This is not surprising, noting that the theory used to derive the estimation method does not guarantee parameter convergence. The update law was chosen for asymptotic stable tracking of the controller. Consider the terms being estimated directly with the Lyapunov method shown in Figure 3.16, namely

$$\begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \end{pmatrix} = \begin{pmatrix} \theta_1/\theta_2 \\ \theta_3/\theta_4 \\ \theta_5/\theta_6 \\ 1/\theta_2 \\ 1/\theta_4 \\ 1/\theta_6 \end{pmatrix}.$$

Tracking of ξ_1 , ξ_2 , and ξ_3 is evident, while the estimates of ξ_4 , ξ_5 , and ξ_6 appear to simply grow in magnitude irregardless of what the true parameters are doing.

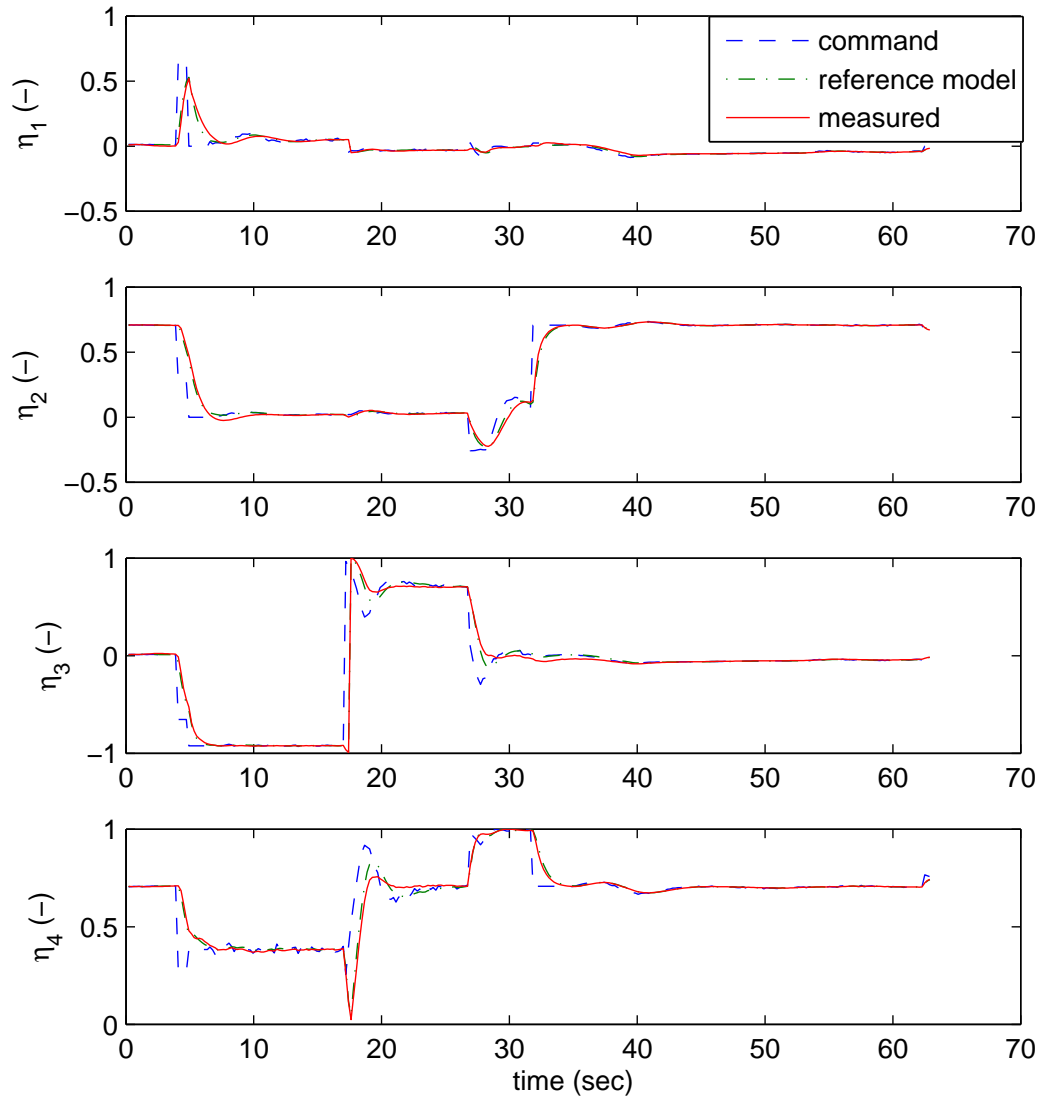


Figure 3.12: Above, attitude results during simulation testing of the backstepping adaptive controller with Lyapunov-based parameter estimation are given.

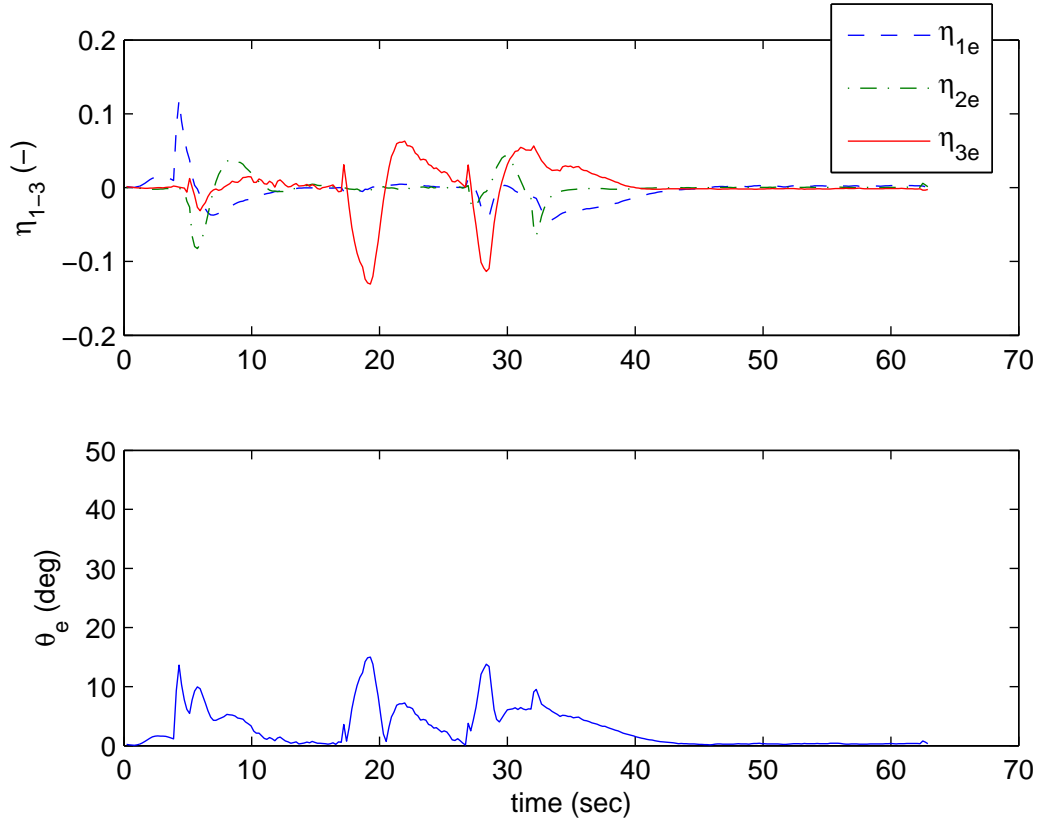


Figure 3.13: Quaternion error measured between the actual and reference model during the simulation test of the backstepping adaptive controller with Lyapunov-based parameter estimation is shown.

Considering equations (3.45), (3.46), and (3.52), the update law of these terms is

$$\begin{pmatrix} \dot{\hat{\xi}}_1 \\ \dot{\hat{\xi}}_2 \\ \dot{\hat{\xi}}_3 \\ \dot{\hat{\xi}}_4 \\ \dot{\hat{\xi}}_5 \\ \dot{\hat{\xi}}_6 \end{pmatrix} = \begin{pmatrix} \bar{V}^2 \gamma_3 \tilde{p} \\ \bar{V}^2 \gamma_3 \tilde{q} \\ \bar{V}^2 \gamma_3 \tilde{r} \\ \gamma_4 \left(k_2 \tilde{p}^2 + \frac{1}{2} \eta_4 \eta_1 \tilde{p} + \dot{p}_d \tilde{p} \right) \\ \gamma_5 \left(k_2 \tilde{q}^2 + \frac{1}{2} \eta_4 \eta_2 \tilde{q} + \dot{q}_d \tilde{q} \right) \\ \gamma_6 \left(k_2 \tilde{r}^2 + \frac{1}{2} \eta_4 \eta_3 \tilde{r} + \dot{r}_d \tilde{r} \right) \end{pmatrix}. \quad (3.56)$$

From further investigation it can be seen that $\dot{\hat{\xi}}_4$, $\dot{\hat{\xi}}_5$, and $\dot{\hat{\xi}}_6$ are dominated with the terms $k_2 \tilde{p}^2$, $k_2 \tilde{q}^2$, and $k_2 \tilde{r}^2$, which will always be positive. This results in a consistent increase of $\hat{\xi}_4$, $\hat{\xi}_5$, and $\hat{\xi}_6$ and consequently, a consistent decrease of $\hat{\theta}_2$, $\hat{\theta}_4$, and $\hat{\theta}_6$.

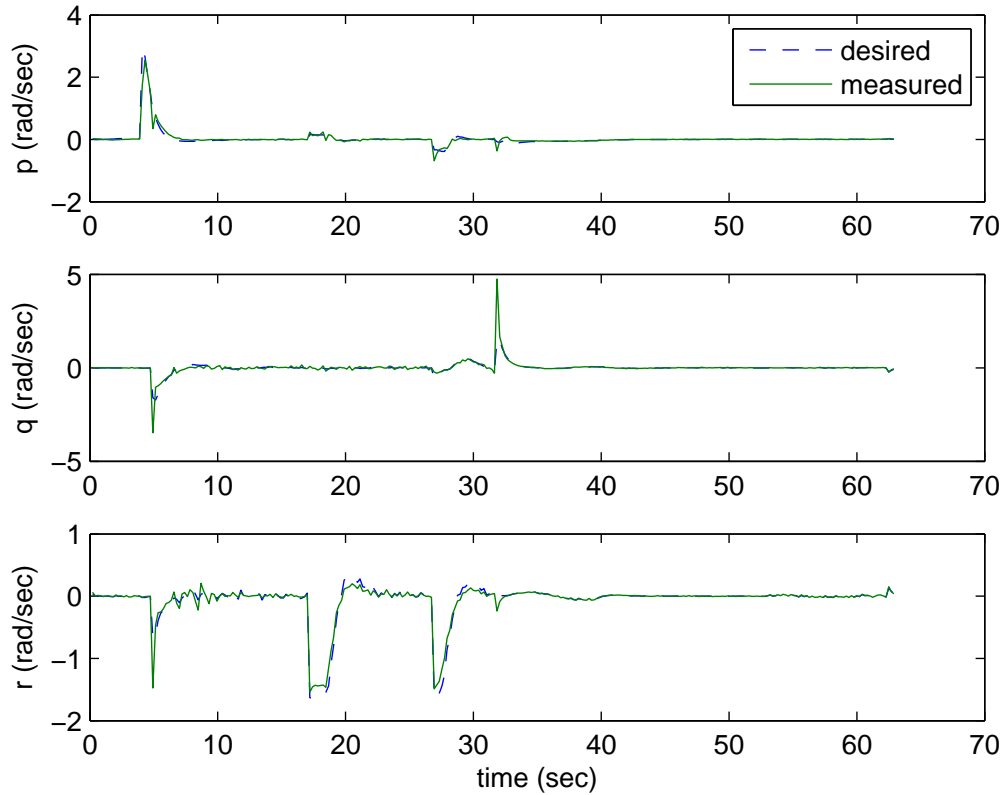


Figure 3.14: In this figure the angular rates during the attitude simulation test of the backstepping adaptive controller with Lyapunov-based parameter estimation are exhibited.

Because of this behavior the system is prone to instability. When $\hat{\theta}_2$, $\hat{\theta}_4$, and $\hat{\theta}_6$ become relatively smaller than the true parameters, high frequency oscillations can occur. This is due to an increase in actuator output, when compared to the intended. As a result, this behavior causes the $\hat{\theta}$ terms with even subscripts to decrease smaller yet from angular rate error, destabilizing the system. With small gains, such as the ones used in this simulation, the absence of such behavior can be prolonged.

An example of model tracking is given in Figure 3.17, where a small segment of hover and level flight during the experiment is presented. This plot of angular acceleration shows poor tracking in both modes of flight. Note that the transition occurs at about 32 seconds. It can be seen that during level flight, when the effectiveness of the elevator is underestimated (see Figure 3.15), large high-frequency oscillations

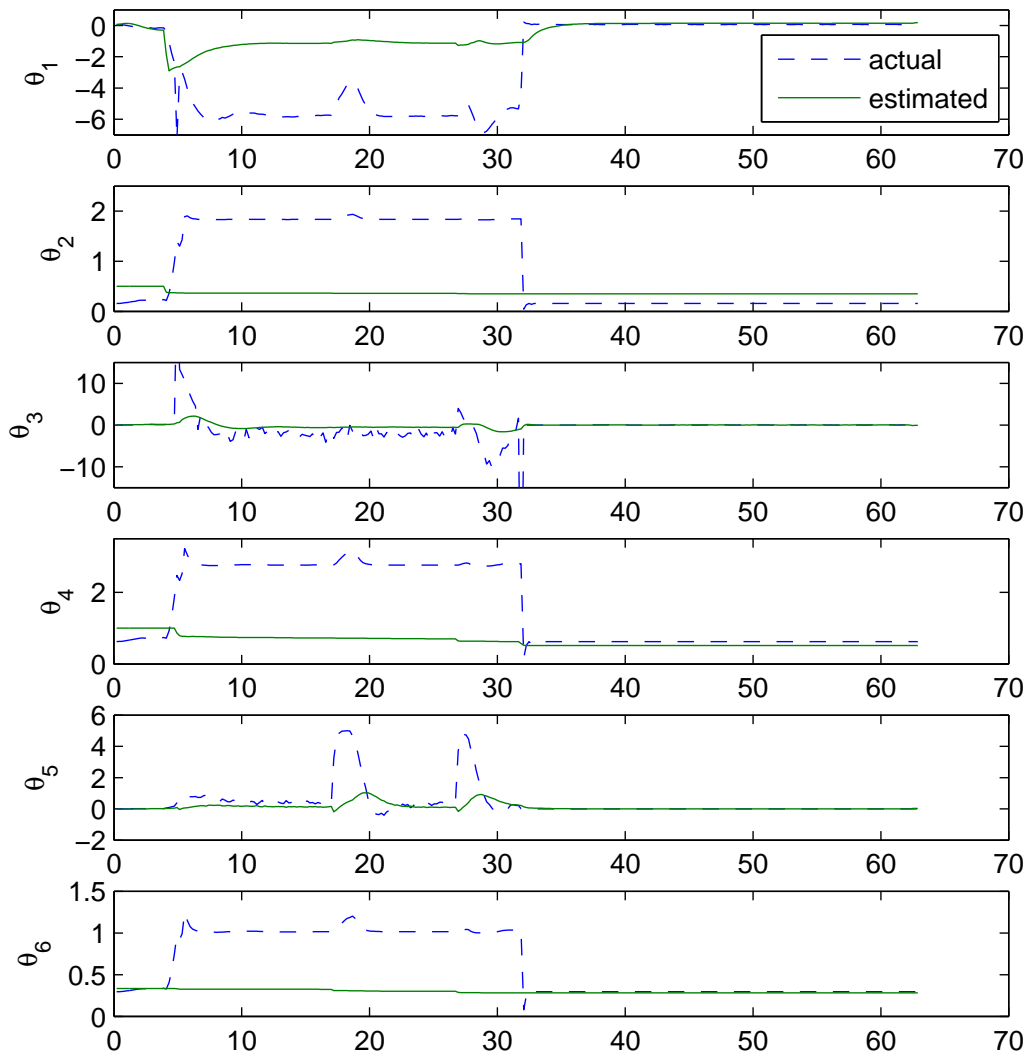


Figure 3.15: Parameter estimation throughout the simulation test of the adaptive backstepping controller with Lyapunov-based parameter estimation is shown. Note that θ_1 , θ_2 , and θ_3 are the roll, pitch, and yaw acceleration biases. Also, θ_2 , θ_4 , and θ_6 are aileron, elevator, and rudder effectiveness terms.

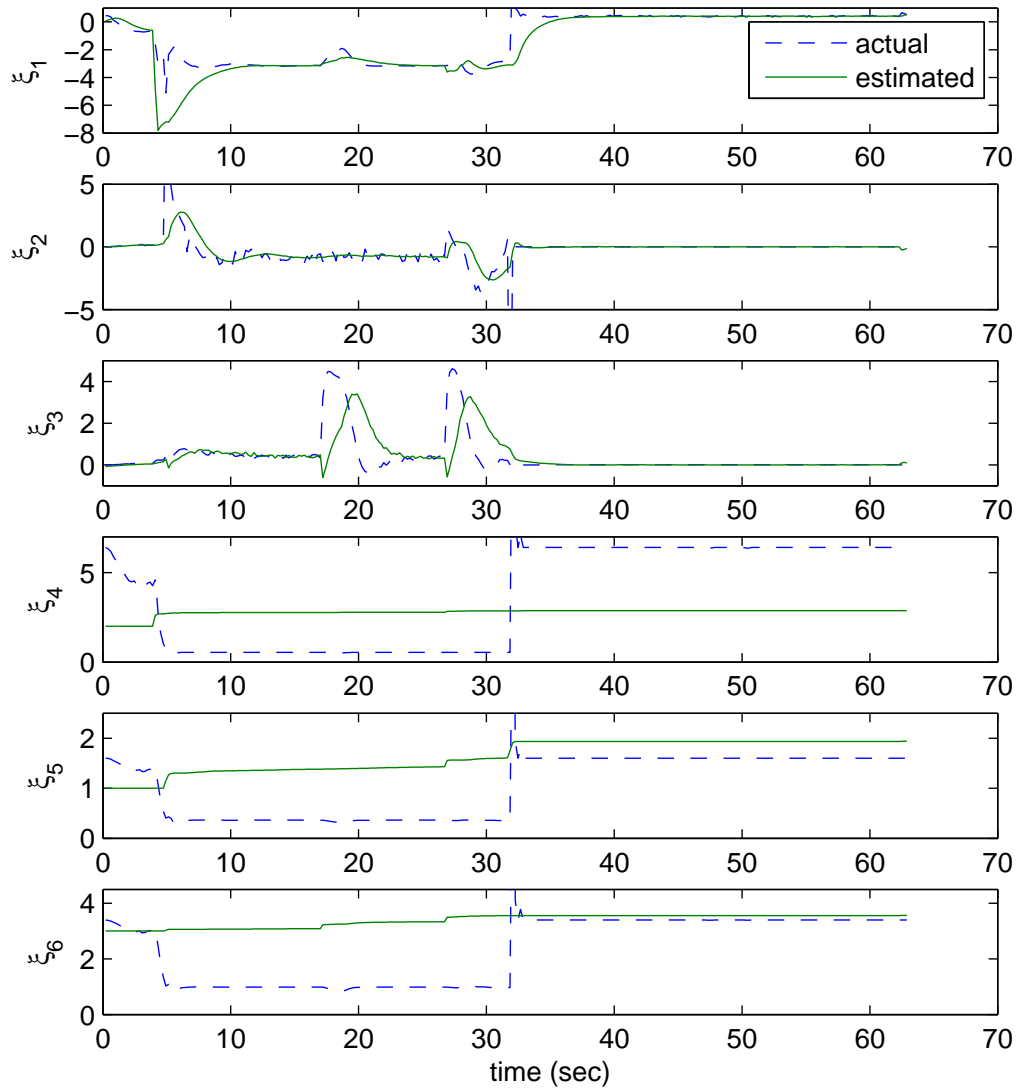


Figure 3.16: This figure gives the parameters estimated directly with the Lyapunov-based method during the simulation test of adaptive backstepping controller.

in pitch acceleration occur. The parameter associated with elevator effectiveness ($\hat{\theta}_4$) still decreases despite this. With a small update gain, the estimated parameter decreases slowly, prolonging instability. However, even with small gains this system destabilizes in simulation if the flight time is extended. This discussion explains why small initial values for estimated actuator effectiveness terms (the $\hat{\theta}$ parameters with even subscripts) and large gains both lead to instability.

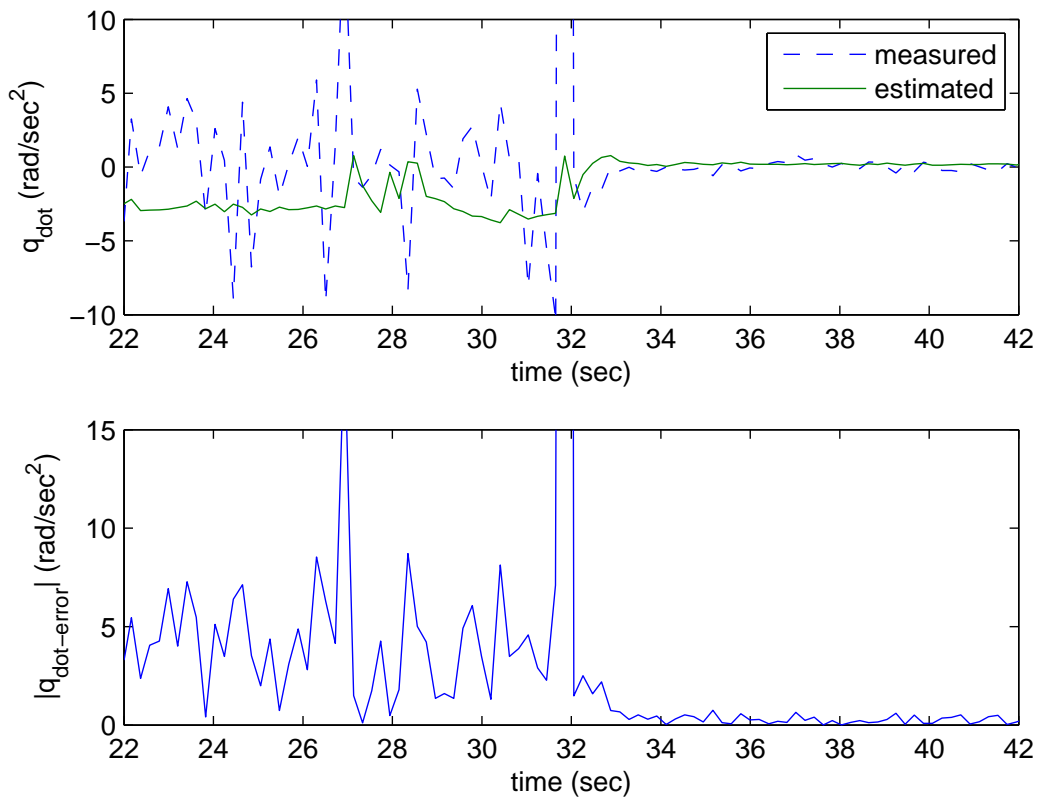


Figure 3.17: Pitch angular acceleration and acceleration error throughout the simulation test of the adaptive backstepping controller with Lyapunov-based parameter estimation is shown. Note that pitch acceleration is shown during only a 20 second segment of the flight. The transition from level to hover flight occurs at 32 seconds in the middle of this plot.

3.4 Hardware Results

Hardware results of the baseline gain-scheduled and adaptive backstepping with stabilized recursive least-squares parameter estimation controllers are presented in this section. Only the adaptive least-squares method was tested in hardware. This was because the performance of the least-squares method was better in both attitude control and angular acceleration estimation, and also because of the instability of the Lyapunov-based method. A similar flight path was flown during hardware testing with commands in the following order: a vertical takeoff, a hover waypoint, two level flight waypoints, a hover waypoint, and a hover land. An example of this path can be seen in Figure 3.18. The adaptive controller performance is shown to be slightly better than the baseline method. In results that will be presented, it can be seen that in particular circumstances both controllers lose control of the aircraft for brief moments due to poor airframe design, which will be discussed in further detail. Desired attitude and throttle were given from the methods discussed in Chapter 4 and actual attitude was estimated with the scheme presented in Chapter 5.

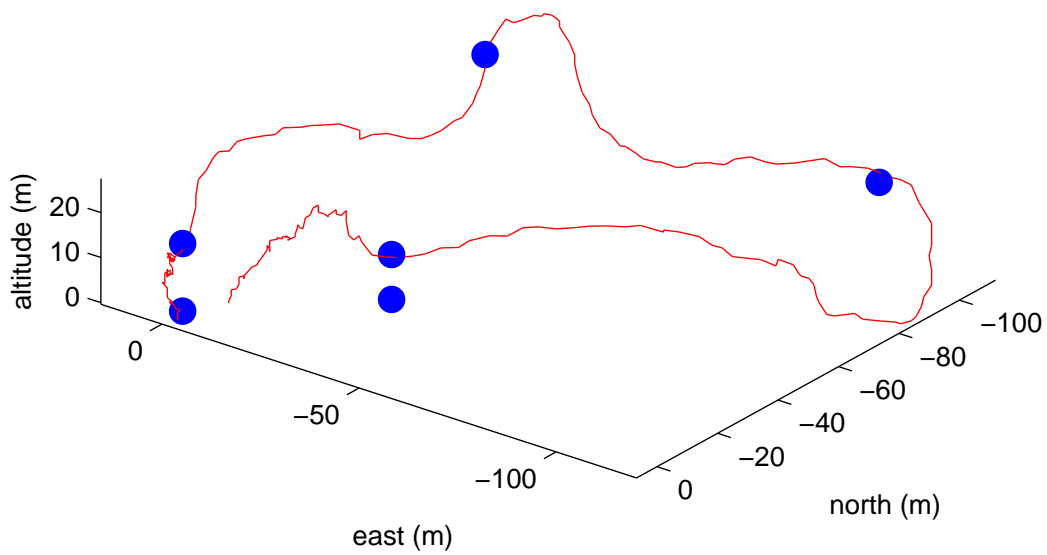


Figure 3.18: Above, the waypoint path selected for the attitude control hardware experiment is shown. A vertical takeoff was performed at 0 north and 0 east. This was followed by a hover waypoint, two level waypoints, another hover waypoint, and finally a hover land. This flight was executed with the adaptive backstepping controller and recursive least-squares parameter estimation.

3.4.1 Gain-Scheduled Quaternion Feedback Attitude Control

Baseline gain-scheduled PID quaternion feedback attitude control results are given in this section. Gains for this controller were tuned empirically from observations of flight performance. Significant time and effort was put forth to improve the controller's performance to give a reasonable representation of the method's capabilities.

Attitude logged during the baseline attitude experiment can be seen in Figure 3.19, and error is shown in Figure 3.20. Although not as impressive as simulation results, the attitude performance of the system throughout the flight is reasonably good. Note that because the hover-to-level transition and the first level flight path occur in a south direction, the fourth element of the quaternions approach zero. A value of zero for the fourth element of a quaternion indicates a rotation of 180 degrees, which explains the symmetric behavior of the second and third elements in the figure. Relatively smooth transitions between gains can be seen as the aircraft transitions from hover to level flight at about 15 seconds and back from level to hover flight at around 51 seconds. The scheduling of gains by airspeed and propeller wash airspeed squared can be seen to be effective throughout the flight. At the end of the flight, large spikes in error are shown, particularly in the roll or x -axis. This is during the descent-to-land portion of the flight. During a descent, air begins to flow in the reverse direction over the control surfaces. Moreover, throttle which provides the propeller wash airflow is reduced for the descent. Both of these contribute to stagnation of air over the control surfaces, rendering them ineffective. This same behavior can be seen with the adaptive controller. With improvements to the airframe design, this problem most likely could be remedied by decreasing the distance between the propeller and the control surfaces.

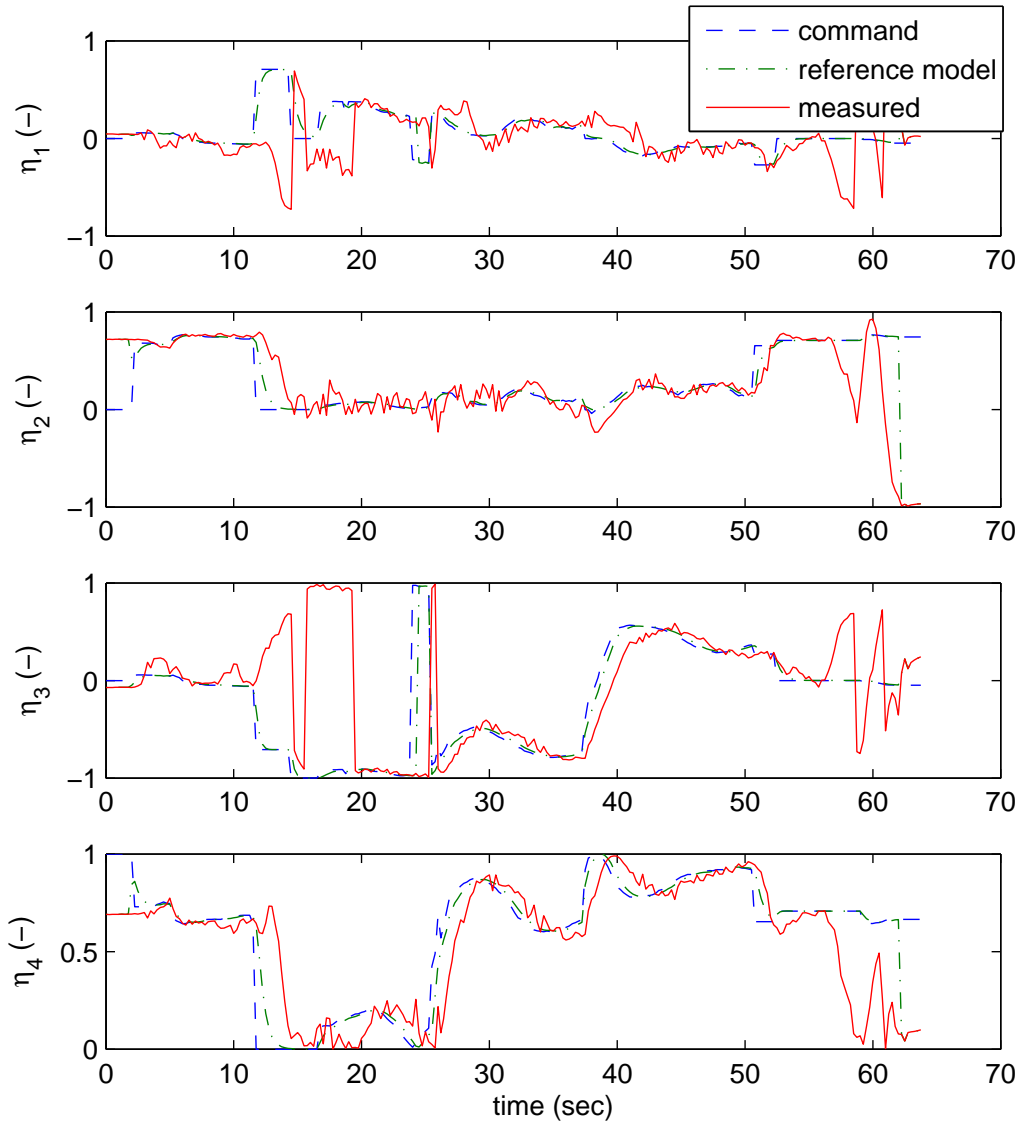


Figure 3.19: Attitude during the hardware test of the PID gain-scheduled quaternion feedback controller is shown. Note that because the hover-to-level transition and the first level flight path occur in a south direction, the fourth element of the quaternions approach zero. This explains the symmetric behavior of the second and third elements.

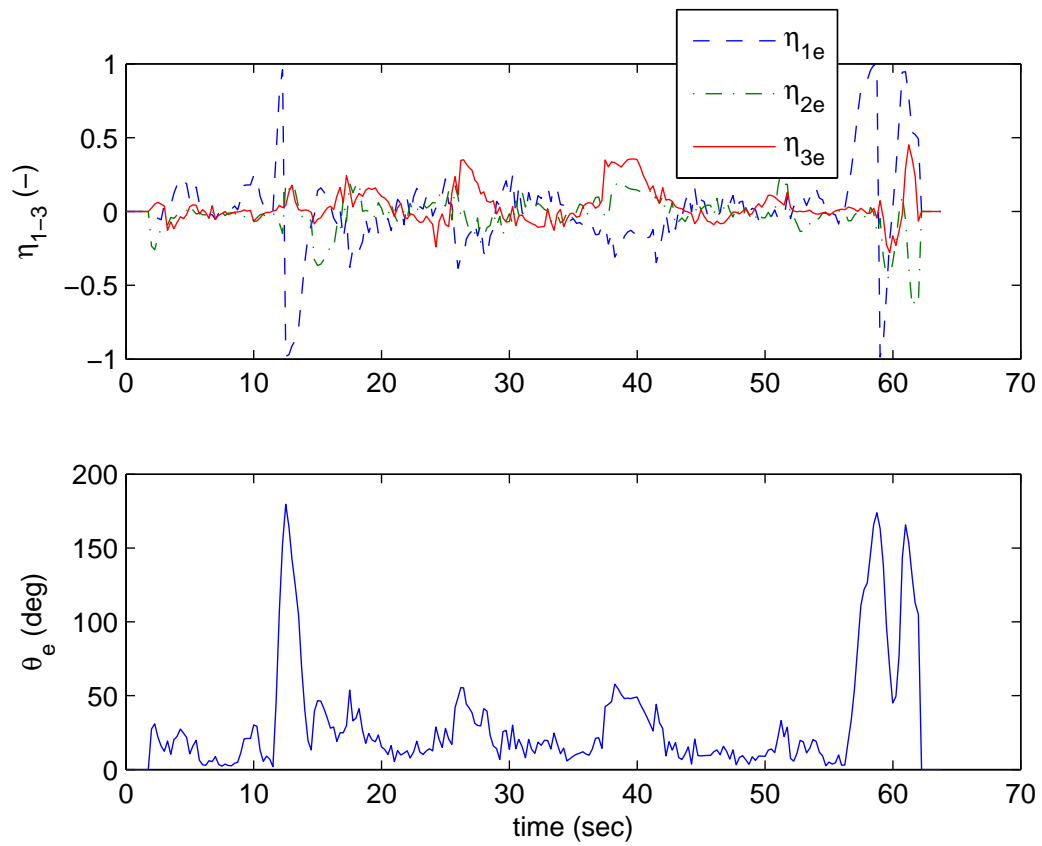


Figure 3.20: This figure presents the attitude error during the hardware test of the PID gain-scheduled quaternion feedback controller.

3.4.2 Stabilized Recursive Least Squares With Data Forgetting

Adaptive backstepping hardware performance is discussed in this section. The stabilized recursive least-squares algorithm with data forgetting was implemented for parameter estimation and model learning. Previously obtained data (airspeed, propeller wash airspeed, angular rates, and control surface actuator settings) was used to tune the least-squares algorithm off-line. The two backstepping controller gains were then adjusted during flight testing from observations of attitude and angular rate tracking.

Attitude results acquired during the hardware test flight are shown in Figure 3.21, where transitions occur at about 15 and 50 seconds. The same symmetric behavior of the second and third quaternion elements in this figure is apparent when the respective fourth element approaches zero. The quaternion attitude error during the experiment is located in Figure 3.22. Results are shown to be quite similar to, and in some ways, better than the baseline controller. In level flight, error acquired by the adaptive method is slightly less than that obtained by the gain-scheduled controller. No large spikes of error occur during the first transition, which can be seen in the results of the baseline method. The lack of an increase in error during transitions indicates smooth and fast adaptation. The same loss of control from air stagnation that was discussed in the previous section, indicated by large spikes of error during the descent, are also apparent in these results.

Angular rates throughout the flight are available in Figure 3.23. As stated earlier, it is through the tracking of desired angular rates that attitude is controlled with the backstepping method. Reasonable tracking with considerable oscillations can be seen. This implies that the rate gain (k_2) is possibly too large. Also, areas where desired rates are persistently larger than actual rates indicate that the controller cannot achieve the desired rate performance, denoting that the attitude gain (k_1) is too large as well.

On-line parameter estimation performed during the flight is shown in Figure 3.24. Results appear to be similar to those presented from the simulation of the estimation algorithm. The first thing to note is that adaptation, like in simulation,

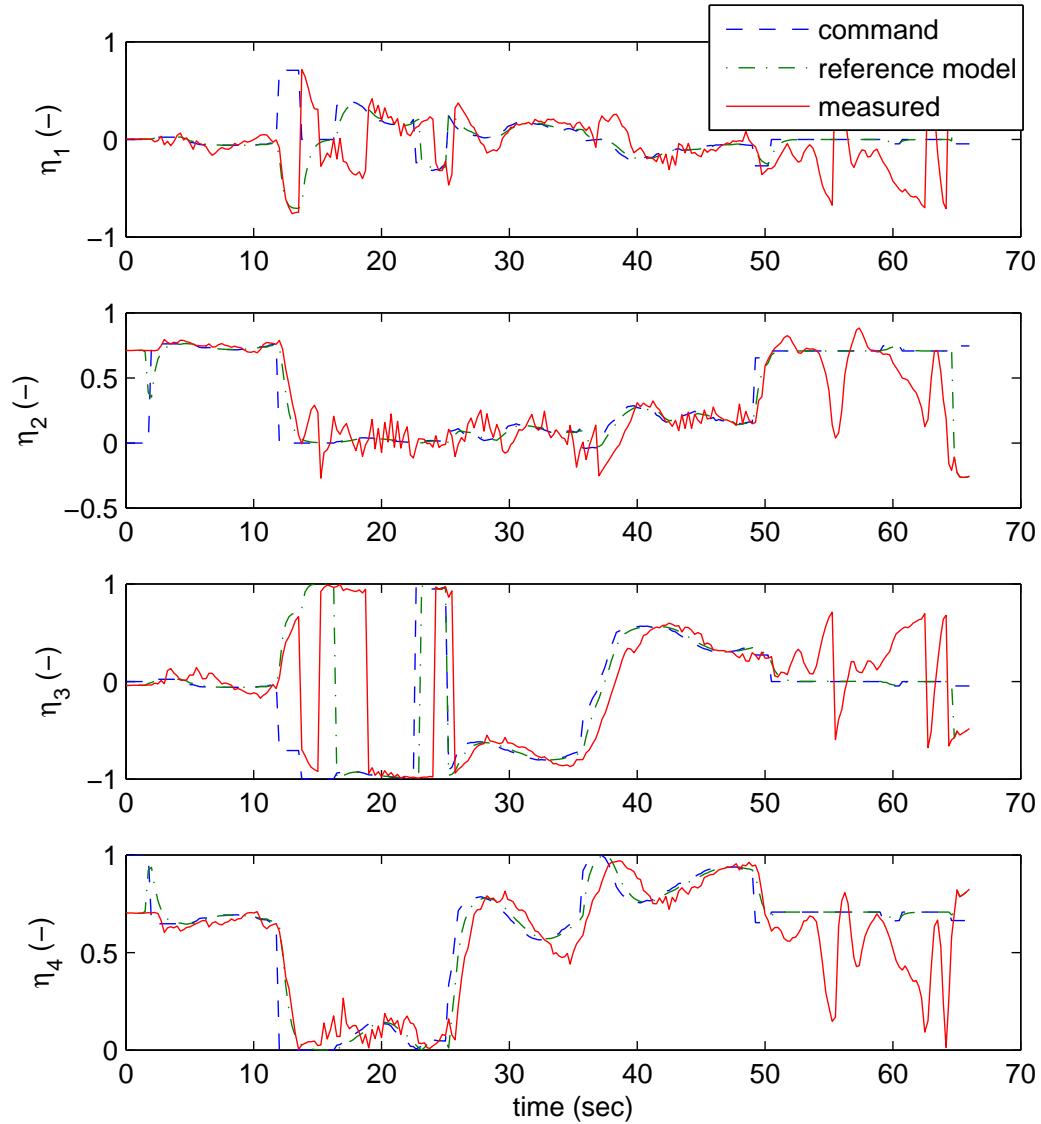


Figure 3.21: Attitude results during hardware flight testing of the backstepping adaptive controller with least-squares parameter estimation are given. Note that because the hover-to-level transition and the first level flight path occur in a south direction, the fourth element of the quaternions approach zero. This explains the symmetric behavior of the second and third elements.

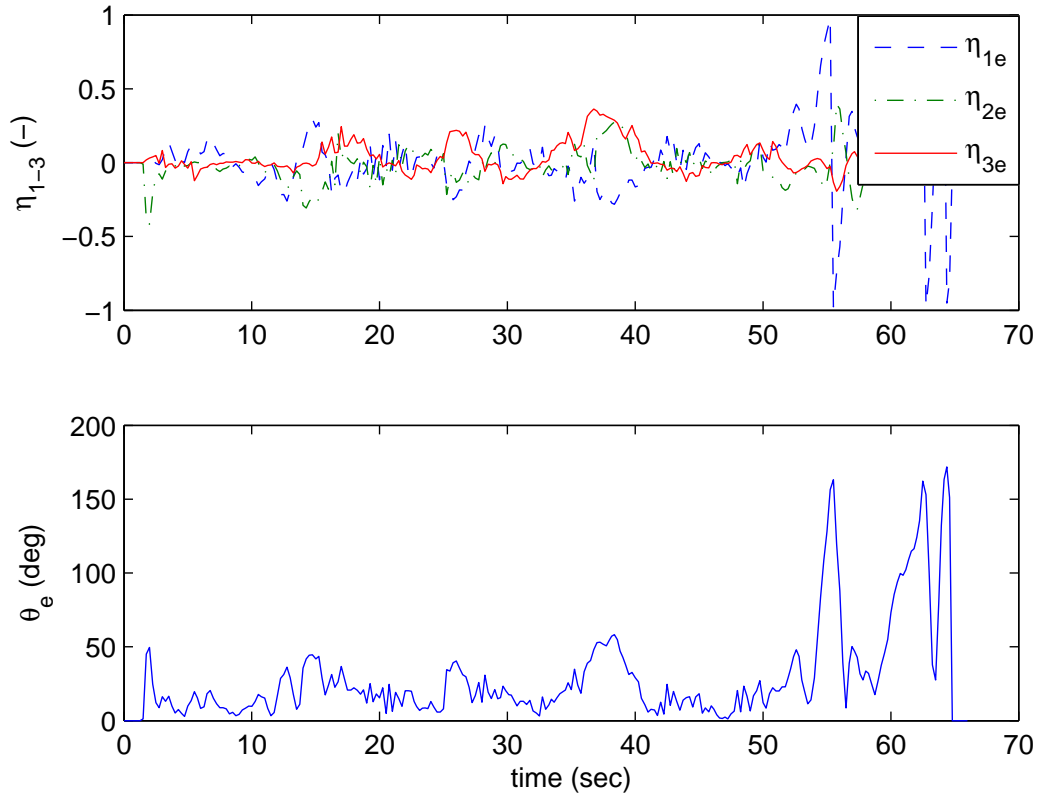


Figure 3.22: This figure gives the quaternion error measured between the actual and reference model during the hardware flight test of the backstepping adaptive controller with least-squares parameter estimation.

occurs quite fast. Consider for example $\hat{\theta}_2$, $\hat{\theta}_4$, and $\hat{\theta}_6$, noting that transitions to level and hover flight occur at close to 15 and 50 seconds, it can be seen that estimation of control surface effectiveness for the aileron, elevator, and rudder change drastically, in some cases almost instantaneously. Observe also that as expected, the actuator effectiveness terms are all relatively small in hover when compared to level flight. Because the propeller wash airflow in hover is significantly less than airflow in level flight, the control surface effectiveness is reduced. Level-flight bias angular acceleration due to poor aileron trim, which is identified quickly after the transition, can be seen as the nominal value of θ_1 changes from roughly 0 to 10. This term reduces back to roughly zero after the transition to hover, indicating that aileron trim is poor in level flight only. Accelerations from sideslip and angular yaw rate damping are shown

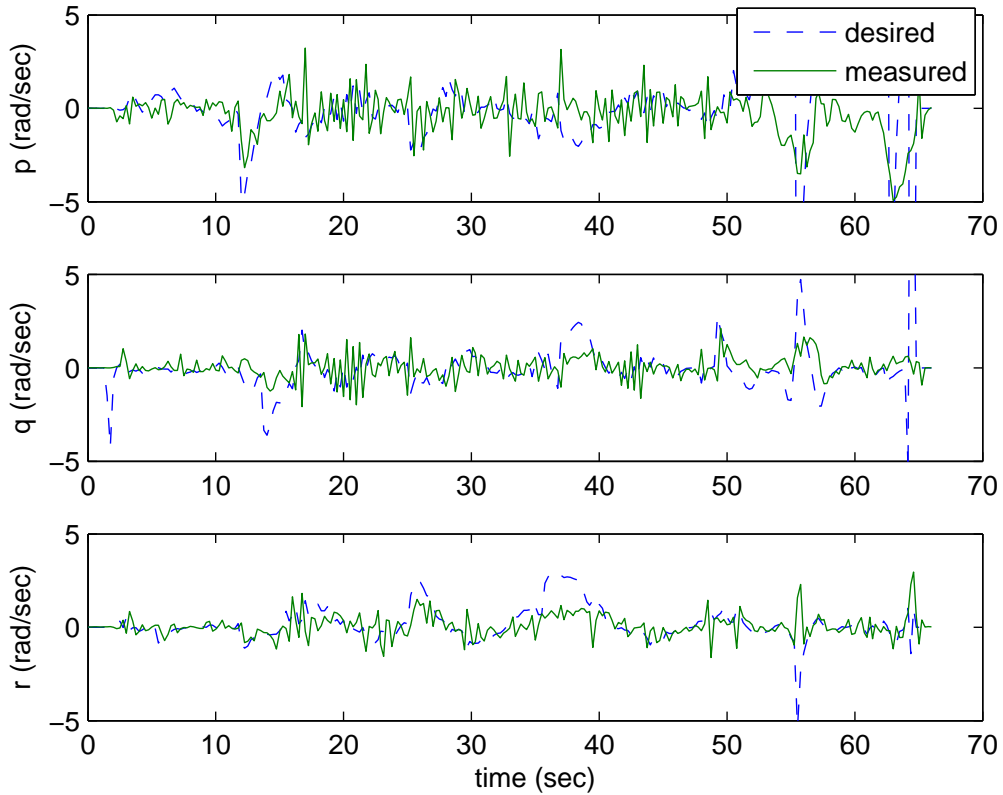


Figure 3.23: Angular rates during the attitude hardware flight test of the backstepping adaptive controller with least-squares parameter estimation are presented.

in the estimation of θ_5 . It can be seen that as expected in right hand turns, negative angular acceleration opposing rotation is present. These turns occur at around 15, 25, and 35 seconds as noted from positive yaw rates in Figure 3.23.

Good angular pitch acceleration tracking is presented in Figure 3.25. The plot of measured and estimated acceleration shows excellent modeling during both level and hover flight. Observe that the transition occurs at about 51 seconds. Pitch acceleration modeling error is 3.7 rad/sec^2 on average throughout the flight. This is relatively small considering that the rate gyros with noise are differentiated for the angular acceleration measurement, delay in the actuator output is present, and the noisy velocity measurement squared is used in the estimation of angular acceleration.

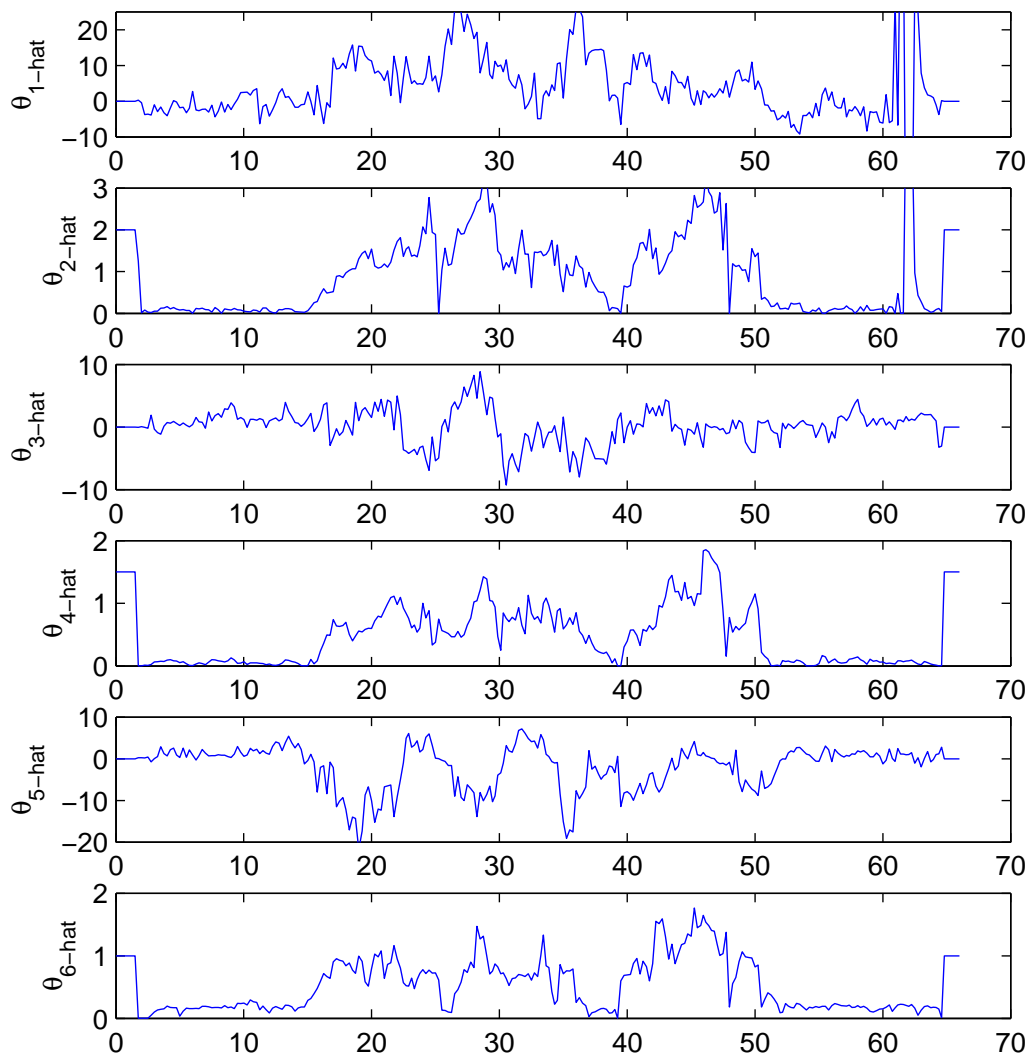


Figure 3.24: Parameter estimation throughout the hardware test of the adaptive backstepping controller with recursive least-squares parameter estimation is shown. Note that $\hat{\theta}_1$, $\hat{\theta}_2$, and $\hat{\theta}_3$ are the roll, pitch, and yaw acceleration biases. Also, $\hat{\theta}_2$, $\hat{\theta}_4$, and $\hat{\theta}_6$ are aileron, elevator, and rudder effectiveness terms.

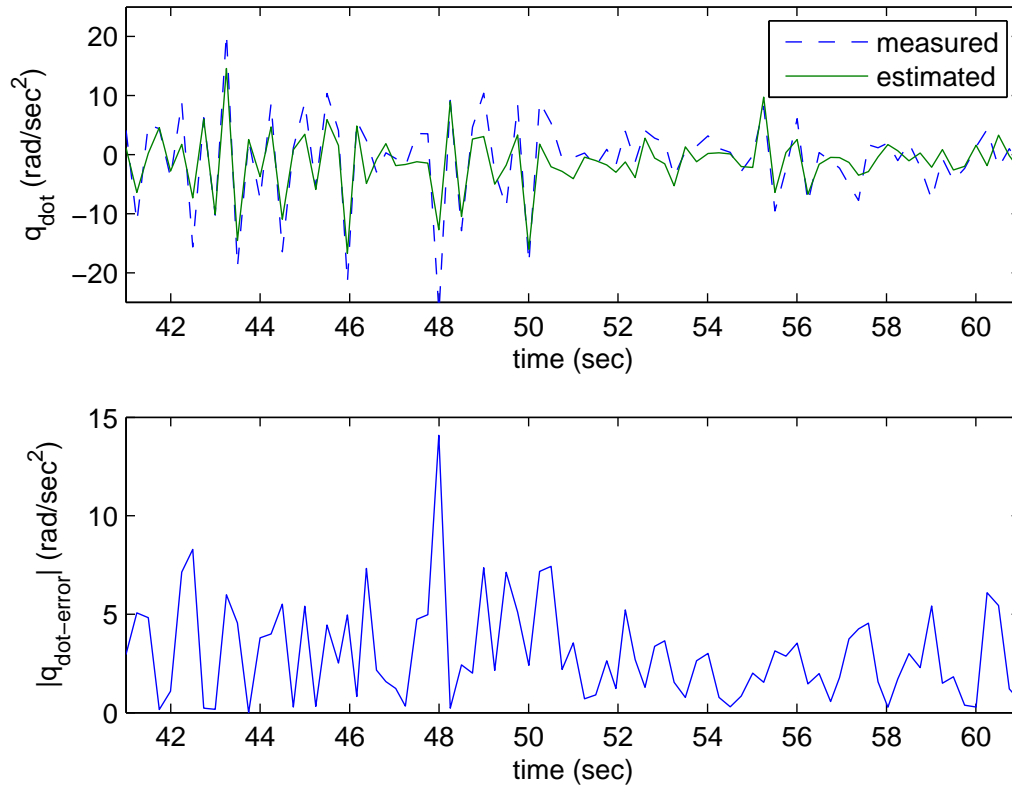


Figure 3.25: Pitch angular acceleration and acceleration error throughout the hardware test of the adaptive backstepping controller with recursive least-squares parameter estimation is shown. Note that the pitch acceleration is shown during only a 20 second segment of the flight. The transition from level to hover flight occurs at 51 seconds in the middle of this plot.

3.5 Summary

In Chapter 3, methods of tailsitter attitude control were discussed. The design of a standard gain-scheduled PID controller for the determination of baseline performance was given. An effective quaternion backstepping technique for model cancellation and stable consistent tracking of reference model performance was also derived. This method was used in conjunction with two algorithms that were also developed for on-line parameter estimation. It was shown that with least-squares parameter estimation the adaptive attitude controller was slightly better than the baseline method in both simulation and hardware.

To obtain attitude control results, navigational control for tailsitter maneuvers was required. Derivation of such methods will be discussed in the following chapter (Chapter 4). This includes hover north/east position and altitude control, level flight path following and altitude/airspeed control, and transition between both modes of flight.

Chapter 4

Navigational Control

Tailsitter attitude control requires attitude trajectories to navigate maneuvers such as hover position tracking with altitude hold, level flight path following with altitude and airspeed control, and transitions between the two modes. Development of controllers to accomplish this will be discussed in this chapter. Considering Figure 1.2, both a desired quaternion and throttle setting are required from the navigational controller. Note that these methods have not been developed for optimality, but to adequately perform their desired function.

4.1 Hover Position Control

Hover north/east position control will be discussed in this section. The PID feedback loop structure will be used to determine how to tilt the aircraft in hover for position tracking. Simulation and hardware results will be presented for method verification.

4.1.1 Controller Derivation

For position control in hover, the obvious solution is to tilt the aircraft from a vertical orientation in the direction of the desired position. Proportional feedback control can be used to perform this task. For position damping and disturbance rejection, derivative and integral control can also be applied.

The nominal vertical orientation that has been chosen from which to rotate is

$$\bar{\eta}_v = \begin{pmatrix} 0 \\ \sqrt{2}/2 \\ 0 \\ \sqrt{2}/2 \end{pmatrix}. \quad (4.1)$$

This quaternion will be referred to as the vertical quaternion. The orientation can be understood as the aircraft nose pointing up along the negative inertial z -axis and the underside pointing north along the inertial x -axis (Figure 4.1).

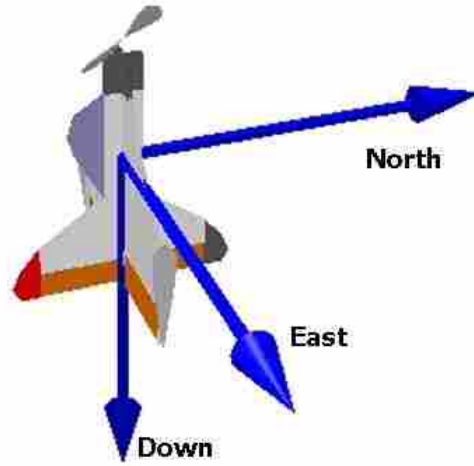


Figure 4.1: Above is a diagram that shows the vertical quaternion orientation defined in this chapter.

Using composition, a desired quaternion can be written as

$$\bar{\eta}_d = \bar{\eta}_c \otimes \bar{\eta}_v. \quad (4.2)$$

The correction quaternion ($\bar{\eta}_c$) is composed with the vertical quaternion to result in the desired orientation. To perform feedback control the correction quaternion is

formed from composite rotations or quaternions:

$$\bar{\eta}_c = \bar{\eta}_{cd} \otimes (\bar{\eta}_{ci} \otimes \bar{\eta}_{cp}). \quad (4.3)$$

The three composite correction quaternions, namely $\bar{\eta}_{cp}$, $\bar{\eta}_{ci}$, and $\bar{\eta}_{cd}$, are constructed to produce proportional, integral, and derivative feedback control, respectively.

For proportional control, the angle of rotation is obtained from the norm of the position error

$$\theta_{cp} = k_p \sqrt{x_e^2 + y_e^2}, \quad (4.4)$$

where k_p is a gain and $(x_e, y_e, 0)^T$ is the inertial north/east positional error vector. Given θ_{cp} , the normalized axis of rotation, expressed in the vertical quaternion reference frame, that tilts the aircraft in the direction of the desired position is

$$\hat{\eta}_{cp} = \begin{pmatrix} 0 \\ \frac{y_e}{\sqrt{x_e^2 + y_e^2}} \\ \frac{x_e}{\sqrt{x_e^2 + y_e^2}} \end{pmatrix} \times \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}. \quad (4.5)$$

As shown, position error expressed in the vertical quaternion reference frame is crossed with the vertical quaternion reference frame negative x -axis. These two terms (θ_{cp} and $\hat{\eta}_{cp}$) define the proportional control correction quaternion. In a similar manner, the integral and derivative control correction quaternions can be defined by

$$\theta_{ci} = k_i \sqrt{\left(\int x_e dt\right)^2 + \left(\int y_e dt\right)^2}, \quad (4.6)$$

$$\theta_{cd} = k_d \sqrt{\dot{x}^2 + \dot{y}^2}, \quad (4.7)$$

$$\hat{\eta}_{ci} = \begin{pmatrix} 0 \\ \frac{\int y_e dt}{\sqrt{(\int x_e dt)^2 + (\int y_e dt)^2}} \\ \frac{\int x_e dt}{\sqrt{(\int x_e dt)^2 + (\int y_e dt)^2}} \end{pmatrix} \times \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \quad (4.8)$$

and

$$\hat{\eta}_{cd} = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ \frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \end{pmatrix}. \quad (4.9)$$

4.1.2 Simulation Results

Simulation of the hover position controller shows that the method is effective at tracking a desired north/east input. The experiment performed is a step input from a takeoff position ($x = 0.0$, $y = 0.0$) to a northwest location ($x = 30.0$, $y = -10.0$). Only proportional control was applied because given the available sensors, position rate information is poor in hover. Also, integral control with inadequate damping is prone to instability and overshoot. Position results of the controller during the experiment are shown in Figure 4.2. The system steps to and tracks the desired position with light oscillations due to limited drag damping, which is small at low airspeeds.

The desired or command attitude of the system generated from the controller is shown in Figure 4.3. Note that only small rotations from vertical are required for position tracking, and good attitude tracking facilitates the success of the position controller.

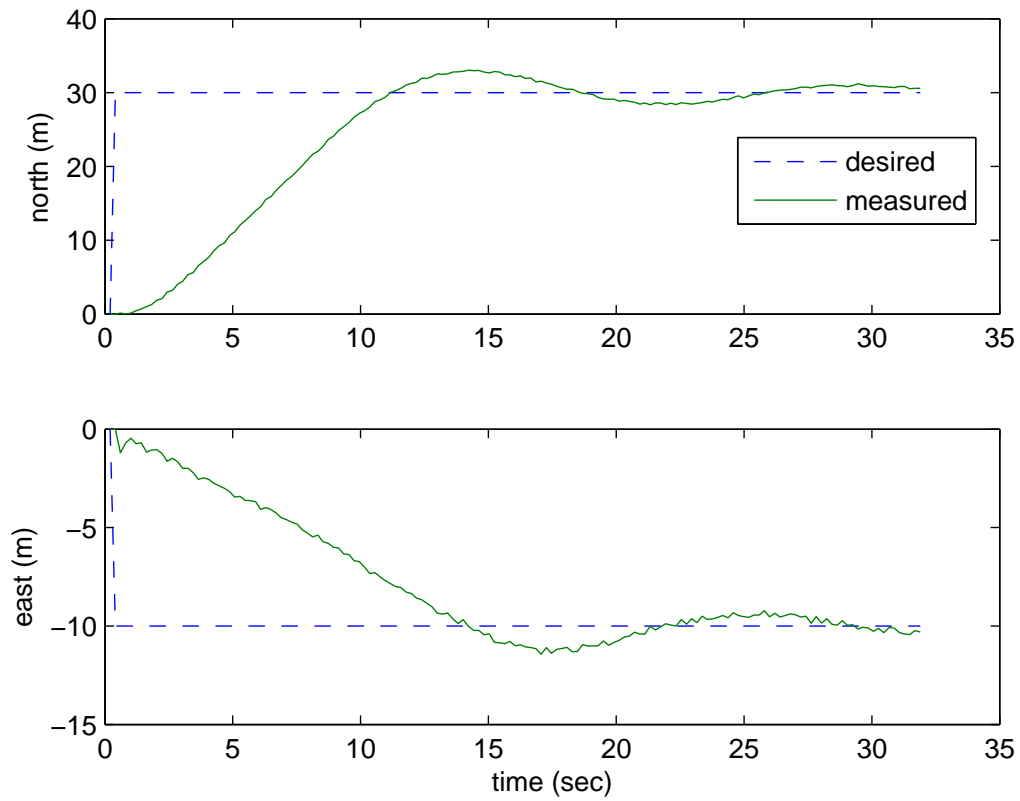


Figure 4.2: This figure gives position tracking of the hover position controller in simulation. A north/east step input from a takeoff position ($x = 0.0$, $y = 0.0$) to a northwest location ($x = 30.0$, $y = -10.0$) is applied.

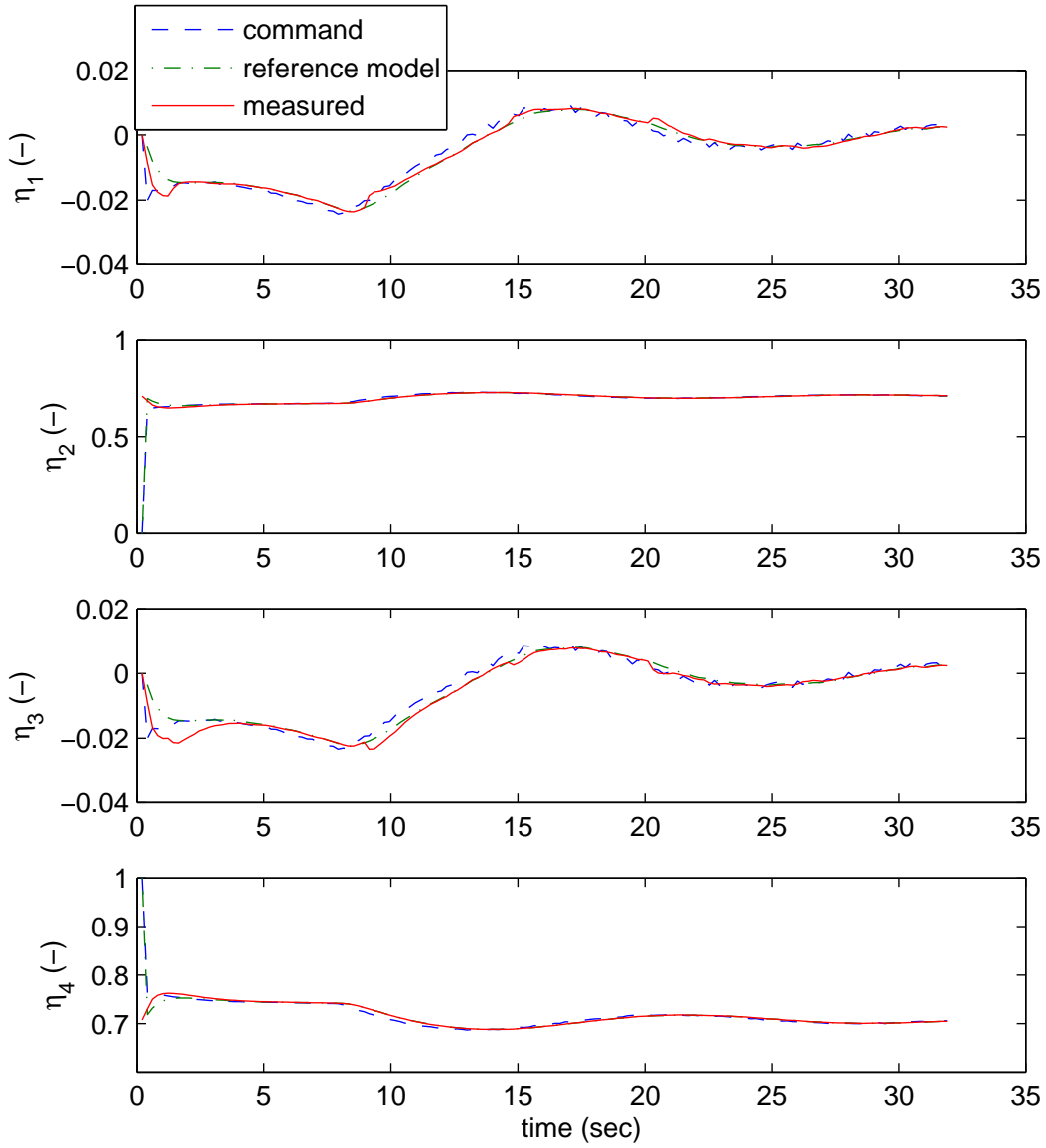


Figure 4.3: Attitude control during the hover north/east position tracking experiment in simulation are shown.

4.1.3 Hardware Results

A similar experiment was performed in hardware on the tailsitter with a step input from takeoff ($x = 0.0, y = 0.0$) to a northwest position ($x = 30.0, y = -10.0$). As with the simulation experiment, only proportional control was implemented due to poor positional rate information and limited positional damping. A rotation limit from vertical was also applied to ensure that the aircraft did not tilt too far from vertical. Position results during the experiment are shown in Figure 4.4. Translation to the desired location and stable tracking of the point (within 5 meters in both directions) for a considerable amount of time verifies the controller methodology.

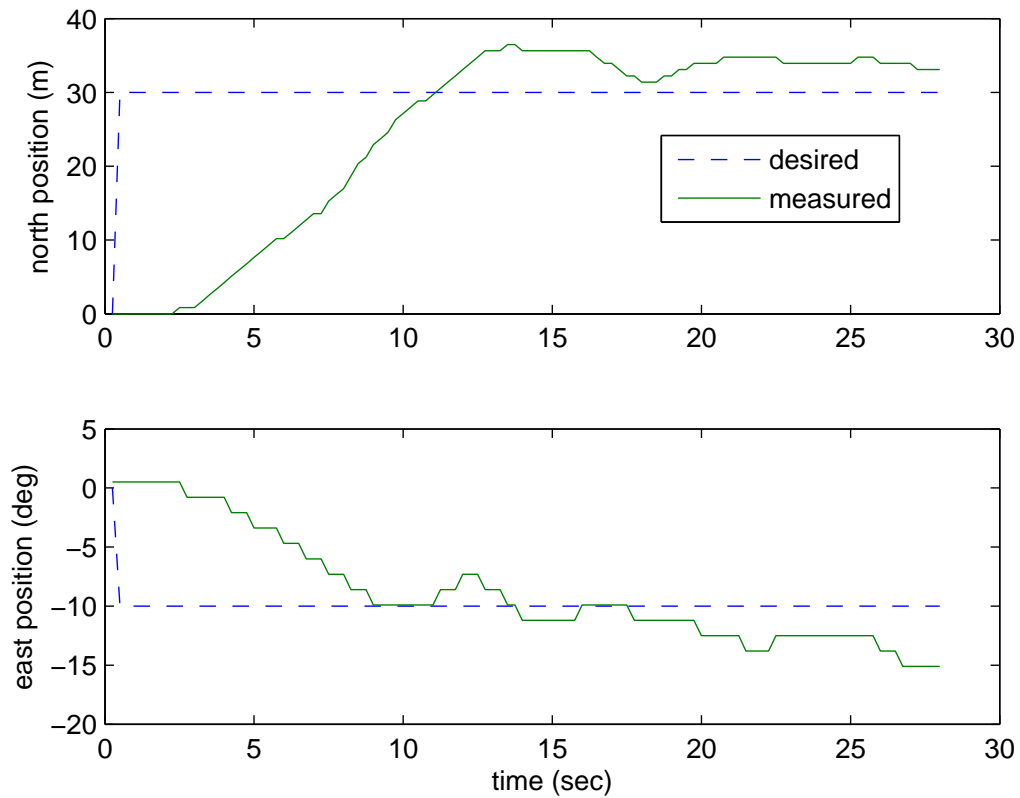


Figure 4.4: Position tracking of the hover position controller in hardware is shown. A north/east step input from a takeoff position ($x = 0.0, y = 0.0$) to a northwest location ($x = 30.0, y = -10.0$) is applied.

The attitude results of the same experiment are shown in Figure 4.5. Attitude tracking in the hardware results is not as impressive as results given from simulation. Despite this, by tracking the desired quaternion, the tailsitter location performance was acceptable, verifying the technique derived.

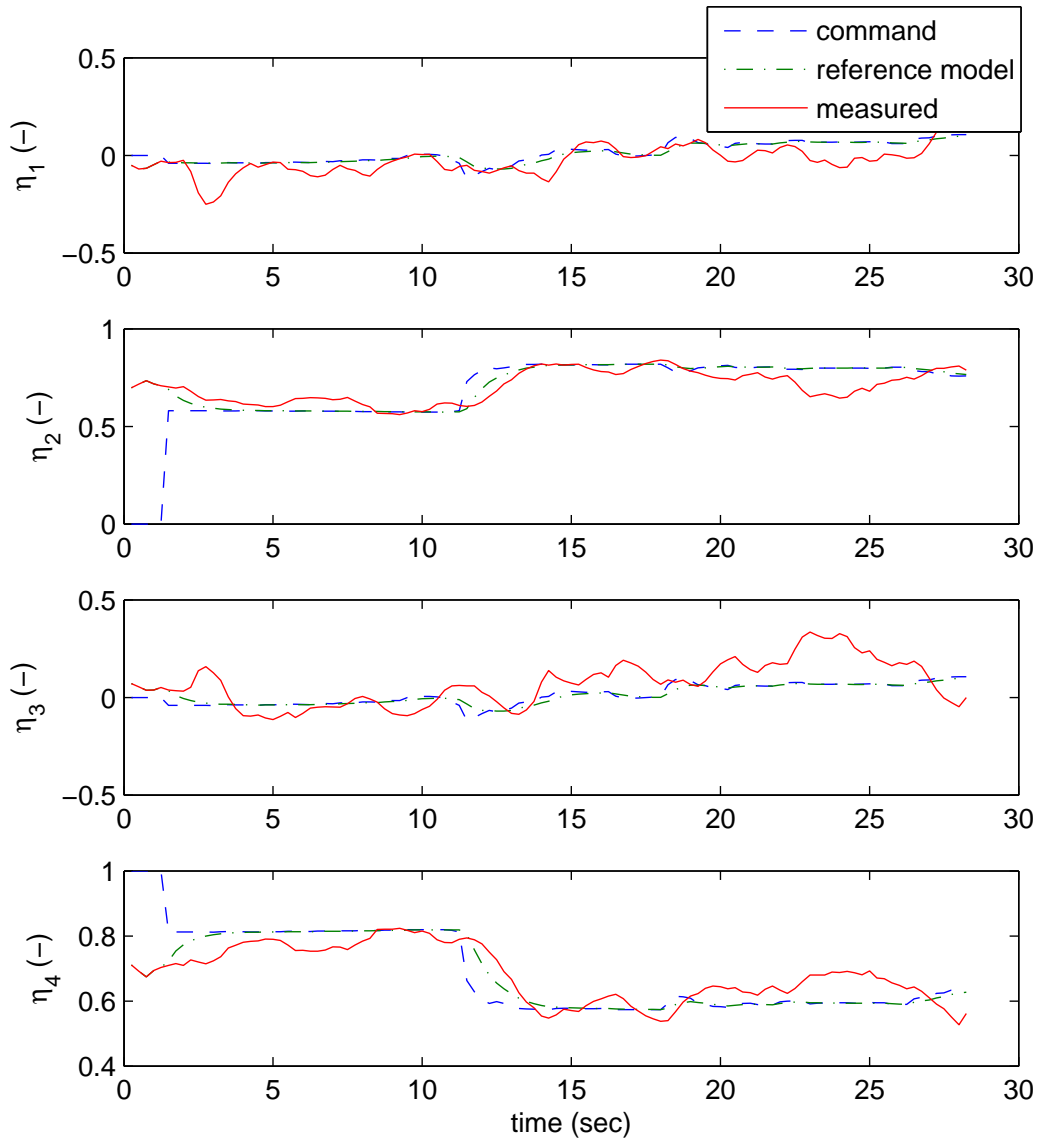


Figure 4.5: Attitude control during the hover north/east position tracking experiment in hardware is shown.

4.2 Hover Altitude Control

Altitude tracking in hover flight, which is essential for the tailsitter autopilot, will be discussed in this section. This problem has proven to be challenging due to the inconsistency of the thrust system used in the hardware experiments conducted. The thrust system is comprised of a brushless electric motor spinning a large propeller and powered by an electronic speed control and lithium polymer batteries (see Chapter 2). It was discovered that thrust produced by the system depends significantly on the condition of the battery being used. Derivation of a method for on-line identification of the thrust system and altitude control to address this problem will be given. Simulation and hardware results from an altitude control experiment will also be presented for technique validation.

4.2.1 Controller Derivation

Several methods of successive PID loop closure with feedforward control were implemented in an attempt to deal with the inconsistency of the thrust system. The intent of these methods was for inner loops to track either propeller wash airspeed or acceleration with integral control to compensate for error in the feedforward model. These attempts all failed.

In this process it was found that the rate at which the system spins the propeller matches the following linear function:

$$\omega_p = C_{\omega_p 0} + C_{\omega_p, E} E + C_{\omega_p, \delta_t} \delta_t, \quad (4.10)$$

where δ_t is the throttle percentage setting and E is the battery voltage. Propeller rate data collected on the thrust system can be seen in Figure 4.6 to support this claim. From further testing it was found that these system parameters were sensitive to the battery being used. Depending on the condition of the battery, significant modeling error was found. Typically new and unused batteries produced higher propeller speeds than older batteries.

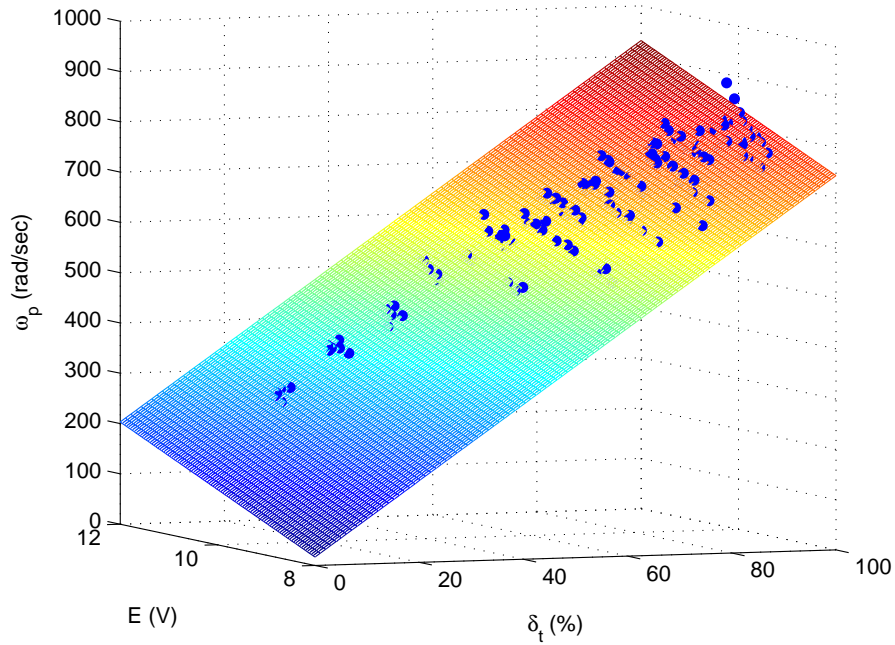


Figure 4.6: This figure shows propeller angular rate data collected in a static throttle test to show the linear relationship between propeller speed and both the battery voltage and throttle setting.

It was also found that the speed of the air exiting the propeller (V_p) can be modeled as a simple linear function of the angular rate of the propeller as follows:

$$V_p = C_{V_p, \omega_p} \omega_p. \quad (4.11)$$

Figure 4.7 verifies this method with propeller angular rate and exiting airspeed data collected in a static throttle test.

As a result of these discoveries, on-line thrust model identification through linear parameter estimation followed as a natural solution. The stabilized recursive least squares with data forgetting algorithm was chosen as the means for parameter estimation.

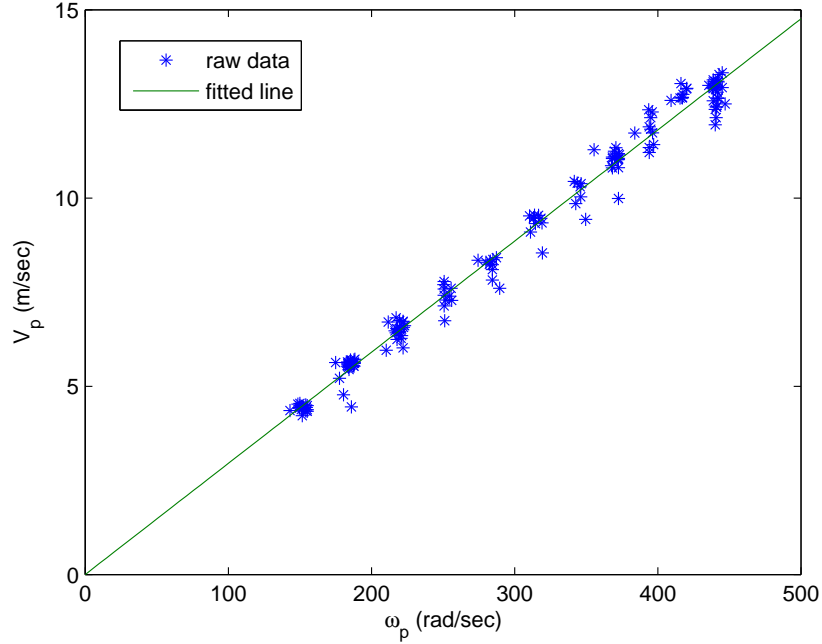


Figure 4.7: Propeller data collected in a static throttle test to show the linear relationship between the speed of air exiting the propeller and the propeller angular rate is shown.

Assuming that in hover flight the aircraft airspeed (V) is zero, the thrust (T) produced from the propeller exiting airspeed is

$$T = \frac{1}{2} \rho A_p V_p^2 \quad (4.12)$$

from equation (2.4), where ρ is the air density and A_p is the area of the disc covered by the spinning propeller. From equations (4.10), (4.11), and (4.12), thrust can be rewritten as

$$T = \left(\sqrt{\frac{1}{2} \rho A_p C_{V_p, \omega_p} (C_{\omega_p 0} + C_{\omega_p, E} E + C_{\omega_p, \delta_t} \delta_t)} \right)^2. \quad (4.13)$$

Noting that onboard accelerometers measure forces divided by mass, and assuming that in hover, the only body reference frame x -axis force is from thrust yields

$$a_x = \frac{T}{\mu}, \quad (4.14)$$

where a_x denotes the measured body reference frame x -axis acceleration from forces and μ is the tailsitter mass. Combining equations (4.13) and (4.14) results in

$$a_x \mu = \left(\sqrt{\frac{1}{2} \rho A_p C_{V_p, \omega_p}} (C_{\omega_p 0} + C_{\omega_p, E} E + C_{\omega_p, \delta_t} \delta_t) \right)^2, \quad (4.15)$$

and taking the square root of both sides to be linear in system parameters gives

$$\sqrt{a_x \mu} = \theta_1 + \theta_2 E + \theta_3 \delta_t. \quad (4.16)$$

Note that θ_1 , θ_2 , and θ_3 are the system parameters

$$\begin{aligned} \theta_1 &= \sqrt{\frac{1}{2} \rho A_p C_{V_p, \omega_p} C_{\omega_p 0}}, \\ \theta_2 &= \sqrt{\frac{1}{2} \rho A_p C_{V_p, \omega_p} C_{\omega_p, E}}, \text{ and} \\ \theta_3 &= \sqrt{\frac{1}{2} \rho A_p C_{V_p, \omega_p} C_{\omega_p, \delta_t}}. \end{aligned} \quad (4.17)$$

In vector form, equation (4.16) can be written as

$$\sqrt{a_x \mu} = \Phi^T \Theta, \quad (4.18)$$

where

$$\Phi = \begin{pmatrix} 1 \\ E \\ \delta_t \end{pmatrix}$$

and

$$\Theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix}.$$

As a result, the difference between the estimated and measured square root of thrust is minimized with the following objective function:

$$J = \sum_{k=1}^N \frac{1}{2} \left(\Phi[k]^T \hat{\Theta}[N] - \sqrt{a_x[k]\mu} \right)^2 \lambda^{N-k} + \sum_{k=1}^3 \frac{1}{2} \alpha_k \left(\hat{\theta}_k[N] - \hat{\theta}_k[N-1] \right)^2. \quad (4.19)$$

The same stabilized recursive least squares with data forgetting algorithm that was derived in Chapter 3 is then used to estimate thrust parameters that minimize equation (4.19), namely

$$\hat{\Theta}[N] = \hat{\Theta}[N-1] + P[N]\Phi[N] \left(\sqrt{a_x[k]\mu} - \Phi[N]^T \hat{\Theta}[N-1] \right) \quad (4.20)$$

and

$$P^{-1}[N] = \lambda P^{-1}[N-1] + \Phi_1[N]\Phi[N]^T + \alpha(1-\lambda), \quad (4.21)$$

where α denotes the diagonal matrix

$$\alpha = \begin{pmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{pmatrix}.$$

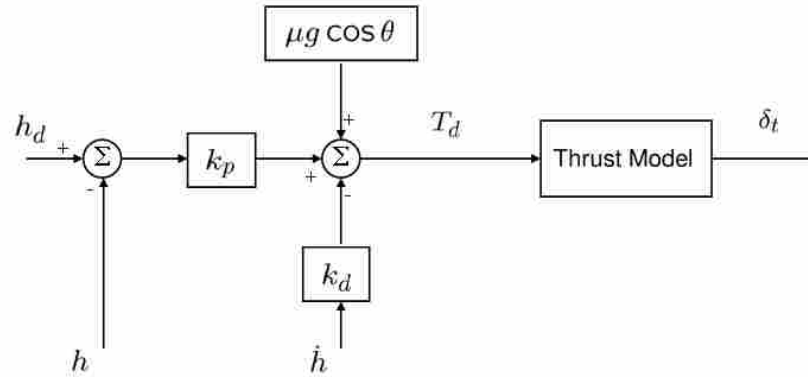


Figure 4.8: Above is a block diagram of the hover altitude controller.

Since the thrust system is identified on-line, a simple feedforward controller to counter gravity with a proportional and derivative (PD) feedback loop around altitude can be implemented to give desired thrust. Throttle for matching desired thrust (T_d) can then be obtained from the identified thrust model as

$$\delta_t = \frac{\sqrt{T_d} - \hat{\theta}_1 - \hat{\theta}_2 E}{\hat{\theta}_3}. \quad (4.22)$$

A block diagram of this simple controller is found in Figure 4.8. Altitude from barometric pressure was used in the experiments as an altitude and altitude rate measurement. Because the pressure sensors used on the autopilot have significant noise, the differentiated altitude signal was filtered with a first-order low-pass filter to reject noise amplified in the process.

4.2.2 Simulation Results

In the altitude hover experiment, the system was given three step inputs. The aircraft started from a vertical takeoff location and then was commanded to 15, 10, and finally back to 0 meters. Simulation results of the altitude tracking controller are seen in Figure 4.9. These results show the altitude controller tracking the changes in desired altitude with acceptable error. In the same figure, altitude rate is shown to remain small with derivative control, which is significant because large descent rates reduce attitude actuator effectiveness.

Estimated thrust from the model is shown to track the measured thrust during the same experiment in Figure 4.10. This plot can be used as a metric of how well the thrust model matches the true system. Desired thrust is also shown on the same figure to verify the system's ability to produce the intended thrust.

System parameters throughout the test are shown in Figure 4.11. In simulation, the actual parameters are known. It can be seen that the true parameters are learned reasonably well within a few seconds of takeoff, despite initial error.

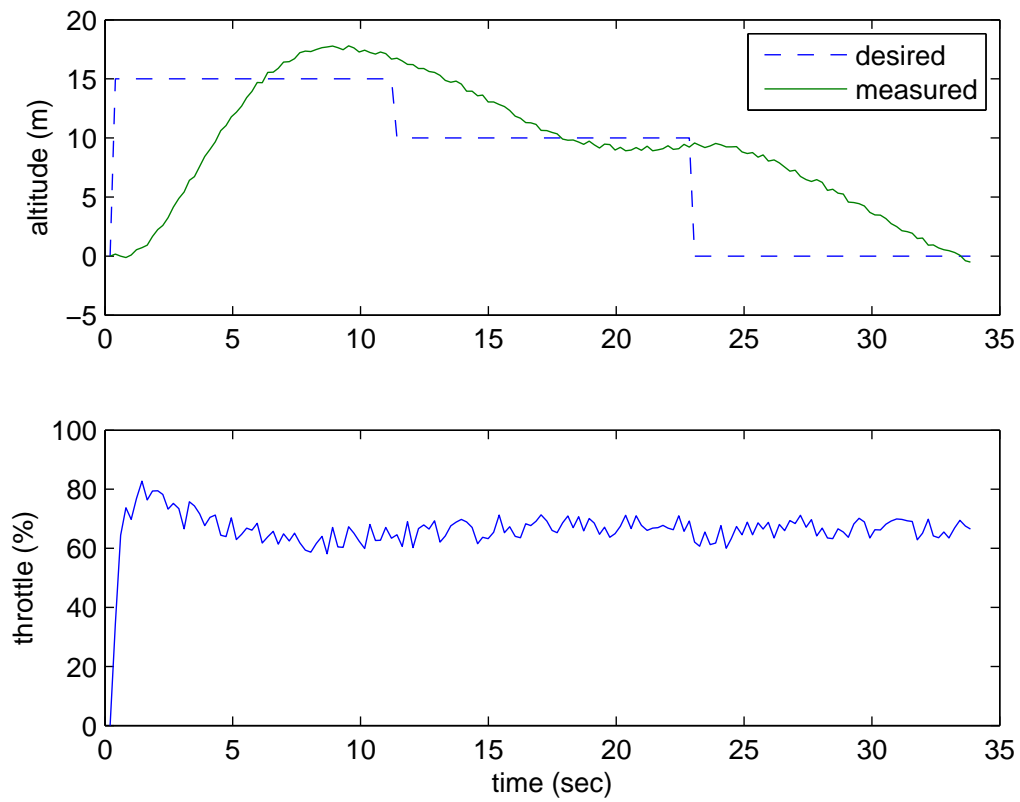


Figure 4.9: In this figure, altitude, altitude rate, and throttle setting during the simulation test of the hover altitude controller are shown.

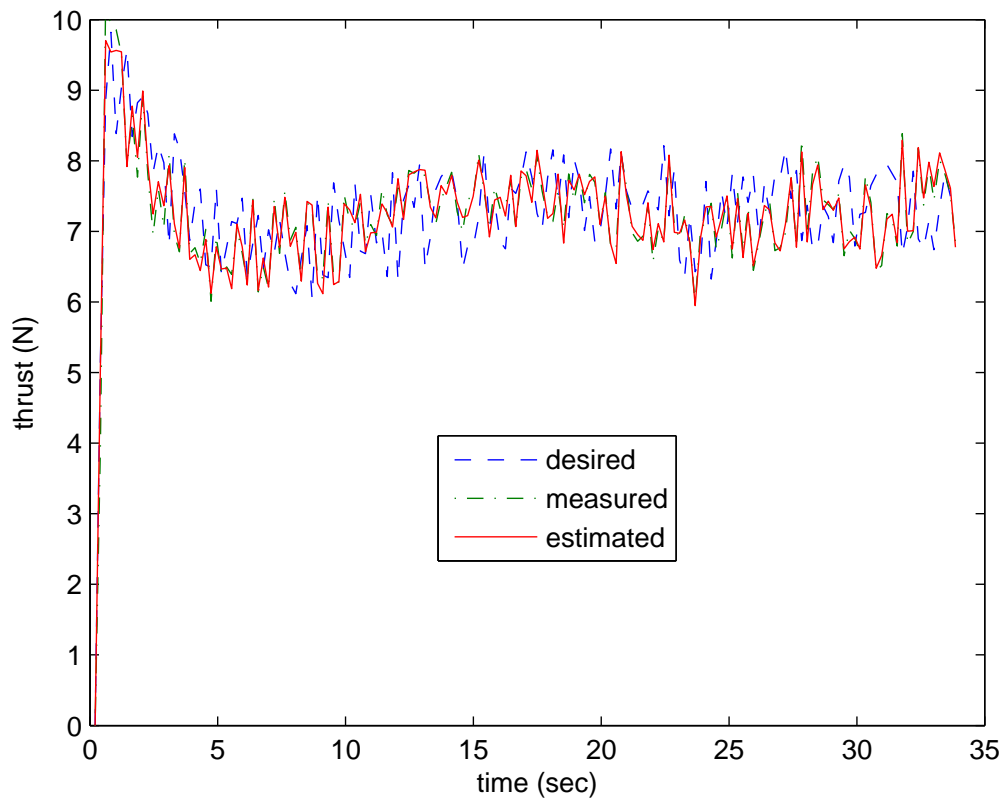


Figure 4.10: This figure presents the thrust during the simulation test of the hover altitude controller.

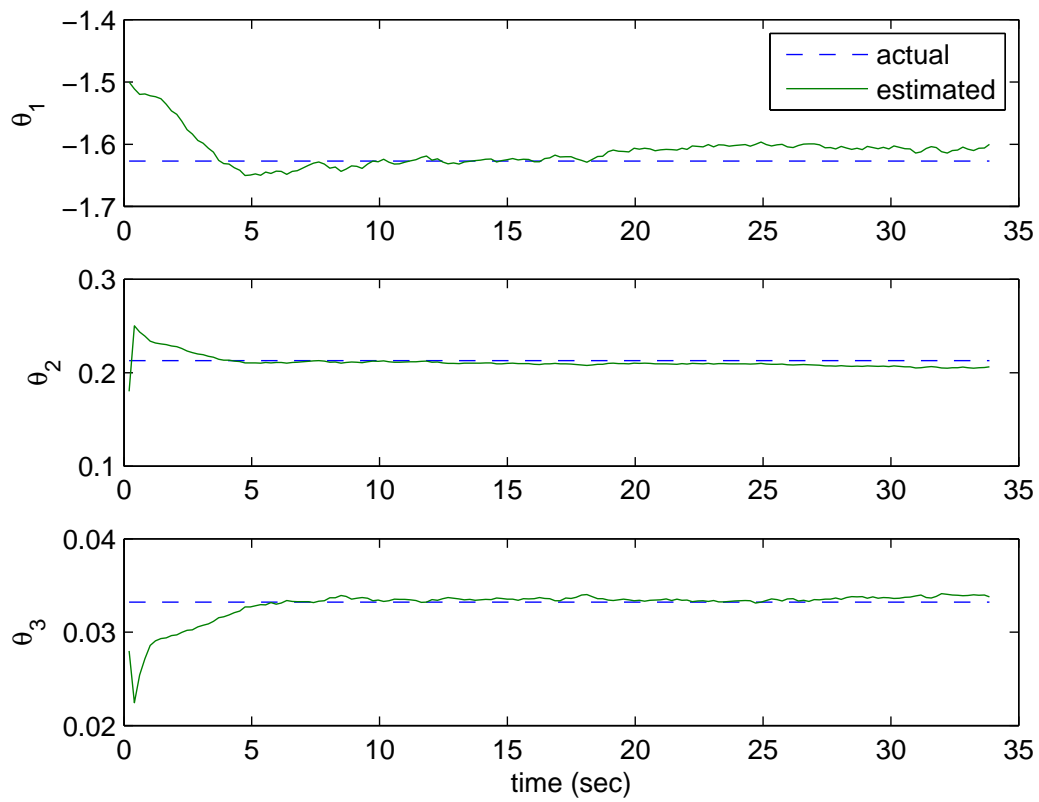


Figure 4.11: Thrust system parameters during the simulation test of the hover altitude controller are shown.

4.2.3 Hardware Results

The same test was performed for hardware verification. The tailsitter starting from takeoff was given inputs to 15, 10, and 0 meters. Acceptable altitude tracking results similar to what was observed in simulation are given in Figure 4.12. Altitude rate is also shown to be minimized with derivative control in the same figure.

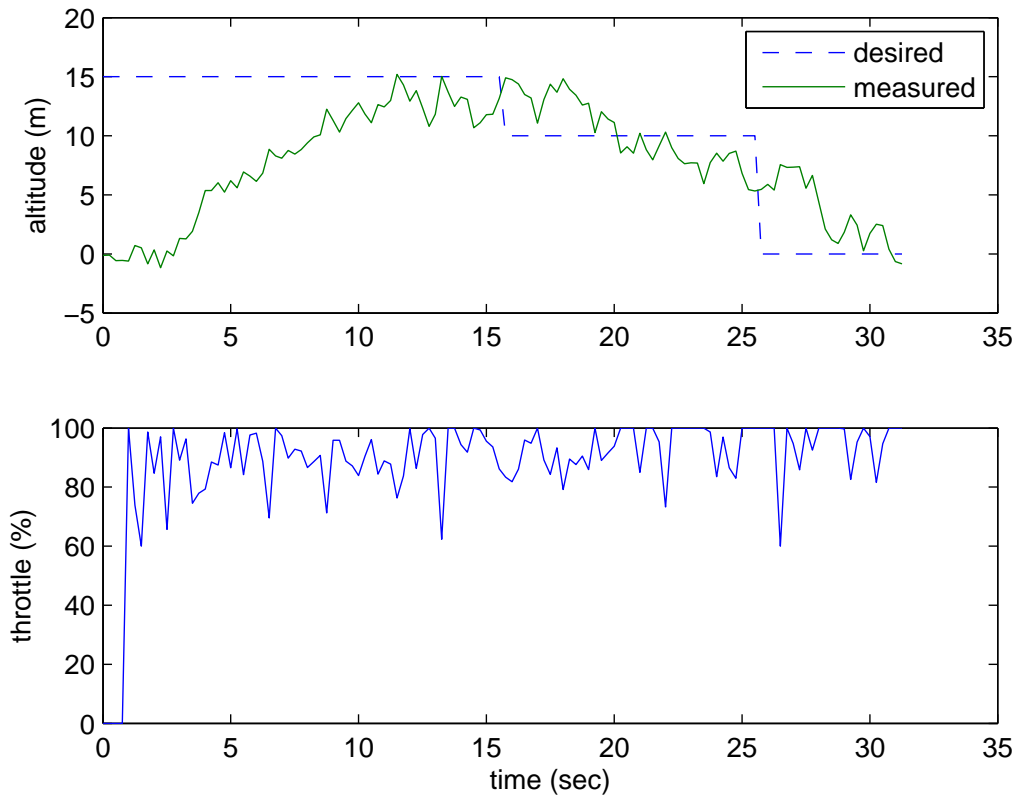


Figure 4.12: In this figure, altitude, altitude rate, and throttle setting during the hardware test of the hover altitude controller are given.

The system thrust model is verified in Figure 4.13. Estimated thrust matches the measured signal reasonably well, despite noise in the x -axis accelerometer. The desired thrust is also tracked by the measurement signal, except in throttle saturation, which occurs mainly in three locations toward the end of the experiment (about 23, 25, and 28 seconds).

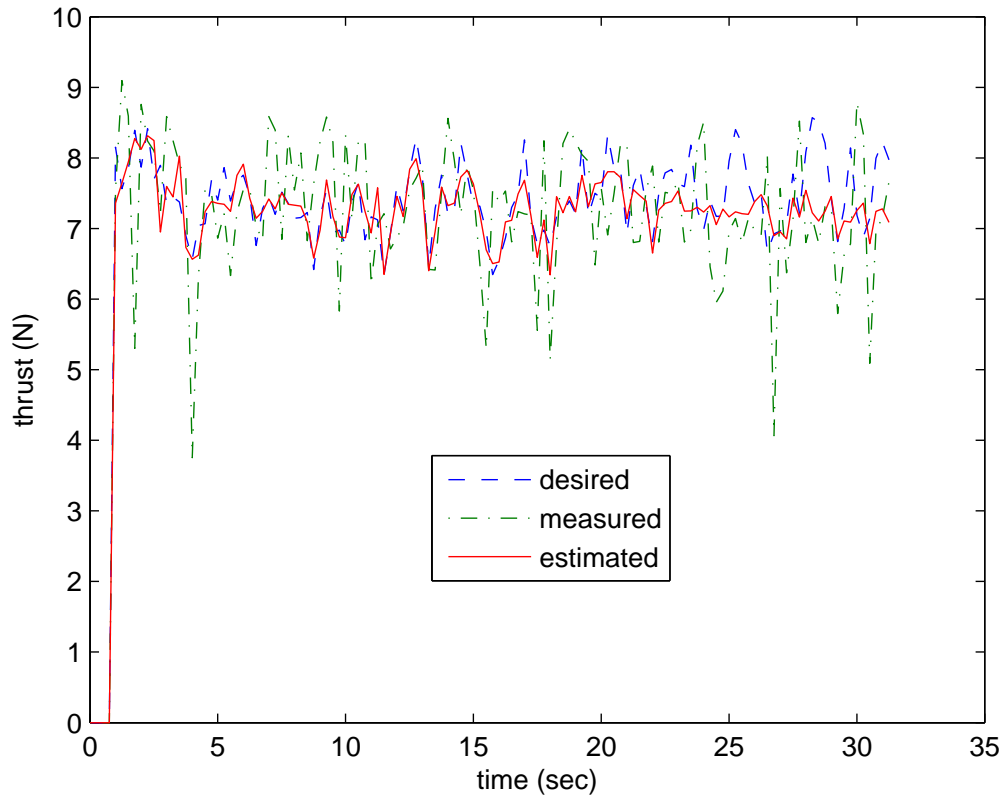


Figure 4.13: Thrust during the hardware test of the hover altitude controller is given.

The system parameter adaptation (see Figure 4.14) also resembles results found in simulation. It can be seen that the estimated parameters settle at a particular value within a few seconds during the initial climb. A slow change in the terms occurs as the system descends. Because the adjustments are small and gradual, this behavior is most likely due to actual changes in the thrust system. Both the matching of the estimated and measured thrust signals and the settling of the system parameters, together with considerable changes in throttle commands, indicate that the true system parameters have been accurately identified.

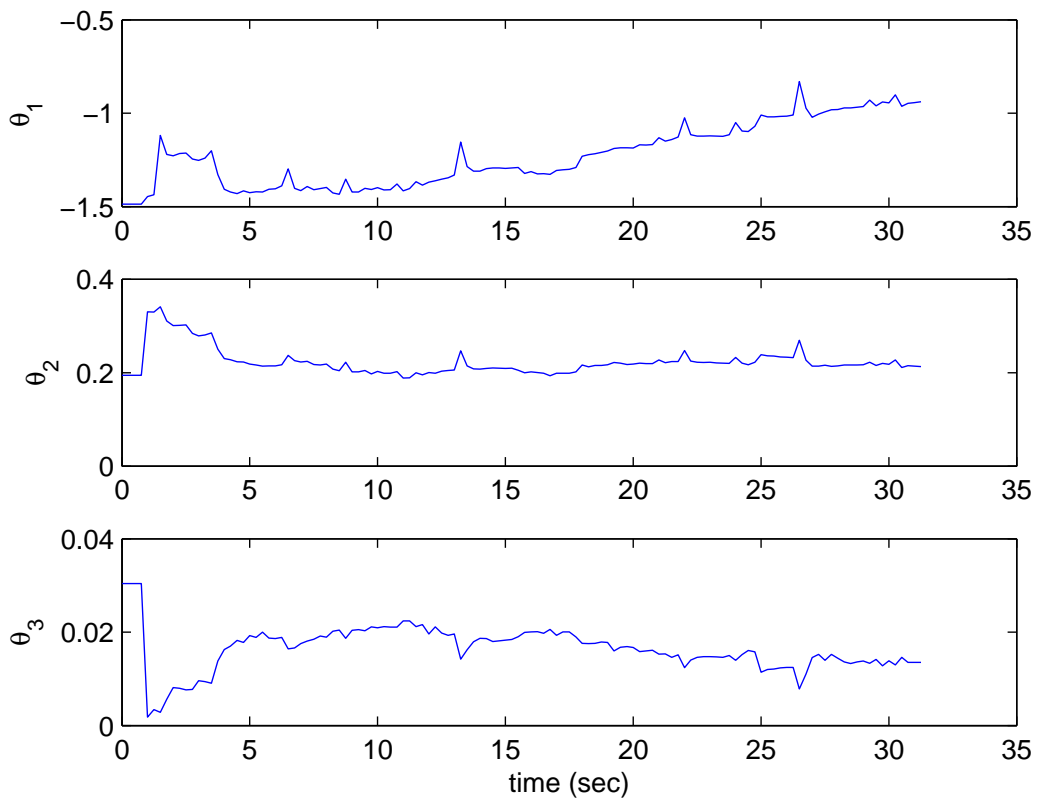


Figure 4.14: Above are the thrust system parameters during the hardware test of the hover altitude controller.

4.3 Level Flight Navigational Control

Level flight path navigation, as well as altitude and airspeed control, are discussed in this section. The techniques presented are based on proven control methodology previously developed at Brigham Young University. Adaptations are made to conform the methods to the quaternion-based autopilot. For concept verification, hardware and simulation results are presented at the end of the section.

4.3.1 Controller Derivation

Tailsitter level flight waypoint tracking is based on a nonlinear vector field sliding mode controller [40]. The technique generates a desired course ensuring that the aircraft aligns itself with the vector field sliding surface. The inverse tangent function is used to define the vector field in the following manner:

$$\chi_d = \chi_{path} - \chi_\infty \frac{2}{\pi} \tan^{-1} ky, \quad (4.23)$$

where χ_{path} is the path course, χ_∞ is the approach course when the vehicle is a considerable distance from the path, k is a gain that determines the transition of the desired course from $\chi_{path} - \chi_\infty$ to χ_{path} , and y is the perpendicular distance of the vehicle to the path. An example vector field is shown in Figure 4.15.

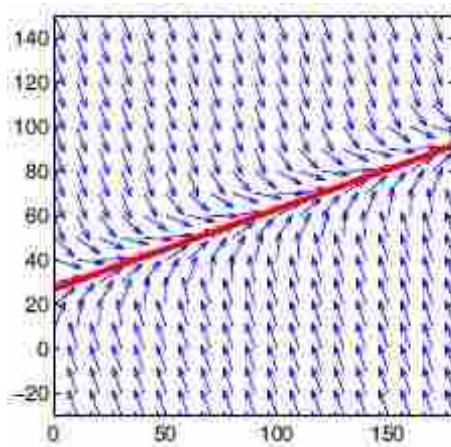


Figure 4.15: This shows a vector field sliding surface example. The red line depicts the desired path and the blue arrows represent the vector field.

For a quaternion-based autopilot, the natural method of tracking a course is through yaw rotations. Thus, instead of banking the aircraft to turn, the desired heading is simply set as the course error added to the current heading angle given by

$$\psi_d = \psi + (\chi_d - \chi), \quad (4.24)$$

which results in yaw or skid-to-turn maneuvers.

Airspeed and altitude are tracked with switching logic between PID controllers. This method is based on the understanding that for airspeed to be maintained during large changes in altitude, the rate at which energy enters the system has to increase or decrease from the nominal value. At the same time, small changes in altitude can occur faster with pitch angle commands due to the dynamics of aircraft. Consequently, above and below the desired altitude, threshold limits are set defining three regions or modes of control. Below the lower limit, throttle is fixed high for climbing and airspeed is controlled with pitch angle generated from PID control. Similarly, above the upper threshold, throttle is set low and airspeed is maintained with the same pitch angle controller. Inside of the altitude threshold, however, altitude is tracked with pitch angle and airspeed with throttle, which is also accomplished with PID loops. A diagram visually portraying this switching logic is shown in Figure 4.16.

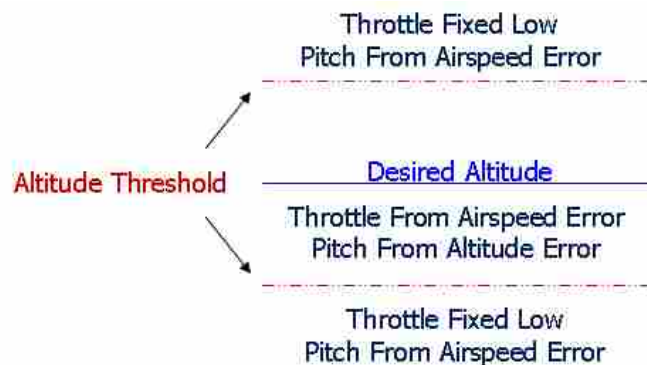


Figure 4.16: In this diagram the logic for the level flight altitude and airspeed controller is shown.

Given a desired heading (ψ_d) and pitch angle (θ_d), the desired quaternion is found from

$$\bar{\eta}_d = \begin{pmatrix} -\sin(\psi_d/2) \sin(\theta_d/2) \\ \cos(\psi_d/2) \sin(\theta_d/2) \\ \sin(\psi_d/2) \cos(\theta_d/2) \\ \cos(\psi_d/2) \cos(\theta_d/2) \end{pmatrix}. \quad (4.25)$$

This is simply two quaternion rotations composed together:

$$\bar{\eta}_d = \bar{\eta}_{\theta_d} \otimes \bar{\eta}_{\psi_d}, \quad (4.26)$$

where

$$\bar{\eta}_{\psi_d} = \begin{pmatrix} 0 \\ 0 \\ \sin \frac{\psi_d}{2} \\ \cos \frac{\psi_d}{2} \end{pmatrix}$$

and

$$\bar{\eta}_{\theta_d} = \begin{pmatrix} 0 \\ \sin \frac{\theta_d}{2} \\ 0 \\ \cos \frac{\theta_d}{2} \end{pmatrix}.$$

4.3.2 Simulation Results

Simulation results of the level flight navigational controllers are presented in this section. The simulation test involved four level flight waypoints in an hourglass pattern spaced a hundred meters apart. Altitudes were varied so that on each leg of the path the aircraft was either climbing or descending. The resulting path flown can be seen in Figure 4.17, with course and heading tracking shown in Figure 4.18. From these figures, it can be seen that the path following navigation accurately guides the aircraft, such that the vehicle converges to and follows the desired path.

Results of the level flight altitude and airspeed control method are shown in Figure 4.19. Threshold values can be seen in the altitude plot to determine what

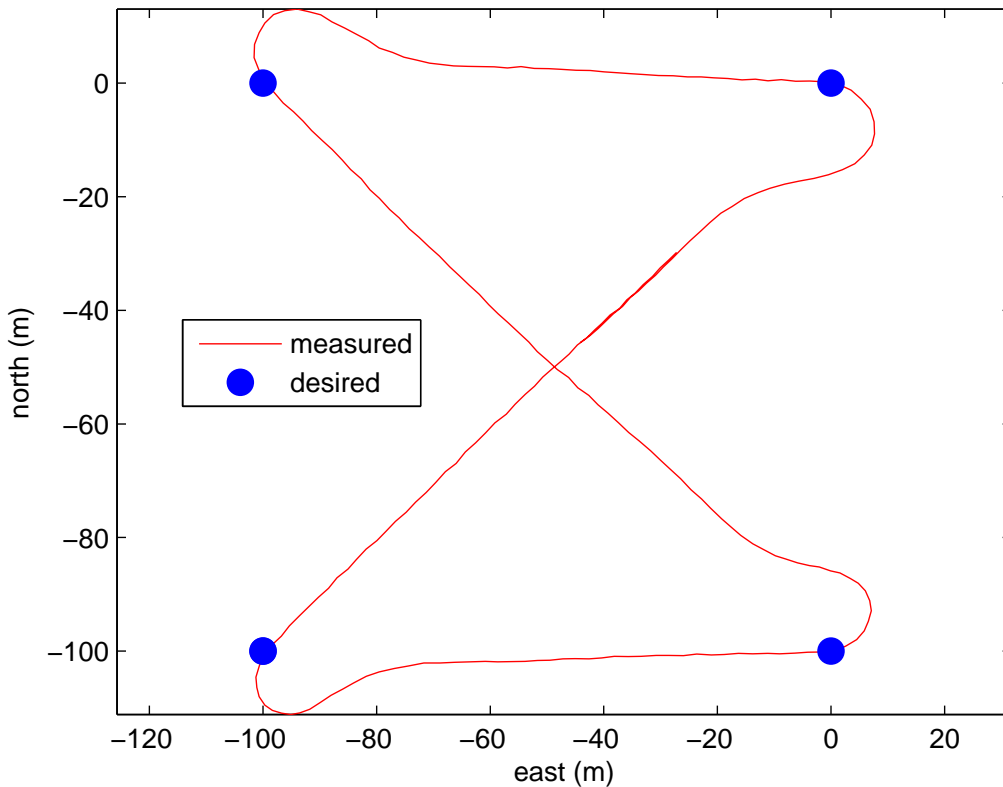


Figure 4.17: This shows the path flown in the simulated test of level flight navigational control.

control mode the system is in. Altitude is shown to be tracked reasonably well with pitch angle. Also, the climb and descent outside of the thresholds were both executed with fixed throttle. Considerable error in airspeed occurs at 10 seconds due to the pitch angle reaching saturation; otherwise, airspeed is maintained well with pitch. For airspeed controlled from throttle, airspeed is tracked when the pitch angle is close to the nominal setting. As expected, with large changes in pitch indicating a reasonable climb or descent, the airspeed is not tracked as well. Including an improved feedforward model, where the nominal throttle setting is adjusted based upon pitch angle, is one possible solution for improvement. Also, the thresholds of the controller can be tightened to eliminate large changes in pitch while attempting to track airspeed with throttle.

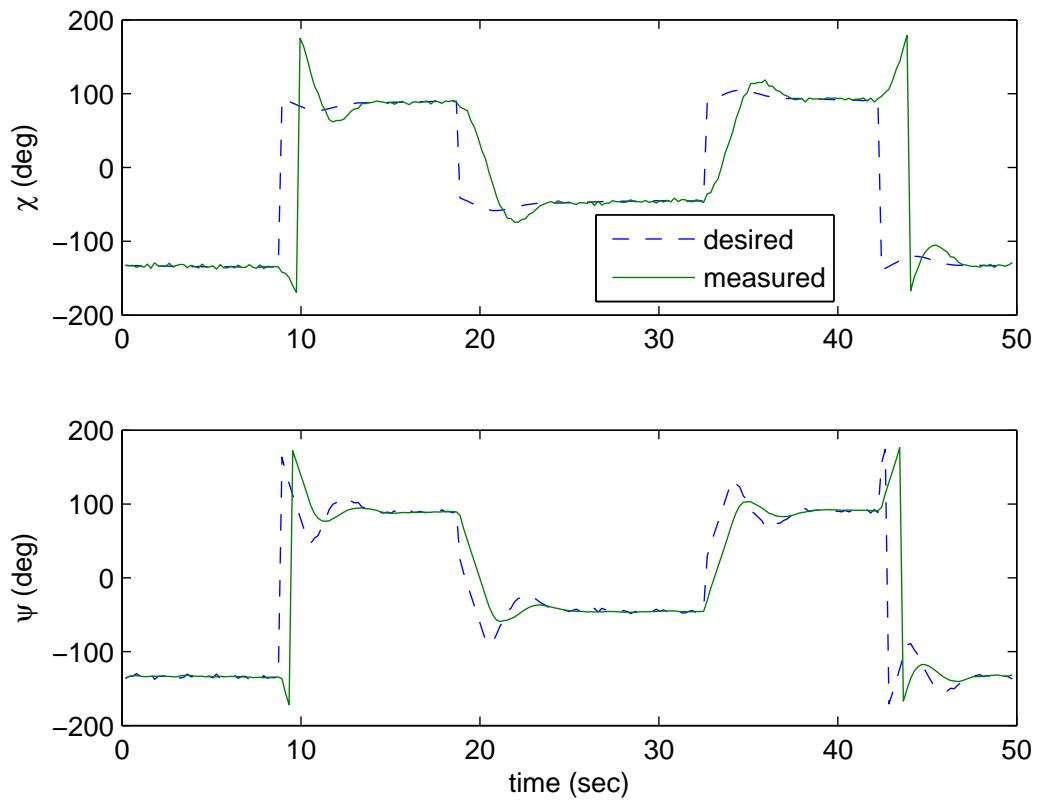


Figure 4.18: Simulation results of course and heading during the level flight control test are given.

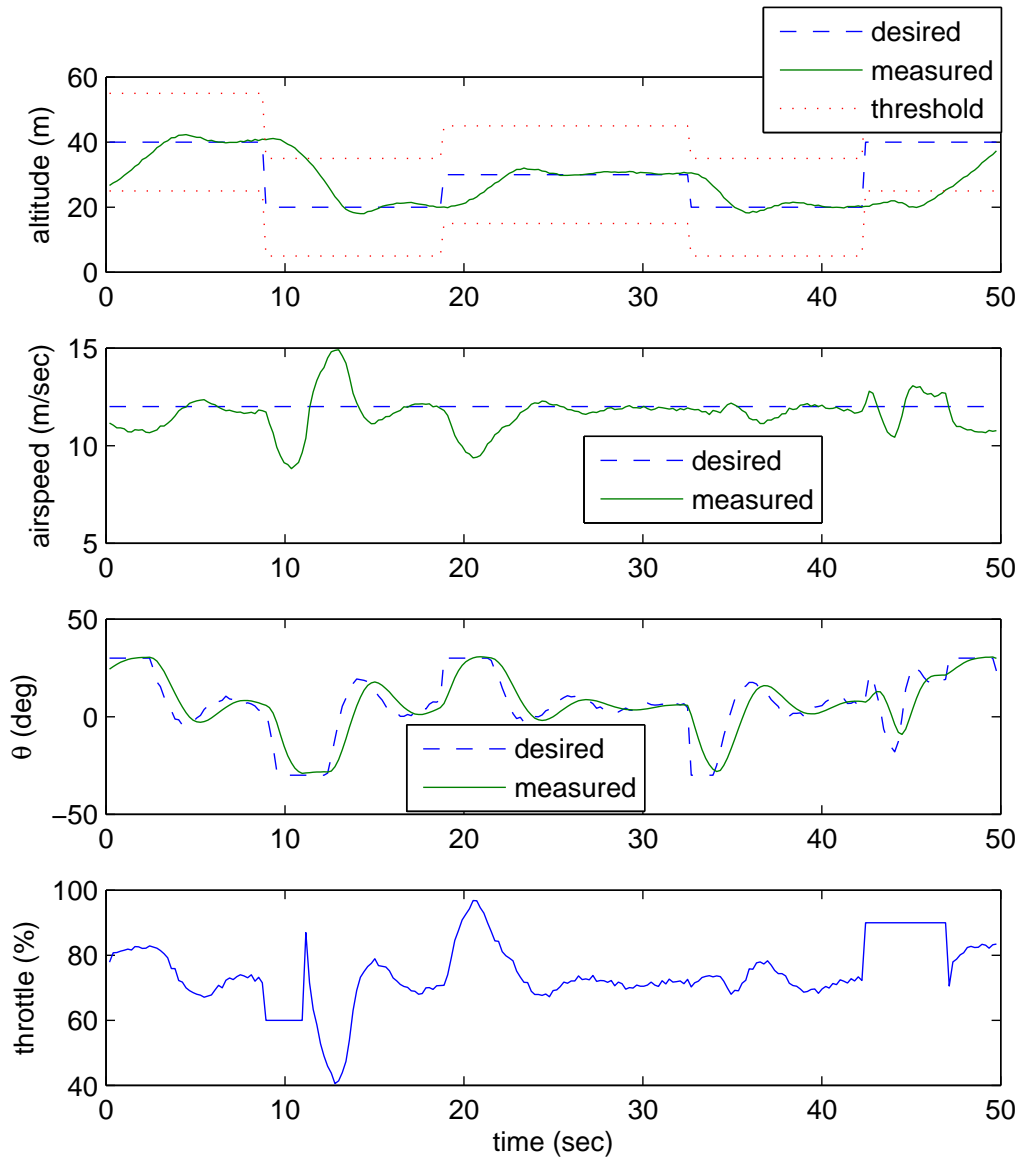


Figure 4.19: Above, the simulation results of airspeed and altitude during the level flight control test are presented.

4.3.3 Hardware Results

Hardware results from the same level flight test are given in this section. The path flown can be seen in Figure 4.20. The executed path is noticeably worse than that of the simulated controller. Course and heading during the flight are shown in Figure 4.20. From this figure it can be seen that heading error is reasonably small and even comparable to the simulation results shown. Course, however, in the hardware test exhibits significant overshoot and delay. This might be attributed to the delay in course measurement given from GPS as a result of filtering. During sharp turns, typical of the skid-to-turn maneuver, course changes rapidly. A course measurement delay in such circumstances would result in significant course error degrading the overall performance of the controller.

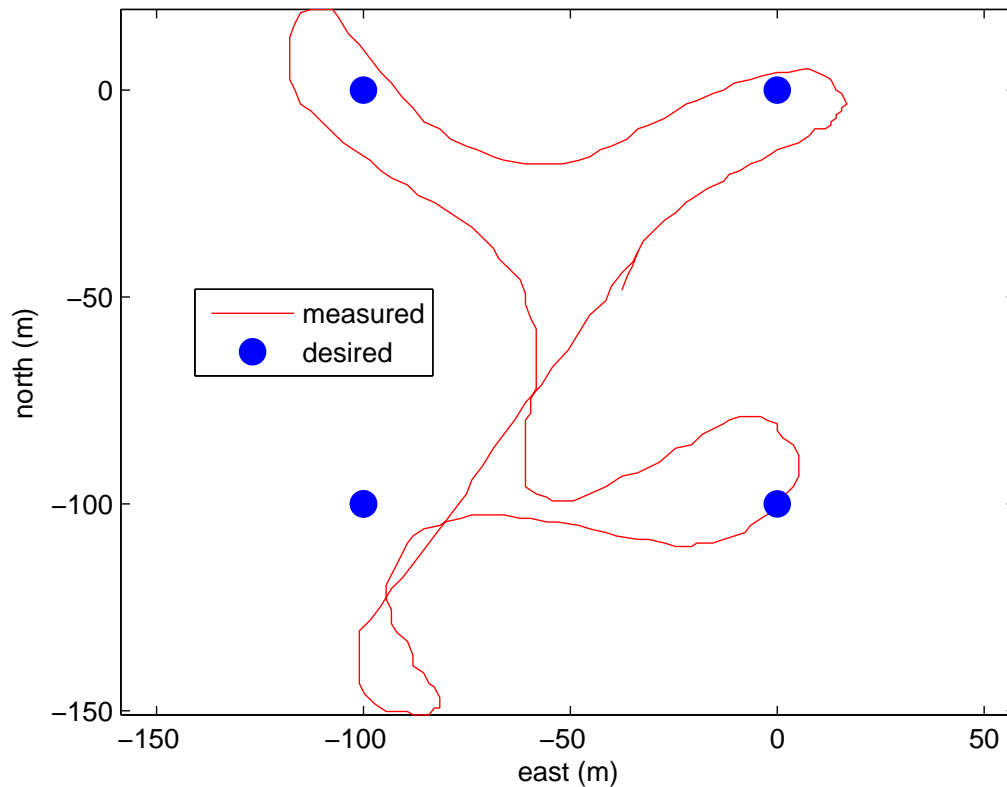


Figure 4.20: This figure exhibits the path flown in the hardware test of level flight navigational controllers.

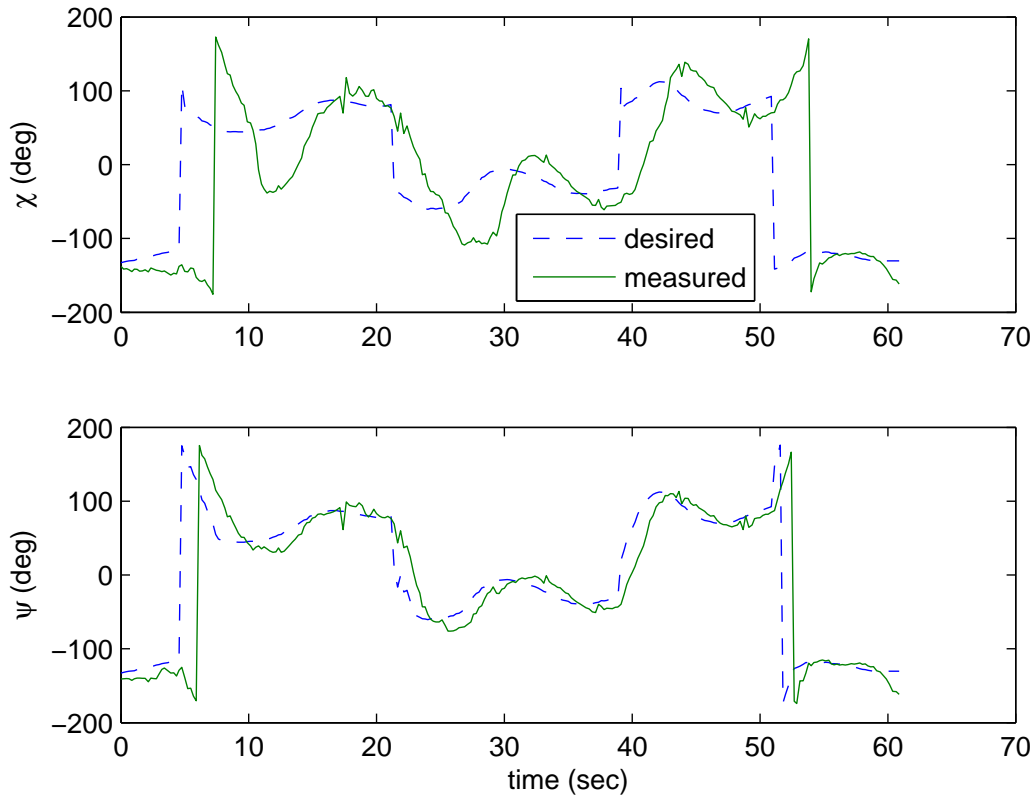


Figure 4.21: Above, the hardware results of course and heading during the level flight control test are shown.

Level flight altitude and airspeed controller results are given in Figure 4.22. Unfortunately, the tailsitter spends little time outside of the threshold limits. Be that as it may, altitude performance is acceptable and comparable to results shown in simulation. Airspeed tracking with throttle in the hardware experiment exhibits considerable error, which can be attributed to both throttle saturation and large changes in pitch as discussed in the simulation results section.

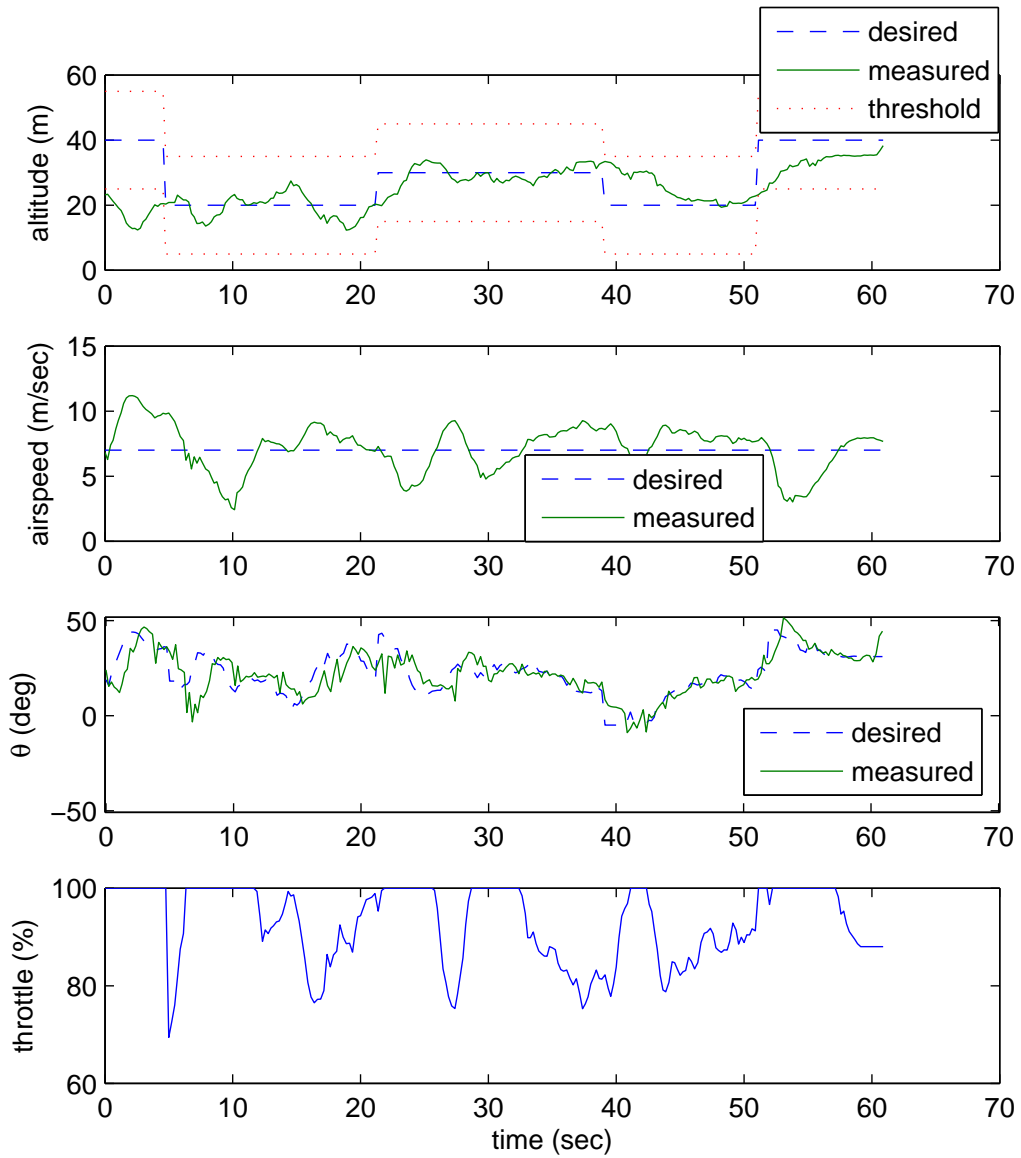


Figure 4.22: Hardware results of airspeed and altitude during the level flight control test are given.

4.4 Transitions

Transitional maneuvers are necessary to change between hover and level flight modes. Simple methods developed for the tailsitter autopilot will be presented in this section. For simplicity, the transitions are broken down into stages, such that one control mode is executed at a time. Throughout the maneuvers, stall is avoided. Only pitch rotations are performed during the actual transition from hover to level flight and vice versa. Note that the methodology was not designed to optimize altitude deviation. Results from actual transitions performed in simulation and hardware are given to validate the methods presented.

4.4.1 Controller Derivation

Hover-to-level flight transitions are executed in two stages. In the first stage, the aircraft rolls to align its underside with the direction of the desired waypoint path. The desired quaternion for such a rotation is given by

$$\bar{\eta}_d = \bar{\eta}_v \otimes \bar{\eta}_{\psi_{path}}, \quad (4.27)$$

where

$$\bar{\eta}_{\psi_{path}} = \begin{pmatrix} 0 \\ 0 \\ \sin \frac{\psi_{path}}{2} \\ \cos \frac{\psi_{path}}{2} \end{pmatrix}.$$

Once the error between the estimated and desired quaternion is reduced to an acceptable point, the second stage is executed. At this point, the actual transition occurs, where the desired quaternion is set equal to $\bar{\eta}_{\psi_{path}}$, resulting in a pitch forward to level flight rotation. Finally, when error is reduced, the normal level flight commands are implemented. Throughout the entire transition maneuver, throttle is set full to increase airspeed and avoid stall conditions.

Level-to-hover flight transitions are performed in a similar manner. As the aircraft approaches a hover waypoint, the normal level flight controller is employed.

When the distance to the desired point is reduced below a predetermined value, the transition is initiated. In the first stage, the aircraft is pitched up to a vertical orientation, such that the underside points in the direction of the heading that was measured before the rotation was executed. The desired quaternion for this orientation is found by

$$\bar{\eta}_d = \bar{\eta}_v \otimes \bar{\eta}_\psi, \quad (4.28)$$

where

$$\bar{\eta}_\psi = \begin{pmatrix} 0 \\ 0 \\ \sin \frac{\psi}{2} \\ \cos \frac{\psi}{2} \end{pmatrix}.$$

Once this is accomplished, the second stage begins wherein the tailsitter rotates to the vertical quaternion and finally, the normal hover position controller is implemented. Throughout this maneuver, throttle is set from the altitude hover control method. Because the approach speed of the aircraft is relatively large and the pull-up maneuver is executed quickly, aircraft stall is not of significant concern.

Note that the reference model (see Chapter 3) is applied to the desired attitude signal. This acts as a pre-filter and represents the achievable dynamics of the aircraft. As a result, large changes in desired attitude, such as the roll-to-heading, pitch-down, and pitch-up commands are smoothed into obtainable trajectories by the reference model.

4.4.2 Simulation Results

Attitude simulation results of the transition methods described are exhibited in Figure 4.23. A hover-to-level transition is shown in the left column; level-to-hover is shown in the right. In the hover-to-level transition, a heading rotation of 45 degrees occurs from 0.5 to 1.5 seconds. The pitch forward to level flight then can be seen up until about 3.0 seconds. The level-to-hover flight transition approaches the desired hover point from a south direction, so no heading rotation is required, only a pitch-up maneuver. A three-dimensional view of these transitions can be seen in Figure 3.4.

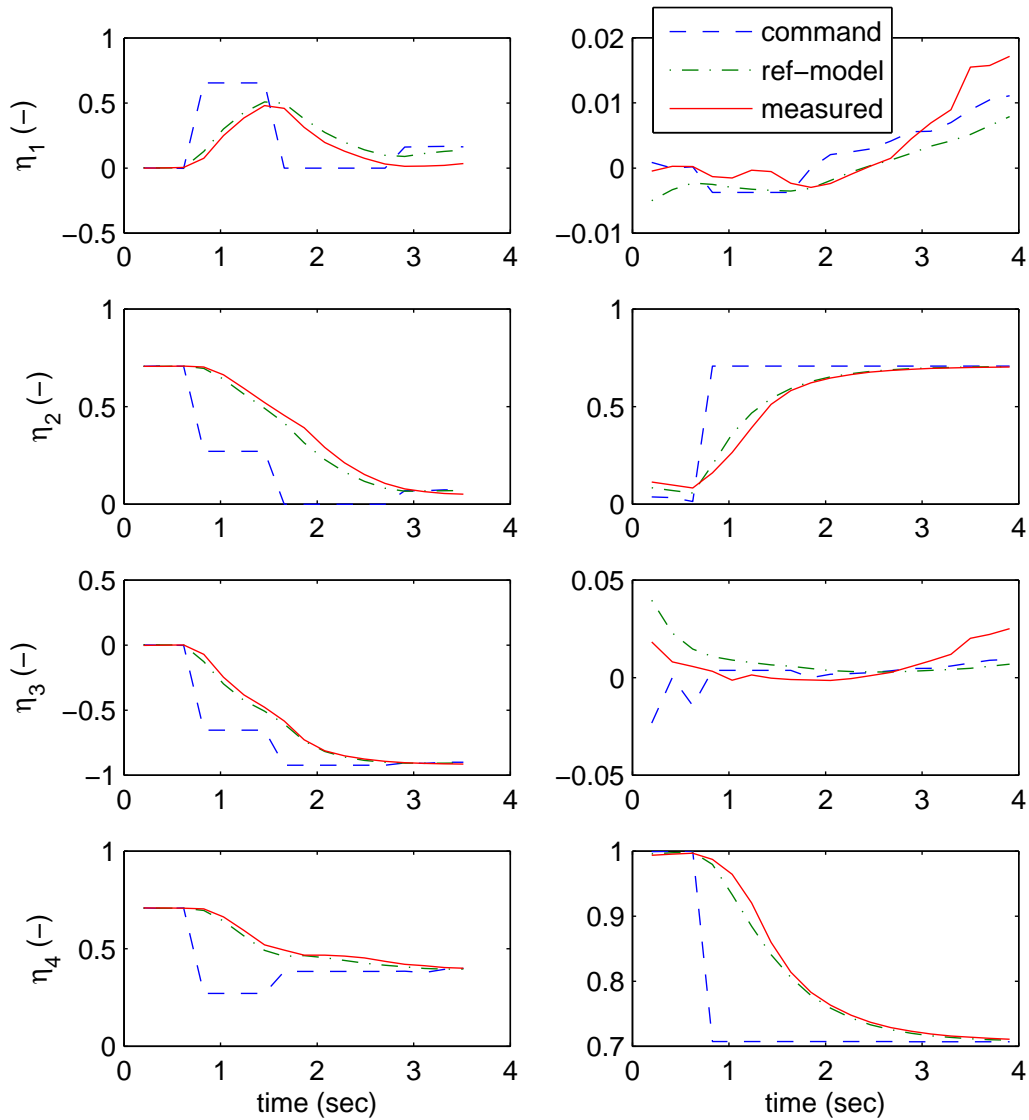


Figure 4.23: This figure shows the attitude simulation results of the tailsitter in transitions. The left column is a hover-to-level flight transition and the right column is a level-to-hover flight transition.

4.4.3 Hardware Results

Transition hardware results are shown in Figure 4.24. On the left column can be seen the attitude during a hover-to-level transition. A heading rotation of 180 degrees is executed from roughly 1 to 2 seconds, followed by a pitch-over maneuver. Note that as discussed in the results section of Chapter 3, a value of zero for the fourth element of a quaternion indicates a rotation of 180 degrees, which explains the symmetric behavior of the second and third elements in the figure. On the right column is shown the succeeding level-to-hover transition. The pitch-up rotation occurs at about 0.5 to 1 seconds. A small heading rotation of 45 degrees is then performed. A three dimensional plot of this path flown can also be seen in Figure 3.18.

4.5 Summary

In this chapter, methods for generating desired quaternion attitude trajectories and throttle commands to produce tailsitter flight maneuvers were developed. Novel quaternion-based north/east position control was created for hover flight. An original method for identifying an inconsistent thrust system to facilitate altitude tracking in hover was also derived. Other level flight techniques based upon previous work were produced for the quaternion tailsitter autopilot. This included path following and altitude/airspeed control. Simple means of transitioning between the two modes of flight were also discussed. All of these navigational controllers were tested in both simulation and hardware for conceptual verification.

In the subsequent chapter (Chapter 5), a scheme similar to the fixed-gain Kalman filter for quaternion attitude estimation will be presented. Note that an estimate of attitude was required for the testing of both attitude and navigational controllers.

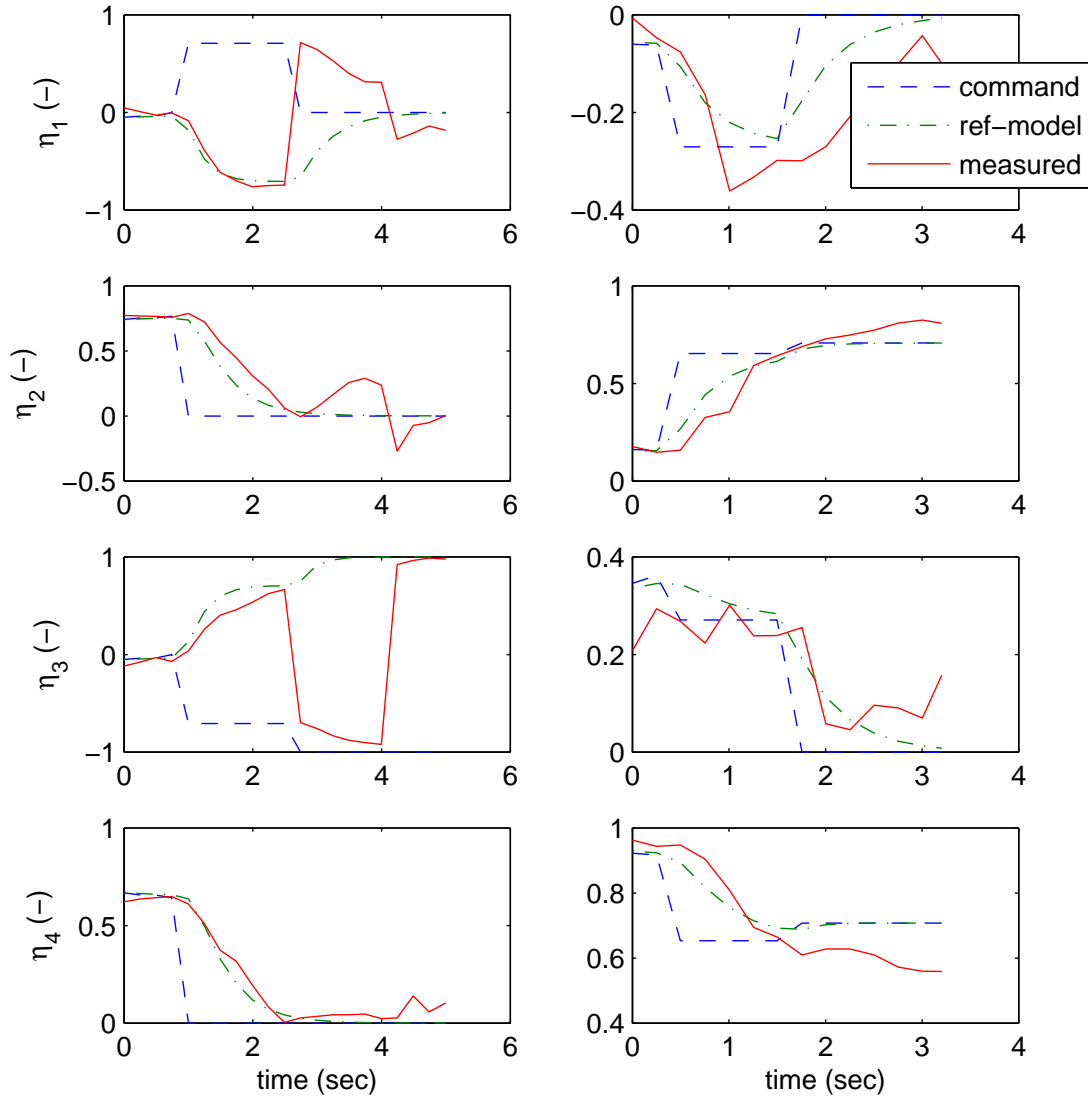


Figure 4.24: This figure gives the attitude hardware results of the tailsitter in transitions. The left column is a hover to level flight transition and the right column is a level to hover flight transition. Note that because the hover-to-level transition occurs in a south direction, the fourth element of the quaternions approach zero. This explains the symmetric behavior of the second and third elements.

Chapter 5

Attitude Estimation

For quaternion estimation, a method employing a GPS, angular rate gyros, accelerometers, and magnetometers was developed. Like the fixed-gain Kalman filter, two stages or updates, namely a model (or time) and measurement update, are employed with an associated gain. Note that to define a basis in three-dimensional space, two vectors are required. Accordingly, for the measurement update of orientation, bearing and gravity vectors are utilized. Typically in miniature autopilot design, bearing is given from GPS course, under the assumption that course is largely correlated with heading. However, for the tailsitter in hover the direction of travel is not necessarily associated with the aircraft bearing at all. As a result, three-axis magnetometers are used in hover flight. In level flight, because the magnetic field and gravity vectors are similar in direction, gravity based error induces significant error in bearing with magnetometers. Consequently, in level flight bearing is resolved from GPS course. An accelerometer penalty is introduced based on the understanding that accelerations from all resulting forces are measured, not just the acceleration opposing gravity. Note that as a secondary goal in the research performed, this method was not developed for optimality, but to adequately estimate attitude in tailsitter flight. For this same reason, because hardware results would require significant effort, only simulation was used for concept validation. These results will be given at the end of the chapter.

5.1 Model Update

The model update used in the tailsitter autopilot executes simple Euler integration of angular rate gyros through quaternion kinematic equations that are known

exactly (equations (A.2) and (A.3)). The update is therefore given by

$$\eta_k = \eta_p + \delta t \frac{1}{2} (\eta_p^\times + \eta_{p4} I_3) \omega_m \quad (5.1)$$

and

$$\eta_{k4} = \eta_{p4} - \delta t \frac{1}{2} \eta_p^T \omega_m, \quad (5.2)$$

where $\bar{\eta}_k$ is the new estimate of attitude based upon aircraft kinematics and ω_m is the measurement of angular rates about the aircraft body reference frame axes acquired from rate gyros. The difference in sample time is referred to as δt , and $\bar{\eta}_p$ is the estimate of attitude from the previous autopilot loop cycle. Because ω_m is only an estimate of angular rates and Euler integration is not exact, it can be seen that to avoid drift other measurements are required.

5.2 Measurement Update

The measurement update of the estimation scheme enlists three sensors, namely accelerometers, magnetometers, and a GPS. Two types of measurements are required to define orientation in three-dimensional space. In this case, gravity and bearing are sensed for measurement corrections. To make these adjustments, two rotations are performed in the form of correction quaternions. The gravity correction is applied first, followed by a subsequent bearing rotation that is constrained to only rotate the estimate about the inertial z -axis. These correctional rotations are constructed, so that the estimated gravity and bearing vectors align with the corresponding measurement update vectors.

5.2.1 Gravity Measurement

For the subsequent estimate ($\bar{\eta}_g$), the alignment of estimated and measured gravity vectors is performed with the composition of a gravity-based correction quaternion ($\bar{\eta}_{cg}$) with the kinematic estimate as

$$\bar{\eta}_g = \bar{\eta}_{cg} \otimes \bar{\eta}_k. \quad (5.3)$$

A description of how $\bar{\eta}_{cg}$ is constructed for such an adjustment will be given in this section.

The angle of rotation between the measured and estimated vectors can be found from the dot product as

$$\theta_{cg} = \cos^{-1} (G_m \cdot R(\bar{\eta}_k) G), \quad (5.4)$$

where G_m is the normalized acceleration vector or measurement of gravity obtained in the body reference frame, $G = (0, 0, 1)^T$ is the normalized inertial frame gravity vector, and $R(\bar{\eta}_k)$ is the rotation matrix that transforms from the inertial reference frame to the reference frame of the last kinematic estimate of attitude. This rotation matrix from a general quaternion ($\bar{\eta}$), is formed as

$$R(\bar{\eta}) = (\eta_4^2 - \eta^T \eta) I_3 + 2\eta \eta^T - 2\eta_4 \eta^\times. \quad (5.5)$$

Given θ_{cg} , the axis of rotation to align the measured and estimated gravity vectors is the cross product of the two in the estimated body reference frame as shown:

$$\hat{\eta}_{cg} = G_m \times R(\bar{\eta}_k) G. \quad (5.6)$$

The gravity based correction quaternion can now be defined as

$$\bar{\eta}_{cg} = \begin{pmatrix} \sin(k_{cg}\theta_{cg}) \hat{\eta}_{cg} \\ \cos(k_{cg}\theta_{cg}) \end{pmatrix}, \quad (5.7)$$

with k_{cg} being the acceleration dependent gain

$$k_{cg} = \frac{k_{cg-nom}}{1 + k_{penalty} |1 - \|A_m\||}. \quad (5.8)$$

Observe that k_{cg-nom} is the nominal gravity based correction gain, A_m is the accelerometer measurement in units of gravity, and $k_{penalty}$ is a gain that penalizes the gravity correction if the norm of the accelerometer measurement differs from grav-

ity. Thus, if accelerations not associated with gravity are measured, less weight is placed on the gravity correction update. Note that because estimated and measured vectors were compared in the reference frame of the previous kinematic estimate of attitude, the gravity based correction quaternion is applied second in the composition operation, as shown in equation (5.3).

5.2.2 Bearing Measurement

For the bearing measurement correction, the composition of the previous estimate ($\bar{\eta}_g$) with an inertial frame z -axis rotation ($\bar{\eta}_{cb}$) for bearing alignment, is applied as follows:

$$\bar{\eta}_b = \bar{\eta}_g \otimes \bar{\eta}_{cb}. \quad (5.9)$$

The method for producing $\bar{\eta}_{cb}$ will be derived in this section. Note that the inertial z -axis rotation is applied first in order to affect bearing only. Because the correction quaternion is an inertial-based rotation, heading vectors are compared in the inertial reference frame.

The declination of the magnetic field in the region of hardware testing has been found to be about 60 degrees, denoting a significant inertial z -component. Because the magnetic field vector is somewhat correlated with gravity, significant error in bearing can occur as a result of the misalignment of the estimated and actual gravity vectors. This occurs because the method (assuming no gravity error) projects the measurement onto the inertial xy -surface. If the two measurements (gravity and bearing) were perpendicular, this behavior would be largely eliminated. Figure 5.1 shows the affect that gravity error has on bearing. In this figure, the magnetic field vector was projected onto the inertial xy -surface, given error in the estimate of roll. The simulated aircraft was set level ($\theta = 0$ and $\phi = 0$) and rotated about the inertial z -axis for all possible heading orientations. The difference between the true and projected vectors is presented as the heading error. Observe that because the aircraft was level in this simulation, heading corresponds to bearing. As a direct result of this problem, course measured from GPS is used in level flight as a bearing measurement,

when typically error in roll occurs during turn maneuvers. Meanwhile, the bearing measurement is obtained from magnetometers in hover, where GPS course gives no indication of bearing. Note also that, in hover, resultant forces on the aircraft are generally small, producing in a reasonable estimate of gravity.

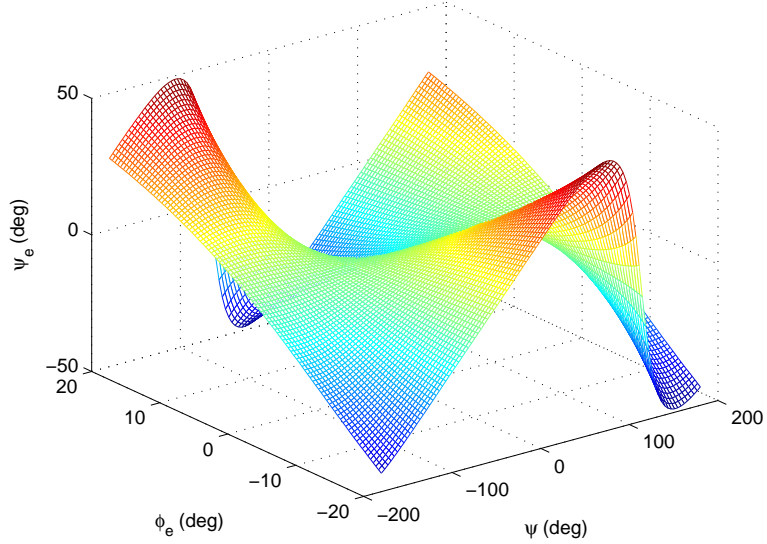


Figure 5.1: This figure shows the error in heading (ψ_e) induced from roll error (ϕ_e). Note that the simulated aircraft is level and thus heading error corresponds to error in bearing.

For the magnetometer based correction, the measurement obtained is rotated into the estimate of the inertial reference frame and then projected onto the xy -surface. This vector is compared to the nominal inertial frame magnetic field vector also projected onto the xy -surface. The angle for such a correction is found by

$$\theta_{cb} = \cos^{-1} \left(\frac{P_{xy} R(\bar{\eta}_g)^T M_m}{\|P_{xy} R(\bar{\eta}_g)^T M_m\|} \cdot \frac{P_{xy} M}{\|P_{xy} M\|} \right), \quad (5.10)$$

and the axis of rotation is

$$\hat{\eta}_{cb} = \frac{P_{xy} R(\bar{\eta}_g)^T M_m}{\|P_{xy} R(\bar{\eta}_g)^T M_m\|} \times \frac{P_{xy} M}{\|P_{xy} M\|}. \quad (5.11)$$

Observe that M_m denotes the magnetic field measurement obtained in the body reference frame, M is the inertial reference frame magnetic field vector, and P_{xy} is the simple projection matrix

$$P_{xy} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The correction quaternion can thus be defined as

$$\bar{\eta}_{cb} = \begin{pmatrix} \sin(k_{cb}\theta_{cb}) \hat{\eta}_{cb} \\ \cos(k_{cb}\theta_{cb}) \end{pmatrix}, \quad (5.12)$$

where k_{cb} is a gain.

The GPS course-based correction quaternion is constructed in a similar manner. The body frame x -vector is transformed to the inertial xy -surface and compared to the GPS course vector. The angle is given by

$$\theta_{cb} = \cos^{-1} \left(\begin{pmatrix} \cos \chi_m \\ \sin \chi_m \\ 0 \end{pmatrix} \cdot \frac{P_{xy}R(\bar{\eta}_g)^T X}{\|P_{xy}R(\bar{\eta}_g)^T X\|} \right), \quad (5.13)$$

and the axis of rotation is

$$\hat{\eta}_{cb} = \begin{pmatrix} \cos \chi_m \\ \sin \chi_m \\ 0 \end{pmatrix} \times \frac{P_{xy}R(\bar{\eta}_g)^T X}{\|P_{xy}R(\bar{\eta}_g)^T X\|}, \quad (5.14)$$

where χ_m is the measured course and X is the simple vector

$$X = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Equation (5.12) is then used as the final correction quaternion.

5.3 Results

In this section, results to validate the quaternion estimation method are discussed. Since estimation is not the main goal of the research and hardware results would require significant effort, only simulation results were obtained for method verification. Note that successful hardware results presented in Chapter 3 and Chapter 4 indicate that the method works reasonably well in hardware flight testing.

The path flown incorporated all tailsitter flight regimes and is shown in Figure 3.4. The path included a hover takeoff, hover waypoint, two level waypoints, a hover waypoint, and a hover land, in that order. Attitude results of the estimation technique are shown in Figure 5.2 with quaternion error between the estimated and actual attitude in Figure 5.3. Transitions of the flight occur at 5 seconds from hover to level flight and 35 seconds from level to hover. It can be observed that estimation of the actual attitude, based on available sensors, is fairly good. In level flight, error about the yaw axis from GPS course lag can be seen. This is indicated by η_3 in the error plot, during both turns at roughly 20 and 30 seconds. Roll error due to centripetal acceleration is shown during the second turn (see η_1 in Figure 5.3).

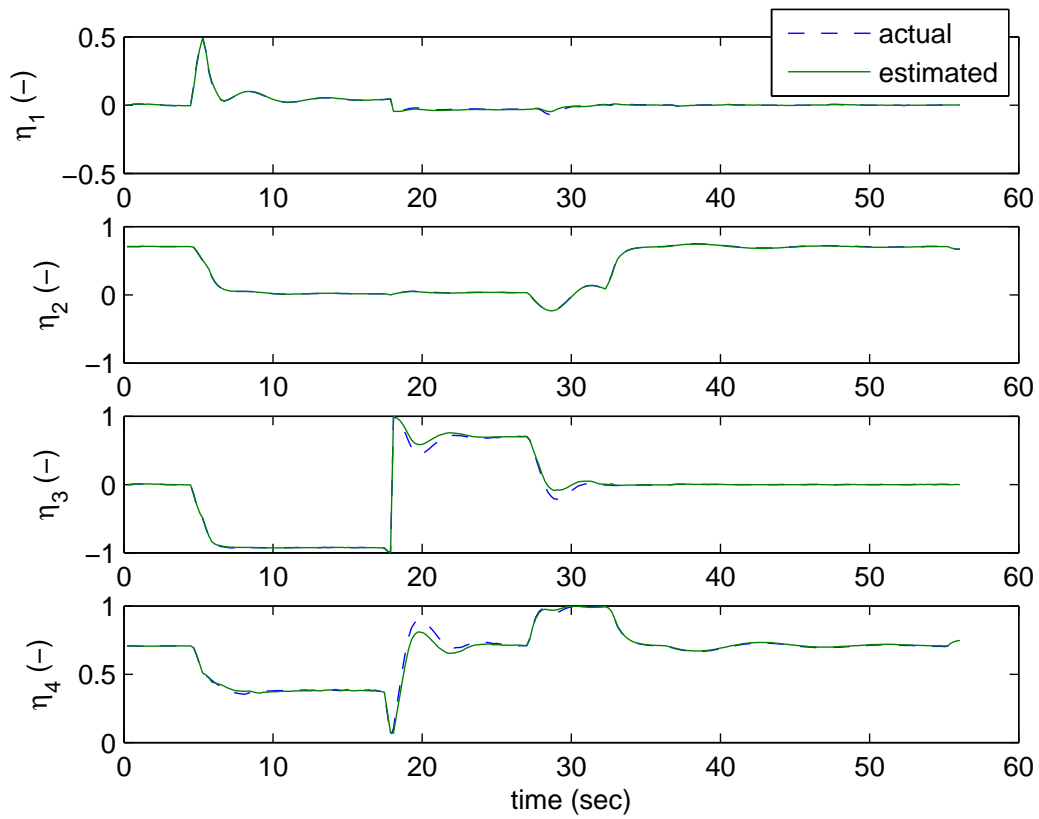


Figure 5.2: Attitude throughout the attitude estimation simulation experiment is shown.

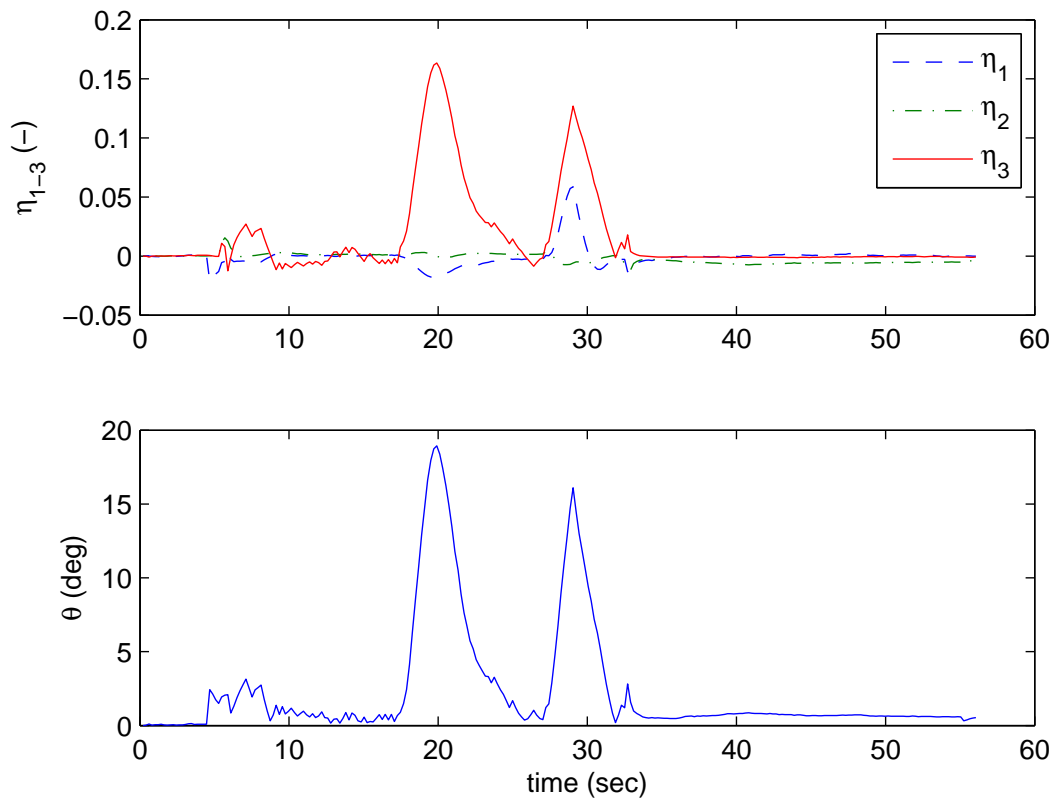


Figure 5.3: In this figure the attitude quaternion error throughout the attitude estimation simulation experiment is exhibited.

5.4 Summary

Quaternion attitude estimation was derived in this chapter. The scheme utilized both a model and measurement update applied with associated gains. For the model update, angular rates measured with gyros were integrated through exactly known quaternion kinematic equations. To prevent drift, a measurement update was then applied. This update incorporated the measurement of two vectors to fix the body-frame basis in three-dimensional space. Gravity from accelerometers and bearing from magnetometers or GPS course were the two measurement vectors. To adjust the estimate so that measurement vectors align, correction quaternions were constructed and applied.

Chapter 6

Conclusions and Recommendations

In this chapter, conclusions made throughout this research will be discussed. In addition, potential improvements and recommendations will be given for possible future work.

6.1 Conclusions

From the research that has been performed, many conclusions can be made. This section will present those findings and discuss their significance.

As the main goal of the research, novel quaternion-based attitude control methods for tailsitter flight have been developed. An adaptive backstepping controller was derived with two forms of on-line system parameter identification. The controller with a regularized data-weighted recursive least-squares algorithm exhibited effective learning of rapidly changing dynamics for model cancellation and stable tracking of reference model attitude. The backstepping controller presented was shown in results and theory to produce consistent performance for any given set of system parameters (with accurate system identification and no actuator saturation). As a result, gains for the technique were also shown to be universal for all airframe platforms with the three conventional control surface actuators (aileron, elevator, and rudder). A quaternion feedback controller with scheduled gains to accommodate changing tailsitter dynamics was also derived for a metric of baseline performance. The adaptive method described earlier, in simulation and hardware results, was comparable to and in general, better than the baseline design.

Unique tailsitter algorithms for navigating flight in hover, level, and transitional modes were generated. North/east position tracking was developed with an

original quaternion-based controller. A novel method of on-line thrust system identification was derived to address an inconsistent throttle system in hover altitude control. Quaternion based level flight path following and altitude/airspeed controllers were developed from adaptations of proven methods. Simple transitional trajectories between these modes were also produced. All of these algorithms were tested and proven in simulation and hardware environments. Results from these flight tests were discussed.

An original method for the estimation of quaternion attitude with improved sensors during hover flight was put forth in this research. The scheme incorporated several sensors in a fixed-gain Kalman-like filter. This method was used in the simulation and hardware flight testing of all algorithms discussed in this research. Moreover, simulation results for concept verification were presented.

Two methods of testing the tailsitter autopilot algorithms were produced from this research as well. Tailsitter simulation with a graphical user interface was generated and presented. This was used in the initial testing and assessment of techniques. In addition, for final hardware verification, a miniature tailsitter equipped with an autopilot system was also created.

In review, all of the unique methods developed in this research were proven to be effective at miniature tailsitter control. As an over-arching contribution of the work performed, a functional autonomous miniature VTOL tailsitter UAV system was produced with hardware results demonstrating its various capabilities.

6.2 Recommendations

Future work in the form of recommendations and possible areas of enhancement will be given in this section. Note that certain suggestions mentioned assume upcoming improvements in hardware.

In retrospect, the main challenge of this project was the airframe design used in flight testing. It is anticipated that considerable improvements in control performance during hover flight could be achieved with the elimination of airflow stagnation. This

would most likely be achieved by reducing the distance between the propeller and control surfaces.

Another area in which hardware improvements could be made is in sensing. With faster and more accurate position information, error in hover north/east tracking and level flight path following would be reduced. Derivative and integral position feedback control in hover flight could be implemented for enhanced performance. Furthermore, with the elimination of course delay, improved path following with diminished overshoot would be anticipated. Also, general improvements in both altitude and airspeed could be accomplished with more accurate feedback from the associated measurements.

For the enhancement of least-squares estimation of aerodynamic parameters, many sensors could be incorporated. Foremost is a measurement of angular acceleration, which would increase accuracy for improved knowledge about the actual system. Moreover, the addition of a measurement of the actual control surface deflections, along with improved velocity measurements, would both contribute to reduced error in the estimation of angular acceleration dynamics. Also, with angle of attack and sideslip angle sensors (which are currently too large and cumbersome for miniature aircraft), more information about contributions to torque would be available.

Further advances could be made in adaptive control with additional effort. With less model reduction, more parameters could be estimated (assuming increased computational and sensing capabilities). This would most likely allow for less data forgetting and enhanced accuracy in system parameter identification. In addition, methods other than backstepping for dynamics cancellation and stable quaternion control could be attempted.

Other general autopilot improvements are also possible with increased effort. Quaternion estimation through actual Kalman filtering would produce the best least squares estimate of attitude given the sensors available. The design of perch-and-stare, in-flight persistent imaging, precision landing, and optimized transitional maneuvers could be endeavored for improved trajectories. Also, an accurate model of

tailsitter dynamics would allow for enhanced development in autopilot control testing and design.

Appendix A

Quaternion Math

In this appendix, quaternion math is briefly discussed. The quaternion definition, given first, is followed by a short presentation of the time derivative. An operation for composing successive quaternion rotations is then exhibited. Finally, resolved with the composition operation, the error quaternion is discussed.

A.1 Quaternion Definition

The quaternion attitude representation can be understood as a single rotation about an axis in three-dimensional space [41]. The quaternion used to represent aircraft attitude is of the form [42]

$$\bar{\eta} = \begin{pmatrix} \eta \\ \eta_4 \end{pmatrix} = \left(\eta_1 \quad \eta_2 \quad \eta_3 \quad \eta_4 \right)^T. \quad (\text{A.1})$$

The terms η and η_4 can be described in the following manner:

$$\eta = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{pmatrix} = \sin \left(\frac{\Theta}{2} \right) \hat{\eta} \quad \eta_4 = \cos \left(\frac{\Theta}{2} \right),$$

where $\hat{\eta}$ is unit norm and describes the axis of rotation and Θ gives the scalar angle of rotation about that axis. To preserve the unity norm, the quaternion is scaled by the angle of rotation Θ as shown.

A.2 Quaternion Derivative

The time derivative of the quaternion can be written as [43, 42]

$$\dot{\eta} = \frac{1}{2}(\eta^\times + \eta_4 I_3)\omega \quad (\text{A.2})$$

and

$$\dot{\eta}_4 = -\frac{1}{2}\eta^T \omega, \quad (\text{A.3})$$

where the notation ξ^\times denotes the skew-symmetric matrix given by:

$$\xi^\times = \begin{pmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{pmatrix}.$$

A.3 Quaternion Composition

Quaternion composition can be defined as [42]

$$\bar{\eta}'' = \bar{\eta}' \otimes \bar{\eta} = \begin{pmatrix} \eta_4 \eta'_4 + \eta'_4 \eta - \eta' \times \eta \\ \eta'_4 \eta_4 - \eta'^T \eta \end{pmatrix}, \quad (\text{A.4})$$

for quaternion rotations, where $\bar{\eta}''$ is the result of two successive rotations represented by $\bar{\eta}$ and $\bar{\eta}'$. Equation (A.4) can also be written as [42]

$$\bar{\eta}'' = \bar{\eta}' \otimes \bar{\eta} = \{\bar{\eta}\}_R \bar{\eta}', \quad (\text{A.5})$$

where

$$\{\bar{\eta}\}_R = \begin{pmatrix} \eta_4 & -\eta_3 & \eta_2 & \eta_1 \\ \eta_3 & \eta_4 & -\eta_1 & \eta_2 \\ -\eta_2 & \eta_1 & \eta_4 & \eta_3 \\ -\eta_1 & -\eta_2 & -\eta_3 & \eta_4 \end{pmatrix}.$$

A.4 Quaternion Error

Consider the following equation,

$$\bar{\eta}_d = \bar{\eta}_e \otimes \bar{\eta}_a = \{\bar{\eta}_a\}_R \bar{\eta}_e, \quad (\text{A.6})$$

where $\bar{\eta}_d$ represents desired attitude, $\bar{\eta}_a$ represents actual attitude, and $\bar{\eta}_e$ represents the error between the two expressed in the actual attitude's body reference frame.

Noting that $\{\bar{\eta}\}_R^T \{\bar{\eta}\}_R = I_4$, we can resolve the error quaternion as

$$\bar{\eta}_e = \{\bar{\eta}_a\}_R^T \bar{\eta}_d, \quad (\text{A.7})$$

which also can be written as

$$\eta_e = -\eta_{4d}\eta_a + \eta_{4a}\eta_d - \eta_a^\times \eta_d \quad (\text{A.8})$$

and

$$\eta_{4e} = \eta_{4a}\eta_{4d} + \eta_a^T \eta_d. \quad (\text{A.9})$$

Bibliography

- [1] K. Wernicke, “The single-propeller driven tailsitter is the simplest and most efficient configuration for the VTOL UAV’s,” in *International Powered Lift Conference*, 2000. 4
- [2] A. Ailon, “Control of a vtol aircraft: Motion planning and trajectory tracking,” in *Mediterranean Conference on Control and Automation*, 2005, pp. 1493–1498. 4
- [3] P. Martin, S. Devasia, and B. Paden, “A different look at output tracking: control of a VTOL aircraft,” in *Conference on Decision and Control*, 1994, pp. 2376–2381. 4
- [4] R. Stone, “Control architecture for a tail-sitter unmanned air vehicle,” in *5th Asian Control Conference*, 2004, pp. 736–744. 4
- [5] —, “The T-wing tail-sitter unmanned air vehicle: from design concept to research flight vehicle,” in *Proceedings of the I MECH E Part G Journal of Aerospace Engineering*, 2004, pp. 417–433. 4
- [6] R. Stone and G. Clarke, “The T-wing: A VTOL UAV for defense and civilian applications,” in *UAV Australia Conference*, 2001. 4
- [7] —, “Optimization of transition maneuvers for a tail-sitter unmanned air vehicle (UAV),” in *Australian International Aerospace Congress*, 2001. 4
- [8] D. J. Taylor, M. V. Ol, and T. Cord, “Skytote advanced cargo delivery system,” in *AIAA/ICAS International Air and Space Symposium and Exposition: The Next 100 Years*, 2003. 4
- [9] B. Wei, H. Weiss, and A. Arapostathis, “Quaternion feedback regulator for spacecraft eigenaxis rotations,” *AIAA Journal of Guidance and Control*, vol. 12, no. 3, 1989. 4
- [10] S. M. Joshi, A. G. Kelkar, and J. T. Wen, “Robust attitude stabilization of spacecraft using nonlinear quaternion feedback,” *IEEE Transactions on Automatic Control*, vol. 40, no. 10, 1995. 4
- [11] R. Bach and R. Paielli, “Linearization of attitude-control error dynamics,” *IEEE Transactions on Automatic Control*, vol. 38, no. 10, Oct. 1993. 4
- [12] D. Shin and Y. Kim, “Reconfigurable flight control system design using adaptive neural networks,” *12*, no. 1, Jan. 2004. 5

- [13] R. L. Broderick, “Statistical and adaptive approach for verification of a neural-based flight control system,” in *Digital Avionics Systems Conference*, vol. 2, 24-28 Oct. 2004, pp. 6.E.1-1 – 6.E.1-10. 5
- [14] P. Melin and O. Castillo, “A new neuro-fuzzy-fractal approach for adaptive model-based control of non-linear dynamic systems: The case of controlling aircraft dynamics,” in *1999 IEEE International FUzzy Systems Conference Proceedings*, August 1999. 5
- [15] M. Steinberg and A. Page, “High-fidelity simulation testing of intelligent and adaptive aircraft control laws,” in *Proceedings of the American Control Conference*, 8-10 May 2002, pp. 3264–3268. 5
- [16] B. Kim and A. Calise, “Nonlinear flight control using neural networks,” *IEEE Transactions on Control Systems Technology*, vol. 20, no. 1, Jan.-Feb. 1997. 5
- [17] M. Bodson and J. Groszkiewicz, “Multivariable adaptive algorithm for reconfigurable flight control,” *IEEE Transactions on Control Systems Technology*, vol. 5, no. 2, Mar. 1997. 5
- [18] D. Shore and M. Bodson, “Flight testing of a reconfigurable control system on an unmanned aircraft,” in *Proceedings of the 2004 American Control Conference*, June-July 2004, pp. 3747–3752. 5
- [19] B. Porter and C. Boddy, “Design of adaptive digital controllers incorporating dynamic pole-assignment compensators for high-performance aircraft,” in *Proceedings of the IEEE 1989 Aerospace and Electronics Conference*, 22-26 May 1989, pp. 372–379. 5
- [20] N. Knoebel, S. Osborne, J. Matthews, and R. Beard, “Computationally simple model reference adaptive control for miniature air vehicles,” in *American Control Conference*, June 2006. 5
- [21] J. Matthews, N. Knoebel, S. Osborne, and R. Beard, “Adaptive backstepping control for miniature air vehicles,” in *American Control Conference*, June 2006. 5
- [22] J. D. Boskovic, L. Chen, and R. K. Mehra, “Adaptive control design for non-affine models arising in flight control,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 27, no. 2, pp. 209–217, March–April 2004. 5
- [23] G. Tao, S. Chen, J. Fei, and S. Joshi, “An adaptive actuator failure compensation scheme for controlling a morphing aircraft model,” in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Dec. 2003, pp. 4926–4931. 5
- [24] S. Chen, G. Tao, and S. Joshi, “An adaptive actuator failure compensation controller for mimo systems,” in *Proceedings of the 41st IEEE Conference on Decision and Control*, December 2002, pp. 4778–4783. 5

- [25] J. Farrell, M. Polycarpou, and M. Sharma, “Adaptive backstepping with magnitude, rate, and bandwidth constraints: Aircraft longitude control,” in *Proceedings of the American Control Conference*, June 2003, pp. 3898–3904. 5
- [26] R. Beard, N. Knoebel, C. Cao, N. Hovakimyan, and J. Matthews, “An \mathcal{L}_1 adaptive pitch controller for miniature air vehicles,” in *AIAA Guidance, Navigation, and Control Conference*, 21-24 August 2006, pp. 5788–5794. 5
- [27] C. Cao and N. Hovakimyan, “Design and analysis of novel \mathcal{L}_1 adaptive control architecture with guaranteed transient performance, part 1: Control signal and asymptotic stability,” in *Proceedings of the 2004 American Control Conference*, 2006. 5
- [28] —, “Design and analysis of novel \mathcal{L}_1 adaptive control architecture with guaranteed transient performance, part 2: Guaranteed transient response,” in *Proceedings of the 2004 American Control Conference*, 2006. 5
- [29] K. Subbarao, M. Steinberg, and J. Junkins, “Structured adaptive model inversion applied to tracking aggressive aircraft maneuvers,” in *AIAA Guidance, Navigation, and Control Conference*, 6-9 August 2001. 5
- [30] J. Valasek, M. Tandale, and J. Rong, “A reinforcement learning - adaptive control architecture for morphing,” *Journal of Aerospace Computing, Information, and Communication*, vol. 2, April 2005. 5
- [31] W. F. Phillips, *Mechanics of Flight*. New Jersey: Wiley, 2004. 14
- [32] <http://www.hobby-lobby.com/pogo.htm>. 21
- [33] <http://www.aero-model.com/motordetails.aspx?series=A20&style=L>. 21
- [34] http://www.aeromicro.com/Catalog/castle_creations_phoenix-25_esc_now_with_3-amp_bec_2462852.htm. 21
- [35] <http://www.hobby-lobby.com/thunderpower.htm>. 21
- [36] <http://procerusuav.com/>. 24
- [37] <http://www.abacuscity.ch/abashop?i=gwNSv4yg76iBCfoADswO&s=142&p=productdetail&sku=54>. 24
- [38] <http://www.sarantel.com/products/>. 24
- [39] <http://www.pnicorp.com/productDetail?nodeId=cMM3>. 24
- [40] D. Nelson, D. Barber, T. McLain, and R. Beard, “Vector field path following for small unmanned air vehicles,” in *American Control Conference*, June 2006. 95
- [41] W. Phillips, C. Hailey, and G. Gebert, “A review of attitude representations used for aircraft kinematics,” *Journal of Aircraft*, vol. 38, no. 4, 2001. 123

- [42] M. Shuster, "A survey of attitude representations," *Journal of the Astronautical Sciences*, vol. 41, no. 4, 1993. 123, 124
- [43] B. T. Costic, D. M. Dawson, M. S. de Queiroz, and V. Kapila, "Quaternion-based adaptive attitude tracking controller without velocity measurements," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 6, 2001. 124