Brigham Young University
BYU ScholarsArchive

All Theses and Dissertations

2009-11-24

Vision-Based Precision Landings of a Tailsitter UAV

Paul Travis Millet
Brigham Young University - Provo

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

Part of the Mechanical Engineering Commons

BYU ScholarsArchive Citation

Millet, Paul Travis, "Vision-Based Precision Landings of a Tailsitter UAV" (2009). *All Theses and Dissertations*. 1961.
https://scholarsarchive.byu.edu/etd/1961

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.

Vision-Based Precision Landings of a Tailsitter UAV

P. Travis Millet

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Timothy W. McLain, Chair
Randy W. Beard
Mark B. Colton

Department of Mechanical Engineering

Brigham Young University

April 2010

Copyright © 2009 P. Travis Millet

All Rights Reserved

ABSTRACT

Vision-Based Precision Landings of a Tailsitter UAV

P. Travis Millet

Department of Mechanical Engineering

Master of Science

We present a method of performing precision landings of a vertical take-off and landing (VTOL) unmanned air vehicle (UAV) with the use of an onboard vision sensor and information about the aircraft's orientation and altitude above ground level (AGL). A method for calculating the 3-dimensional location of the UAV relative to a ground target of interest is presented as well as a navigational controller to position the UAV above the target. A method is also presented to prevent the UAV from moving in a way that will cause the ground target of interest to go out of view of the UAV's onboard camera. These methods are tested in simulation and in hardware and resulting data is shown. Hardware flight testing yielded an average position estimation error of 22 centimeters. The method presented is capable of performing precision landings of VTOL UAV's with submeter accuracy.

Keywords: vision-based navigation, precision landing, unmanned tailsitter, VTOL, UAV

ACKNOWLEDGMENTS

I'd like to thank the members of my graduate committee, Dr. McLain, Dr. Beard, and Dr. Colton, for all their excellent help, suggestions, and contributions to this research. I'd also like to thanks the many members of the BYU MAGICC lab that have helped to contribute to this research. In particular, I'd like to thank my predecessor, Nate Knoebel, for his contributions and well documented work in autonomous tailsitter control. I'd like to thank Neil Johnson for taking me under his wing, (or rotor blade seems more appropriate), and mentoring me in the beginning of this research. Much thanks also to Bryce Ready for hours spent at the whiteboard hammering out the more elusive details of this work with me. My thanks also go out to the many MAGICC lab researchers including Rajnikant Sharma, Brett Millar, Liang (Solomon) Sun, Huili Yu, Jacob Bishop, Jeff Saunders, James Hall, Dan Carlson, and the many others that have been great colleagues and friends. An enormous thanks to the Wright-Patterson AFB μAVIARI operations, specifically Capt. Ryan Carr and Doug Szczublewski, for their hard work, help, and sacrifice in acquiring the hardware flight data presented in this work. Lastly, I'd like to thank my wife Michelle and my son Peter for showering me with laughter and love while I was in the pit of research frustrations, and for basking with me in the ecstacy of hardware results.

TABLE OF CONTENTS

LIST OF TABLES . . . . **vi**

LIST OF FIGURES . . . . **viii**

NOMENCLATURE . . . . **x**

Chapter 1 Introduction **1**
1.1 Motivation 1
1.2 Background 3
1.3 Contributions 4
1.4 Document Organization 4

Chapter 2 UAV Testbed **7**
2.1 UAV Simulation Environment 7
2.2 Flight Test Setup 8
2.2.1 Airframe 9
2.2.2 Vision System 10
2.2.3 Attitude and Altitude Estimation System 10
2.2.4 Ground Station Software 11

Chapter 3 Aircraft Stabilization, Altitude, and Navigation Control **13**
3.1 Attitude Controller 13
3.1.1 Simulation Attitude Controller 14
3.1.2 Hardware Attitude Controller 15
3.1.3 Attitude Controller Results 17
3.2 Altitude Controller 19
3.2.1 Simulation Altitude Controller 20
3.2.2 Hardware Altitude Controller 21
3.2.3 Altitude Controller Results 22
3.3 Navigation Controller 23
3.3.1 Hardware Navigation Controller 24
3.3.2 Navigation Controller Results 26

Chapter 4 Relative Position Estimation **29**
4.1 Coordinate Frames 29
4.2 Pinhole Camera Model 31
4.3 Target Detection 32
4.3.1 Color Segmentation 33
4.3.2 Connected Components 33
4.3.3 Centroid Computation 35
4.4 Coordinate Transformations 35
4.5 Results 36

4.5.1 Position Estimation Simulation Results 37
4.5.2 Position Estimation Hardware Results 39

Chapter 5 Desired Attitude Saturation 43
5.1 Desired Attitude Saturation in Quaternions 45
5.2 Desired Attitude Saturation in Euler Angles 49
5.2.1 Pitch Saturation Angles 49
5.2.2 Roll Saturation Angles 51
5.3 Navigation and Cushion Zones 52
5.4 Results 53
5.4.1 Attitude Saturation Simulation Results 53
5.4.2 Attitude Saturation Hardware Results 54

Chapter 6 Conclusions and Recommendations 61
6.1 Conclusions 61
6.2 Recommendations for Future Work 62

REFERENCES 65

Appendix A Derivation of the Desired Quaternion 67

Appendix B Simulation Forces, Torques, and Equations of Motion 71
B.1 Propeller Forces and Torques 71
B.2 Propeller Wash Forces and Torques 73
B.2.1 Aileron 74
B.2.2 Elevator 75
B.2.3 Rudder 76
B.3 Level Flight Forces and Torques 77
B.4 Equations of Motion 79

LIST OF TABLES

4.1 Relative position estimation flight results 41

5.1 Attitude saturation flight results 59

B.1 Aircraft physical parameters 72
B.2 Level flight coefficients 78

LIST OF FIGURES

1.1 Examples of Unmanned Tailsitters 2

2.1 Simulated Video used to Simulate Vision-Based Landings 7
2.2 Schematic of Controller used in Simulation 8
2.3 Helicopter Testbed 9
2.4 Vision System on Helicopter Testbed 10
2.5 Ground Station Video Hardware 11

3.1 Block Diagrams of PID Attitude Controllers used in Flight Testing. 16
3.2 Comparison of Desired and Actual Attitude States from Simulation Flight Tests. . . 17
3.3 Error between Aircraft's Attitude and Desired Attitude in Hardware Flight Testing. 18
3.4 Block Diagrams of PID Attitude Controllers used in Flight Testing. 19
3.5 Error between Aircraft's Attitude and Desired Attitude in Hardware Flight Testing. 20
3.6 Block Diagram of Altitude Controller used in Simulation. 21
3.7 Block Diagram of Altitude Controller used in Flight Testing. 22
3.8 Altitude Simulation Results. 23
3.9 Altitude Flight Test Results. 24
3.10 Block Diagram of Navigation x-Direction Controller used in Flight Testing. 25
3.11 Block Diagram of Navigation y-Direction Controller used in Flight Testing. 25
3.12 Desired and Actual Location of Aircraft in Simulation. 26
3.13 Desired and Actual Location of Aircraft from Flight Testing. 27

4.1 Inertial Based Reference Frames 30
4.2 Several Reference Frames used in Relative Position Estimation. 30
4.3 Pinhole Camera Model 31
4.4 Connected Components of a Thresholded Image. 34
4.5 Flight Path used in Simulation and Hardware Flight Testing. 37
4.6 Simulation Results of Vision-Based Relative Position Estimation Algorithm. 38
4.7 Hardware Results of Vision-Based Relative Position Estimation Algorithm. 39

5.1 Target Centroid in the Image Frame from Motivating Example. 44
5.2 UAV Position from Motivating Example. 45
5.3 View Frustum of Tailsitter. 46
5.4 Angles and Distances used for Attitude Saturation Computation. 48
5.5 Maximum and Minimum Euler Angles used to Keep Ground Target in View. 50
5.6 Image Cushion and Navigation Zones. 52
5.7 Desired and Saturated Desired Quaternion Components from Simulation. 54
5.8 Location of Target Centroid in Image Frame from Simulation. 55
5.9 Location of Target Centroid in Image Frame from Flight Testing. 56
5.9 Location of Target Centroid in Image Frame from Flight Testing. 57
5.10 Location of Target Centroid in Image Frame if the UAV Tracked the Desired Attitude. 58
5.11 An Example of Pitch Saturation in Flight Testing. 58

B.1 Sketch of Tailsitter Geometry and Nomenclature 72

NOMENCLATURE

X_{UAV}	Calculated x location of the UAV in the target coordinate frame
Y_{UAV}	Calculated y location of the UAV in the target coordinate frame
q_{aircraft}	Quaternion describing the aircraft's orientation expressed in the inertial frame
q_{desired}	Quaternion describing the desired aircraft orientation expressed in the inertial frame
q_{error}	Quaternion describing rotation from q_{aircraft} to q_{desired} expressed in the body frame
α_{az}	Gimbal azimuth angle
α_{el}	Gimbal elevation angle
R_v^b	Vehicle frame to body frame rotation matrix
R_b^g	Body frame to gimbal frame rotation matrix
R_g^{g1}	Gimbal frame to gimbal-1 frame rotation matrix
R_{g1}^{g2}	Gimbal-1 frame to gimbal-2 frame rotation matrix
R_{g2}^c	Gimbal-2 frame to camera frame rotation matrix
K	Camera calibration matrix
x_{gtrans}	X translation from gimbal frame to body frame as measured in the gimbal frame
y_{gtrans}	Y translation from gimbal frame to body frame as measured in the gimbal frame
z_{gtrans}	Z translation from gimbal frame to body frame as measured in the gimbal frame
X_{pixel}	X centroid of the target in the image frame
Y_{pixel}	Y centroid of the target in the image frame
f_x	Image focal length in the image frame x direction
f_y	Image focal length in the image frame y direction
c_x	Image center in the image frame x direction
c_y	Image center in the image frame y direction
p	Roll rate of the aircraft
q	Pitch rate of the aircraft
r	Yaw rate of the aircraft
δ_a	Aileron command
δ_e	Elevator command
δ_r	Rudder command

CHAPTER 1. INTRODUCTION

Tailsitter aircraft exhibit many potential advantages over fixed-wing and helicopter aircraft due to their ability to fly in both hover flight and energy-efficient forward flight. The capability to perform precision landings at a desired location allows a tailsitter to utilize many of the advantages that a tailsitter aircraft has over conventional fixed-wing aircraft. Motivation for this research topic is given, followed by a description of previous research done in this area. The contributions presented by this work are offered, followed by a description of the organization of this document.

1.1 Motivation

Tailsitter aircraft have the capability to hover, allowing them to perform perch-and-stare, descend-and-investigate, and drop/retrieve missions. This capability to hover also allows tailsitter aircraft to take off and land from a variety of locations without the need of a runway. Tailsitter aircraft are also able to transition into level flight where lower energy consumption and faster speeds can be achieved [1]. The versatility of taking off and landing without a runway and of being able to conserve energy in level flight makes the tailsitter aircraft an appealing candidate for many military and civilian tasks. However, the ability to perform precision hover landings at a desired location is paramount to many of the aforementioned tasks. This work focuses on the precision landing of tailsitter unmanned air vehicles (UAVs) with the use of an onboard camera, an estimate of the aircraft's orientation and an estimate of the aircraft's altitude above ground level (AGL).

The generally accepted method of determining UAV position in an outdoor environment is to use a global positioning system (GPS) device. Conventional GPS provides a reliable method of determining the location of a UAV with an accuracy of 5 to 20 meters for commercial grade GPS [2]. For applications with a required landing precision of less than 1 meter, commercial grade GPS units cannot provide a sufficiently accurate measurement. Differential GPS units are

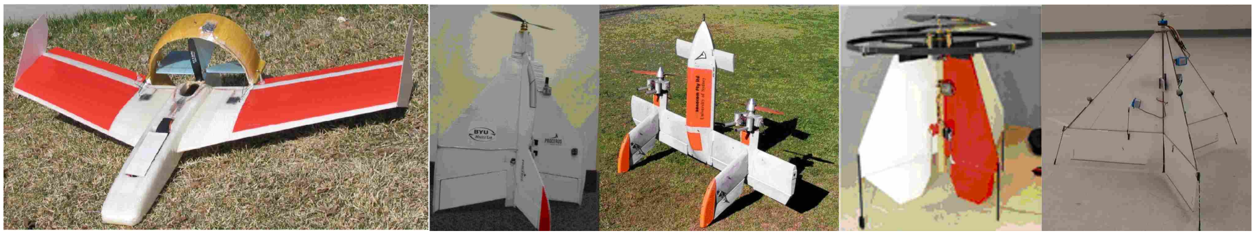


Figure 1.1: Examples of unmanned tailsitter aircraft, from left to right are; the current BYU tailsitter airframe, the previous BYU tailsitter airframe, the T-Wing used at the University of Syndey, the VERTIGO aircraft used at the Institut Supérieur de l'Aéronautique et de l'espace (ISAE), a tailsitter design used at Massachusetts Institute of Technology (MIT).

capable of achieving submeter precision [3], however, differential GPS units are larger, heavier, and more expensive than conventional GPS and require communication with a stationary ground unit at a location close to the landing site. The size and weight of differential GPS units make them infeasible on small, space-limited UAVs and weight-limited vertical take off and landing (VTOL) UAVs. As we will show, a small and lightweight onboard camera is a viable alternative to GPS for providing the UAV position estimation information necessary for precision landings.

Vision-guided precision landings are made possible through the use of UAV orientation information, the UAV's altitude AGL, and the pixel coordinates of a ground target in the image frame of a camera onboard the UAV. With this information the position of the aircraft relative to a ground target can be calculated. With the relative position between the UAV and the target known, the aircraft can then be navigated to land on top of the target or to any other position at which the target is still visible to the UAV's camera.

One of the difficulties of performing the vision-based landing of a tailsitter in hover flight with a fixed, downward facing camera is that, as the tailsitter tilts to move toward the target, there is no guarantee that the target will remain in the image frame. If the target leaves the image frame then no further vision-based position estimates can be calculated, resulting in an ever increasing position error. Using the technique described in this paper, it is critical that the time the target is not in the camera view frustum be minimized. For this reason a method of limiting the UAV's desired attitude is also presented in order to keep the target in view at all times.

1.2 Background

This paper builds on work done by Knoebel et al. [4–6] at the Brigham Young University Multiple Agent Intelligent Coordination and Control (MAGICC) Lab. Researchers at the BYU MAGICC Lab developed an adaptive controller for tailsitter UAVs in both vertical and horizontal modes of flight. They demonstrated the controller's effectiveness in level and vertical flight as well as in performing transitions between these two regimes. Figure 1.1 shows the tailsitter airframes used for previous flight testing at BYU as well as a sampling of other tailsitter airframes used for research by various universities.

Research into vision-based landings of fixed-wing aircraft has been explored by researchers at Brigham Young University. Barber et al. [7, 8] used vision to estimate a desired motion vector used to land a UAV on a ground target. Vision was also used to estimate bias which came from camera misalignment, wind, or state estimation errors. This controller was used to land a fixed-wing UAV on static and moving targets, both with successful results.

Hintze and Theodore et al. [9–12] demonstrated the use of vision-based methods to land a rotorcraft UAV in a GPS-denied environment. They used an onboard camera, a laser range unit and an inertial measurement unit (IMU) to estimate the UAV's position relative to a fixed point on the ground. This information was used to navigate the rotorcraft to the ground without the use of GPS. The method was reported to be accurate enough to serve as an alternative to GPS data in landing maneuvers.

Saripalli et al. [13] used a similar approach to the one we present in this paper to land a helicopter on top of a marked landing pad. They used color thresholding to find the landing pad and computed its center of mass and orientation. This information was then fused with GPS information to refine the position estimate of the aircraft. The helicopter would then align itself with the orientation of the landing pad and position itself over the target. Additionally, the aircraft was capable of performing an autonomous search for the landing pad as well as tracking a moving target and landing on it after the target had remained fixed for a period of time.

Researchers at the University of California at Berkeley [14, 15] have performed vision-based landings and navigation of a helicopter by computing the full pose of the helicopter using linear and nonlinear optimization techniques and a uniquely defined ground target. The solution of the linear optimization of the pose of the aircraft was used to initialize a nonlinear optimization to find a robust pose estimate of the UAV. This pose estimate was then combined with those derived from the GPS and inertial navigation sensors (INS) sensors to calculate a more accurate estimate of the aircraft's pose. The helicopter was given a greater range of motion by gimballing the camera to not lose sight of the target used for navigation [16].

How et al. [17] at the Massachusetts Institute of Technology Aerospace Controls Laboratory have successfully performed precision control on a fixed-wing aerobatic RC model airplane using an offboard VICON MX motion capture system. Hover flight, level flight, transitions, and precision landings were performed using adaptive control and the VICON MX motion capture system. Because the vision system is offboard, the UAV is confined to areas equipped with precision motion capture systems to acquire UAV position information.

1.3 Contributions

Much of the research presented here is shown as related to tailsitter aircraft, however, much of this work could be applied to other vertical take off and landing (VTOL) airframes or to vision-based navigation in general. The contributions made by this work are as follows:

- A method for vision-based landings of tailsitter aircraft is developed using quaternions as rotation operators
- A method for estimating an aircraft's position relative to a ground target using the aircraft's orientation, altitude AGL, and an onboard camera is shown
- A novel method used to saturate an aircraft's desired attitude is provided such that a ground target being used for visual navigation does not leave the image frame
- Simulation and hardware flight testing is used to validate the methods presented

1.4 Document Organization

Chapter 2 of this document describes the UAV testbed used for flight testing as well as the simulation environment used to test the methods presented in this work prior to hardware flight testing. Chapter 3 describes the attitude, altitude and navigational controller used to stabilize the aircraft and navigate the UAV to desired locations. Chapter 4 gives a brief overview of the pinhole camera model and details the relative position estimation algorithm used to estimate the UAV's position relative to a ground target of interest. Chapter 5 describes the novel technique used to keep the ground target in the UAV's view frustum so that future position estimates can be acquired and navigation to the ground can be achieved. In Chapter 6 we present our conclusions and recommendations for future work.

CHAPTER 2. UAV TESTBED

The vision-based landing research presented in this work was validated via computer simulation and by performing vision-based landings on a VTOL UAV. This chapter describes the simulation environment used to test the precision landing method and the hardware setup used in performing vision-based landings in hardware flight testing.

2.1 UAV Simulation Environment

Computer simulation of tailsitter vision-based landings was performed to gain insight into the expected performance of the proposed vision guidance controller as well as to identify unforeseen difficulties in performing vision-based landings. Simulation was performed in the Matlab Simulink environment using simulated video streamed from the open source Aviones aircraft simulator [18]. Simulated video from the Aviones environment is shown in Figure 2.1. The scene rendered by Aviones is based on the aircraft states passed to the Aviones visualizer from the Matlab simulator and is the scene that a camera facing out the tail end of the aircraft would see from onboard the UAV. The Aviones video was then processed in Matlab to determine the aircraft's relative position to the target.

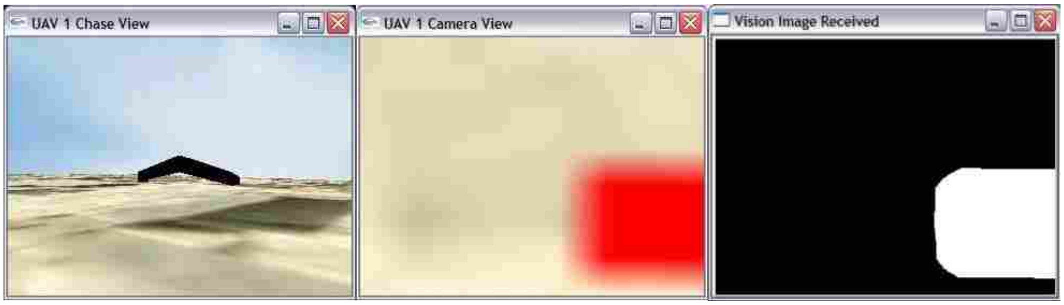


Figure 2.1: Video screens from the Aviones environment used in Matlab simulation: the left pane shows the orientation of the UAV, the middle pane shows the scene from the onboard camera, the right pane is the image from the onboard camera after color segmentation.

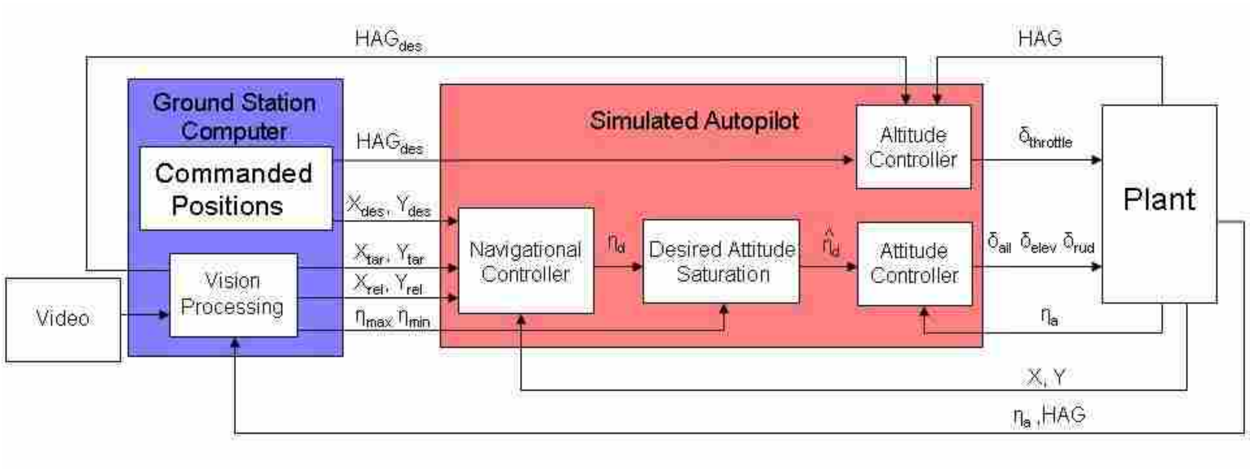

HAG_{des}
HAG
Ground Station Computer
Commanded Positions
HAG_{des}
Simulated Autopilot
Altitude Controller
$\delta_{throttle}$
Plant
X_{des}, Y_{des}
Navigational Controller
η_d
Desired Attitude Saturation
$\hat{\eta}_d$
Attitude Controller
$\delta_{ail}\ \delta_{elev}\ \delta_{rud}$
X_{tar}, Y_{tar}
X_{rel}, Y_{rel}
Video
Vision Processing
$\eta_{max}\ \eta_{min}$
η_a
X, Y
η_a ,HAG

Figure 2.2: Schematic of the controller used in simulation.

The block diagram shown in Figure 2.2 shows the simulation structure used to simulate the vision-based landing technique presented in this paper. The navigational controller shown in Figure 2.2 receives desired positions from both the user defined waypoints as well as from the vision processing program. The desired waypoint locations from the vision processing block override the user-defined waypoints if the ground target of interest is detected in the video frame. The forces and moments acting on the aircraft as well as the equations of motion used to simulate tailsitter aircraft dynamics are described in detail in Appendix B as described in [6].

2.2 Flight Test Setup

The vision-based landing technique described in this paper was tested in hardware using an autonomous helicopter UAV equipped with onboard vision. While the research of vision-based precision landings presented in this work is derived for precision landings of tailsitter aircraft, this work extends beyond tailsitter airframes to the vision-based precision landings of VTOL aircraft in general. We feel that the use of a helicopter in hardware flight testing only further strengthens the contributions of this research in the task of vision-based landings of VTOL platforms. This section also describes the airframe, sensors, and the VICON MX motion capture system used to estimate the UAV's attitude and altitude, that were used to validate the tailsitter vision-based landing techniques described in this document.

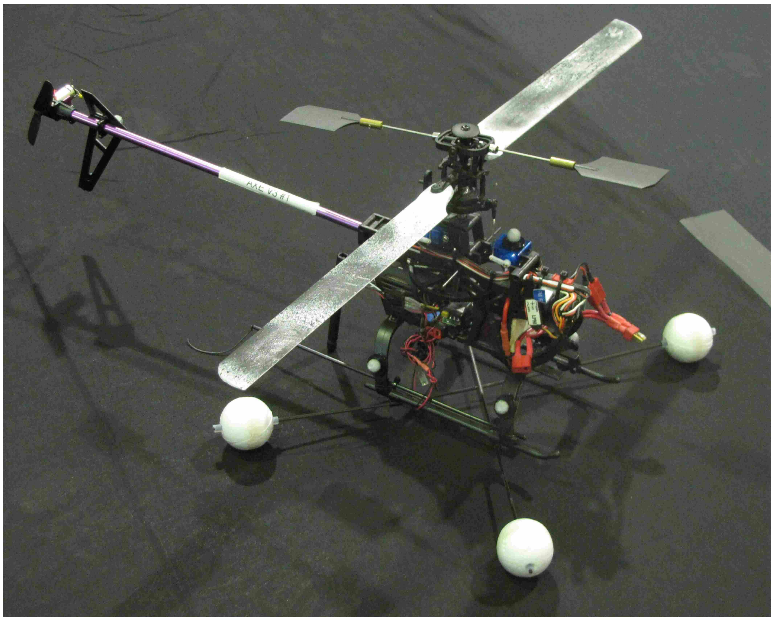


Figure 2.3: Helicopter testbed used for flight testing.

2.2.1 Airframe

The testbed used to validate this research was an off-the-shelf HeliMax CPv3 remote controlled (RC) helicopter shown in Figure 2.3. The helicopter has a 20.75 inch rotor span and has a total weight of one pound. The helicopter uses collective rotor pitch for altitude control and cyclic rotor pitch for both roll and pitch control. A tailrotor at the end of the helicopter's tail boom controls the yaw of the aircraft. The helicopter is driven by a single 380 size brushed motor and powered by an Electrifly 11.1 volt, 950 mA-hr lithium polymer battery. The motor is controlled by a 12 amp electronic speed control also made by Electrifly.

The helicopter was controlled using a 5-channel 72 Mhz transmitter and an onboard receiver. Commands were sent from the ground station to the onboard servos and speed control through the transmitter either by the control algorithms running on an offboard computer or by the safety pilot. Helicopter control was sent to the RC transmitter from the offboard computer via a trainer cable that connected into the back of the transmitter. The safety pilot was able to switch between computer-in-control (CIC) mode and pilot-in-control (PIC) mode using a toggle switch located on the RC transmitter. This allowed the safety pilot to take control in the event of an emergency.

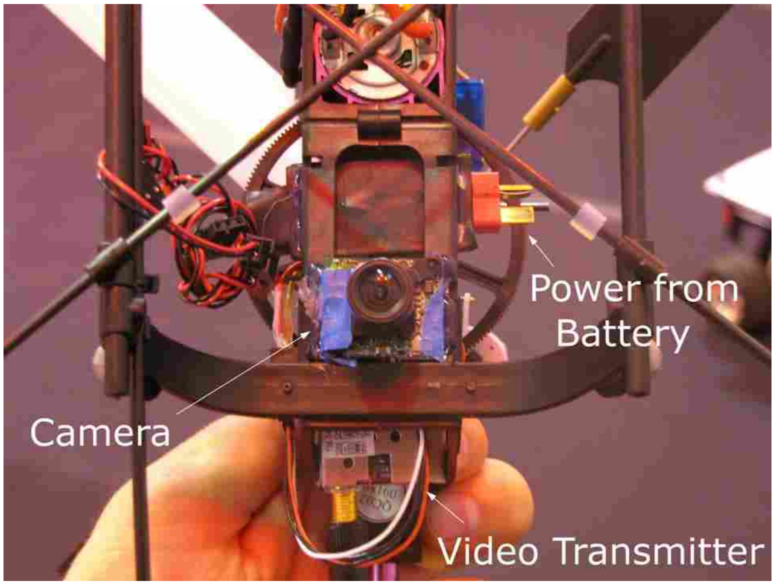


Figure 2.4: Video system mounted on the helicopter testbed used for flight testing.

2.2.2 Vision System

A RHPC-2005 1/4 inch charge-coupled-device (CCD) OEM color camera was mounted on the helicopter to provide onboard video. The camera required five volts of DC power and provided 640 by 480 pixels of resolution that was transmitted to the ground station using a 500 mW analog video transmitter. The video transmitter and the camera in its downward facing position, can be seen mounted on the helicopter airframe in Figure 2.4.

Video sent from the UAV was received at the ground station using a 12 volt, eight channel video receiver made by Black Widow AV. A KWorld 2820 framegrabber was used to convert the analog video into a digital format for processing on the ground station computer. The video receiver and the framegrabber used in testing can be seen in Figure 2.5.

2.2.3 Attitude and Altitude Estimation System

The aircraft's position and orientation were provided by a 35 camera VICON MX motion capture system at the Wright Patterson Air Force Base (WPAFB) Micro Air Vehicle Integration Application Research Institute (μAVIARI) where the flight tests were performed. The VICON MX motion capture system is a commercial off-the-shelf package that used multiple cameras to locate small reflective markers attached to the airframe of the helicopter. The cameras are calibrated such

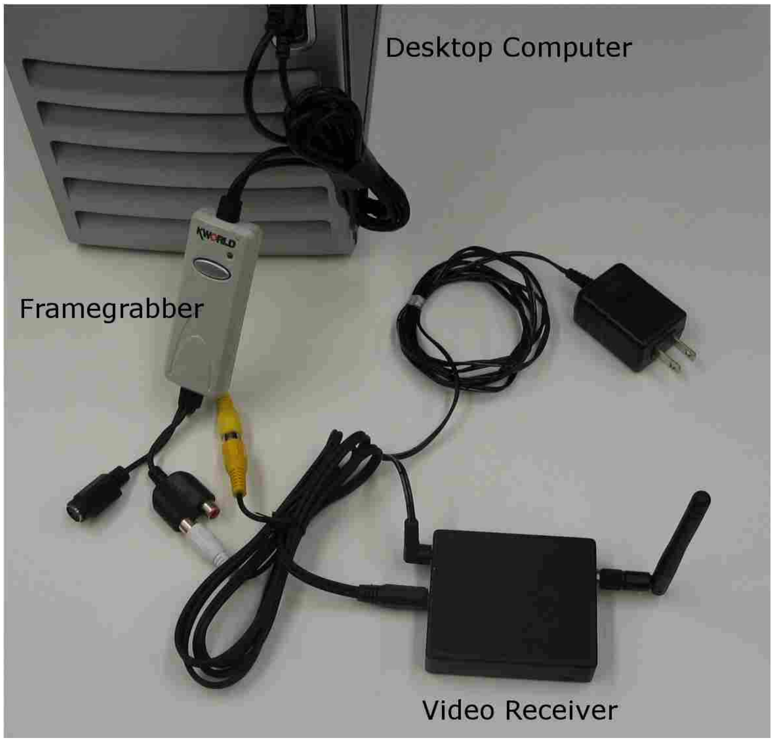


Figure 2.5: Video hardware used at the ground station to acquire video from the UAV and convert it to a digital signal for processing on the ground station computer.

that the position of each camera relative to the other cameras observing the airframe is known. A body frame coordinate system is defined in software using the reflective markers. Using the location of these reflective markers as seen from the multiple cameras and the known locations of each of the VICON cameras, an estimate of the aircraft's attitude and position can be calculated.

The estimates of the aircraft's attitude and altitude were used to perform the autonomous vision-based landings as described in Chapters 4 and 5. The x and y positions of the UAV as derived from the VICON system were not used to perform the vision-based landings. They were, however, used in the control of the helicopter before the target was acquired and when the target was not found in the image frame.

2.2.4 Ground Station Software

The aircraft position and attitude estimation, and the aircraft control algorithms were run on a Boxx 3.4Ghz workstation computer. The control algorithms were implemented using LabView version 8.6 and the vision processing was done in C++ using the OpenCV computer vision library [19].

CHAPTER 3. AIRCRAFT STABILIZATION, ALTITUDE, AND NAVIGATION CONTROL

This chapter details the controllers used to orient the aircraft at a commanded attitude and altitude. Once we were able to realize commanded orientations, we were then able to navigate to desired locations by commanding a desired attitude that would take us to the desired location by means of a navigational controller. There are slight differences between the controllers used in simulation and the controllers used in hardware for reasons explained in this chapter, though the basic architecture remains the same. Attitude, altitude, and navigation controllers are presented in this chapter as well as data showing the response of the aircraft when the controllers were used to track a desired input in simulation and hardware flight testing.

3.1 Attitude Controller

For an aircraft to maintain steady hover flight, it must be able to hold a desired attitude. Also, to navigate to desired locations and stay at a desired location, as is our goal with vision-based landings, the attitude controller must be able to effectively track a time-varying desired attitude. This section details the controller used to orient the aircraft at a time-varying desired attitude so that vision-based landings could be achieved.

One of the difficulties of working with tailsitter aircraft is that the conventional 3-2-1 rotational sequence used to describe an aircraft's orientation using Euler angles proves to be inadequate when used to describe the orientation of tailsitter aircraft. This is because the Euler angle used to describe an aircraft's pitch in the 3-2-1 rotation sequence exhibits a singularity as the aircraft's pitch nears an angle of $\pm\pi/2$, which is the orientation of a tailsitter in hover flight. For this reason quaternions were primarily used to describe vehicle orientation in this work. A thorough explanation of quaternions as rotation operators is described by Kuipers in [20]. The notation for quaternions used throughout this work is of the form

$$\eta_{(\cdot)} = \mathbf{u}\sin(\alpha/2) + \cos(\alpha/2) = \begin{bmatrix} u_{1(\cdot)}\sin(\alpha/2) \\ u_{2(\cdot)}\sin(\alpha/2) \\ u_{3(\cdot)}\sin(\alpha/2) \\ \cos(\alpha/2) \end{bmatrix} = \begin{bmatrix} \eta_{1(\cdot)} \\ \eta_{2(\cdot)} \\ \eta_{3(\cdot)} \\ \eta_{4(\cdot)} \end{bmatrix},$$

where **u** is the unit axis of rotation and α is the angle rotated about **u**. The symbol $(\cdot)$ is used to denote the location where the name of the quaternion is specified, ('a' represents for aircraft quaternion which is the attitude of the aircraft relative to an inertial coordinate frame, 'd' stands for desired quaternion relative to an inertial coordinate frame, 'e' stands error quaternion which is the error between the aircraft and desired quaternions relative to the aircraft's body frame). A more thorough explanation of the different quaternions and coordinate frames used in this work is presented in the nomenclature section of this work. It is assumed that the tailsitter aircraft used with the method described in this paper has means of rotating about each of it's body frame principle axes independently.

3.1.1 Simulation Attitude Controller

In simulation, attitude control was performed directly on the error quaternion η_e which is the quaternion that describes the error between the aircraft quaternion η_a and the desired quaternion η_d. Proportional-integral-derivative (PID) control was used to determine desired actuator positions which were then sent to the servos to actuate the control surfaces.

To compute the error quaternion, we consider the definition of the desired quaternion as described in [21]. The desired quaternion is described mathematically as the quaternion composition of the error quaternion and the aircraft quaternion as represented by

$$\eta_d = \eta_e \otimes \eta_a = \{\eta_a\}_R \eta_e. \tag{3.1}$$

The term $\{\eta_{(.)}\}_R$ denotes the quaternion composed matrix

$$\{\eta_{(.)}\}_R = \begin{pmatrix} \eta_{4_{(.)}} & -\eta_{3_{(.)}} & \eta_{2_{(.)}} & \eta_{1_{(.)}} \\ \eta_{3_{(.)}} & \eta_{4_{(.)}} & -\eta_{1_{(.)}} & \eta_{2_{(.)}} \\ -\eta_{2_{(.)}} & \eta_{1_{(.)}} & \eta_{4_{(.)}} & \eta_{3_{(.)}} \\ -\eta_{1_{(.)}} & -\eta_{2_{(.)}} & -\eta_{3_{(.)}} & \eta_{4_{(.)}} \end{pmatrix}. \tag{3.2}$$

Using the property $\{\eta\}_R^T\{\eta\}_R = I_4$ (where I_4 is the four by four identity matrix) we can solve Equation 3.1 for η_e yielding

$$\eta_e = \{\eta_a\}_R^T \eta_d. \tag{3.3}$$

Because the error quaternion is a direct expression of the error in the orientation of the aircraft, its components can be used directly to compute desired control surface deflections. PD control was applied to the first three components of the error quaternion to compute the desired control surfaces deflections as

$$\delta_a = k_{\eta 1}\eta_{1e} - k_p p, \tag{3.4}$$

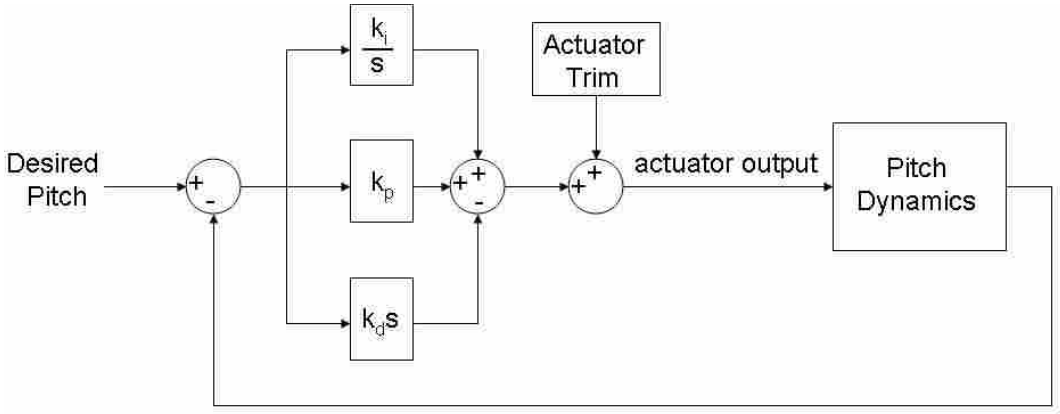
$$\delta_e = -(k_{\eta 2}\eta_{2e} - k_q q), \tag{3.5}$$

$$\delta_r = k_{\eta 3}\eta_{3e} - k_r r. \tag{3.6}$$

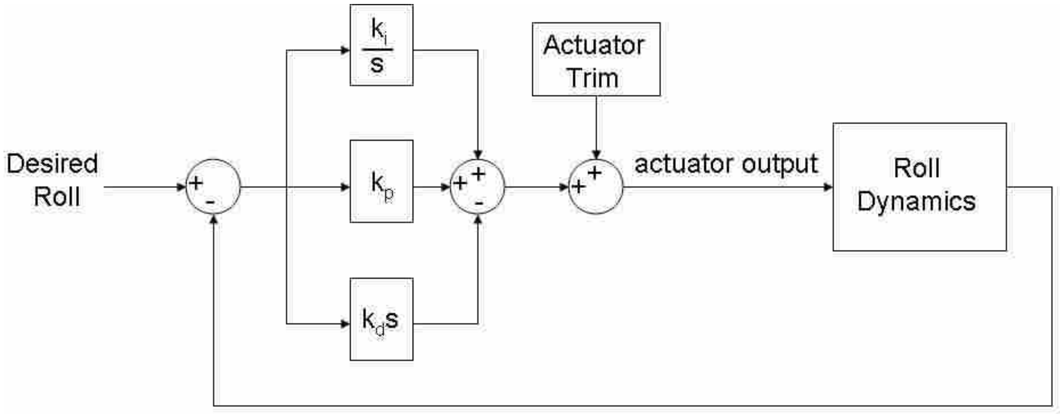
The terms $k_{\eta 1}$, $k_{\eta 2}$, and $k_{\eta 3}$ are the proportional gains while k_p, k_q, and k_r are the derivative gains.

3.1.2 Hardware Attitude Controller

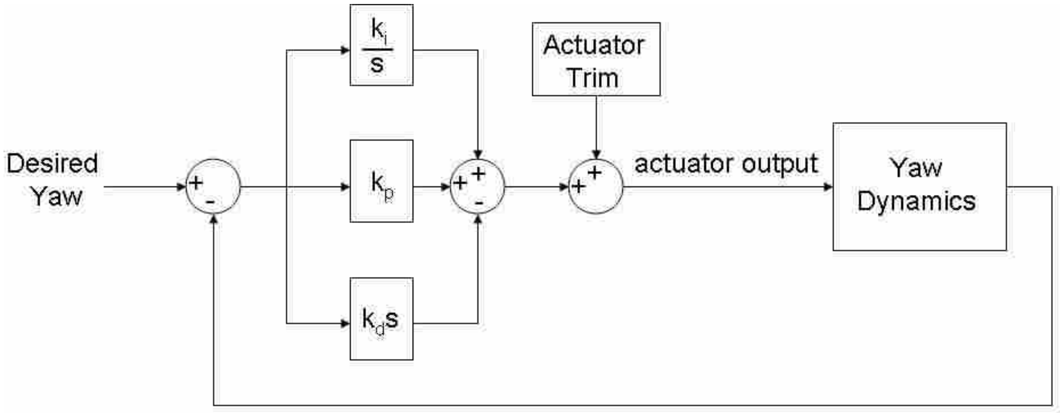
The attitude controller used in flight testing was very similar to the controller used in simulation flight tests. The main difference being that the controller used in hardware flight tests was done using Euler angles instead of directly in quaternions. This was done as the flight testing setup used at the Wright Patterson AFB μAVIARI facilities had been previously programmed using Euler angles. Also note that, the conversion between Euler angles and quaternions is known as outlined in [20].


Desired
Pitch
+
-
$\frac{k_i}{s}$
k_p
$k_d s$
+
+
-
Actuator
Trim
+
+
actuator output
Pitch
Dynamics

(a) Block diagram showing the PID pitch controller used in flight testing


Desired
Roll
+
-
$\frac{k_i}{s}$
k_p
$k_d s$
+
+
-
Actuator
Trim
+
+
actuator output
Roll
Dynamics

(b) Block diagram showing the PID roll controller used in flight testing

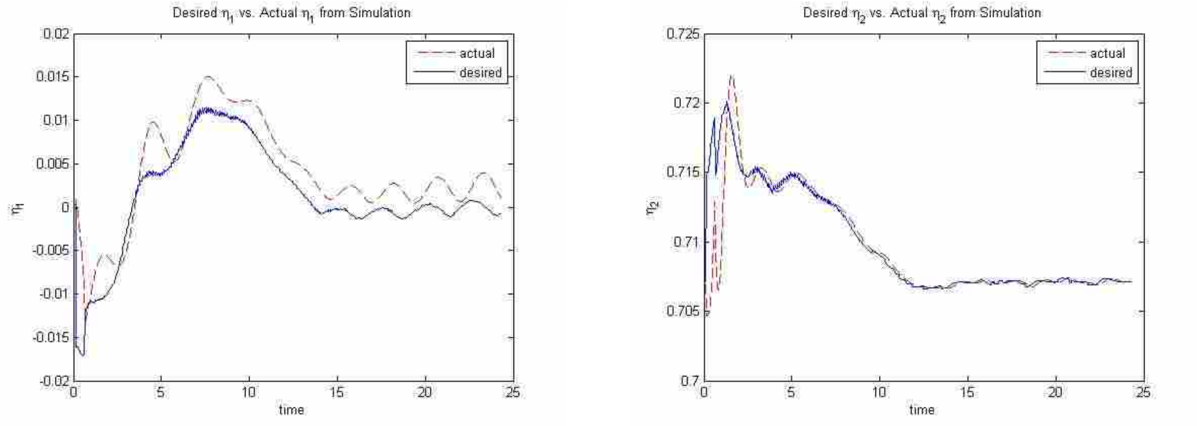

Desired
Yaw
+
-
$\frac{k_i}{s}$
k_p
$k_d s$
+
+
-
Actuator
Trim
+
+
actuator output
Yaw
Dynamics

(c) Block diagram showing the PID yaw controller used in flight testing

Figure 3.1: Block diagrams of the PID attitude controllers used in flight testing.

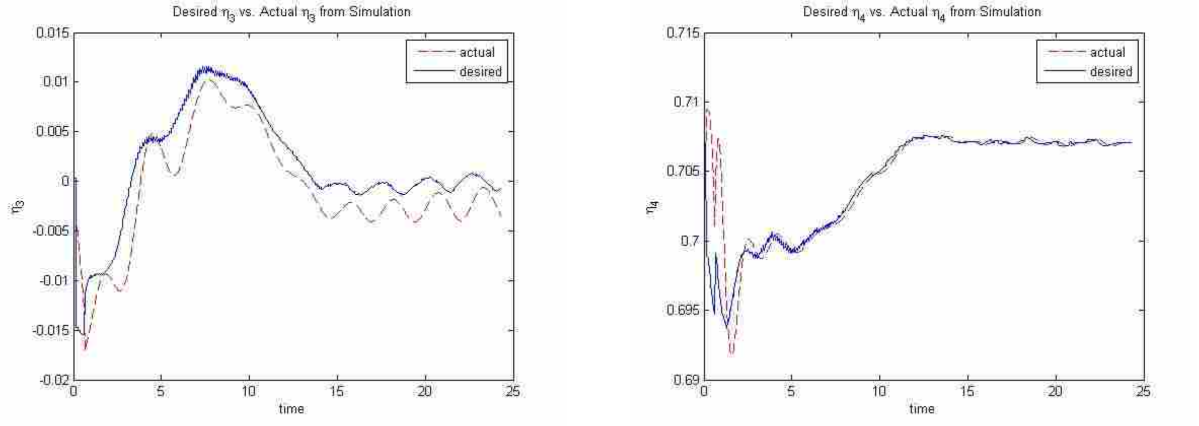
The controller used in hardware flight testing used PID control as well as a feedforward trim term as can be seen in Figure 3.1. This trim term was used to account for biases in the airframe or flying conditions and was set by the operator at the beginning of a flight to a value that minimized the error in the roll, pitch, and yaw terms. After the initial tuning these trim terms remained constant throughout the remainder of the flight.

3.1.3 Attitude Controller Results



(a) Desired η_1 and actual η_1 in simulation.

(b) Desired η_2 and actual η_2 in simulation.



(c) Desired η_3 and actual η_3 in simulation.

(d) Desired η_4 and actual η_4 in simulation.

Figure 3.2: Comparison of desired and actual attitude states from simulation flight tests.

Figure 3.2 shows the different components of the actual and desired attitude from simulated flight testing. These plots show the simulation controller's effectiveness in controlling the aircraft

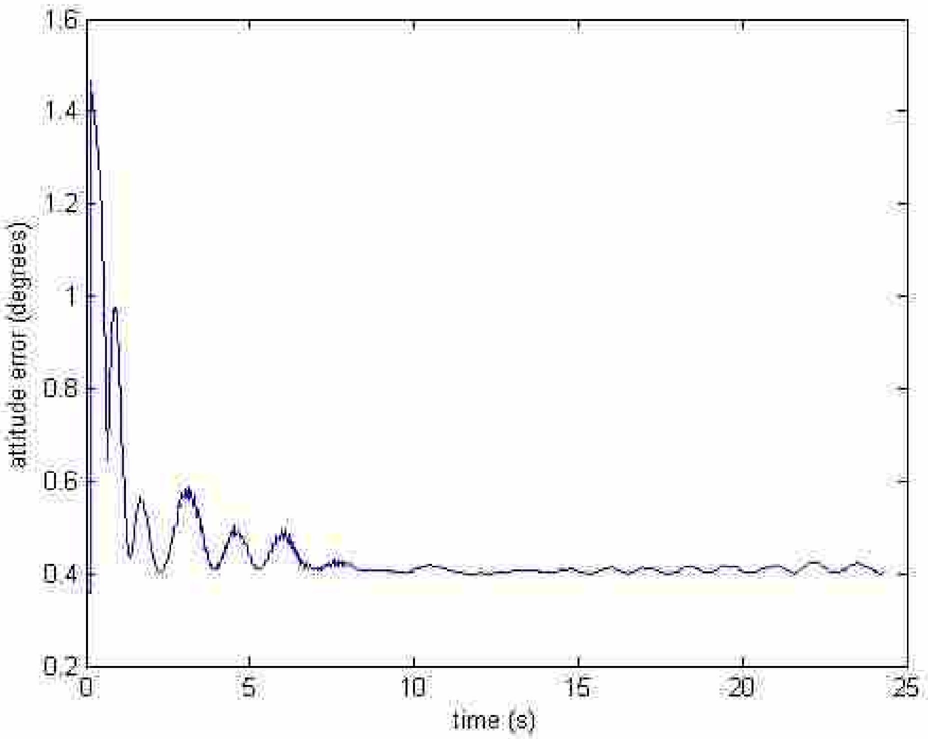
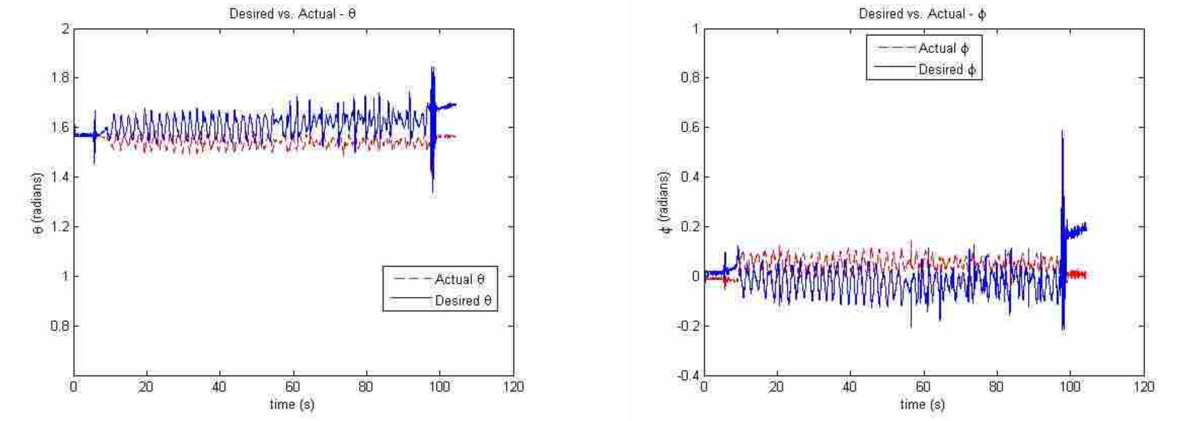


Figure 3.3: Error between the aircraft's attitude and the desired attitude in hardware flight testing.

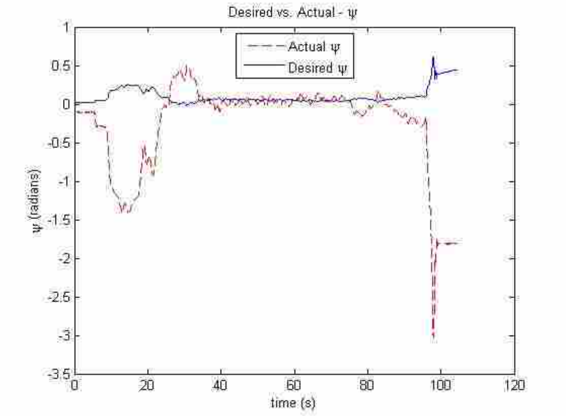
to a desired attitude. Figure 3.3 shows the error between the aircraft attitude and the desired attitude for the flight data shown in Figure 3.2. As can be seen in the plots, the attitude error stays under 1.5 degrees and settles out to approximately 0.4 degrees. The attitude error does not approach zero in simulation because no integral term was used in the simulation attitude controller.

Figure 3.4 shows a comparison of the actual and desired Euler angles commanded to the helicopter used in flight testing. The helicopter left the ground in hover flight at about 6 seconds and landed at approximately 97 seconds. The initial error in the ψ angle was trimmed out by the operator by about 35 seconds into the flight as can be seen in 3.4c. Note that while the results shown in Figure 3.4 were derived from flight testing using a helicopter, the pitch angle is shown as if from a tailsitter with $\theta = \pi/2$ representing the helicopter in hover flight with its rotors parallel to the ground. The roll and pitch attitudes exhibit noticeable oscillation. This is potentially due to the change in the aircraft's center of gravity that occurred when the onboard video equipment was added to the aircraft after the attitude controllers had been tuned. Figure 3.5 shows the error between the aircraft attitude and the desired attitude for the flight test shown in Figure 3.4. As can



(a) Desired pitch and actual pitch in flight testing

(b) Desired roll and actual roll in flight testing



(c) Desired yaw and actual yaw in flight testing

Figure 3.4: Block diagrams for the PID attitude controllers used in flight testing.

be seen in the plot, the attitude error is initially high, this is due to the initial error in the yaw angle that is trimmed out by about 35 seconds into the test. There is large error at the end the plot shown in Figure 3.5, this is because the aircraft lands at 94 seconds at an attitude that is not the desired attitude, specifically the aircraft lands with an error in yaw. Between the time when the aircraft is trimmed (35 s) and when the aircraft lands (94 s) the aircraft has an average attitude error of 5.5 degrees.

3.2 Altitude Controller

A responsive altitude controller with little overshoot is needed for vision-based landings. Because so much more energy is required to stay in hover flight than level flight, time spent waiting

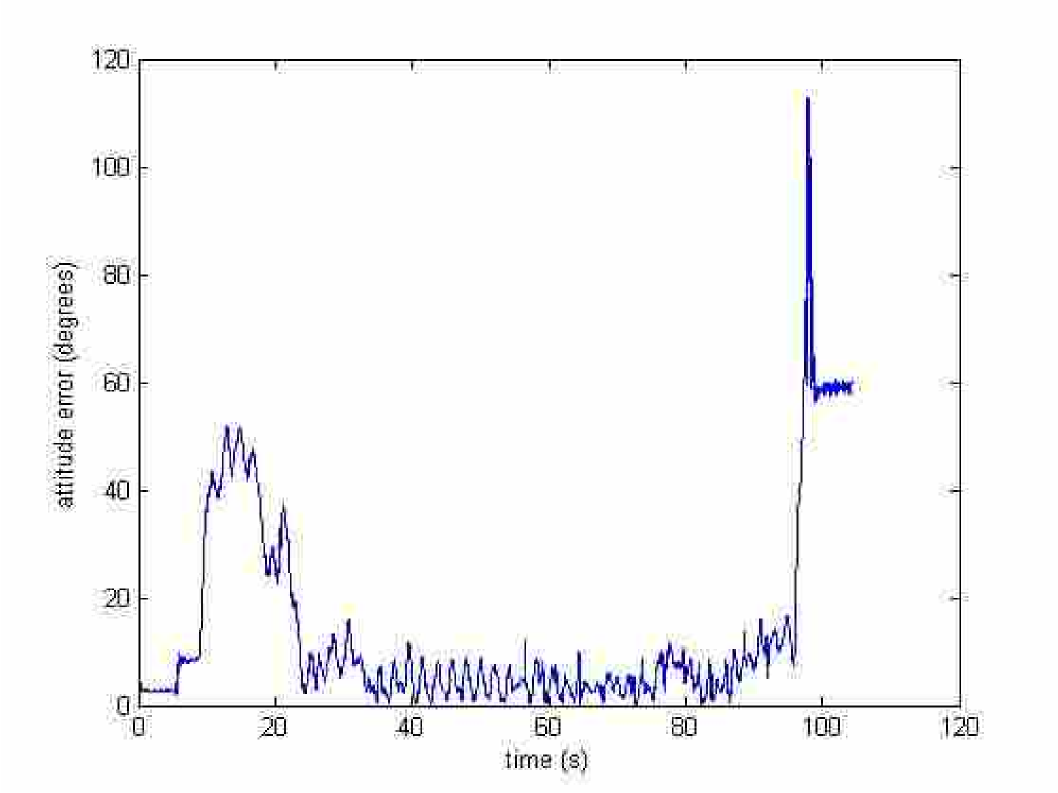


Figure 3.5: Error between the aircraft's attitude and the desired attitude in hardware flight testing.

for the altitude controller to reach steady state can greatly decrease flight duration and therefore must be minimized. Also, any overshoot in the controller when descending to a desired altitude could cause the UAV to land prematurely. For these reasons an altitude controller that responds quickly and has little or no overshoot is desirable for the task of precision landings. This section describes the controller used for altitude control as well as the results from using the altitude controller in simulation and hardware flight testing.

3.2.1 Simulation Altitude Controller

A block diagram of the altitude controller used in simulation can be seen in Figure 3.6. The desired altitude shown in the figure is the altitude of the user-defined waypoint or the altitude of the ground target when the vision-based navigation scheme is turned on. In simulation, the force produced by the motor-prop combination does not change with time for a given throttle setting. For this reason, no integral term is needed in the simulation altitude controller to maintain a constant

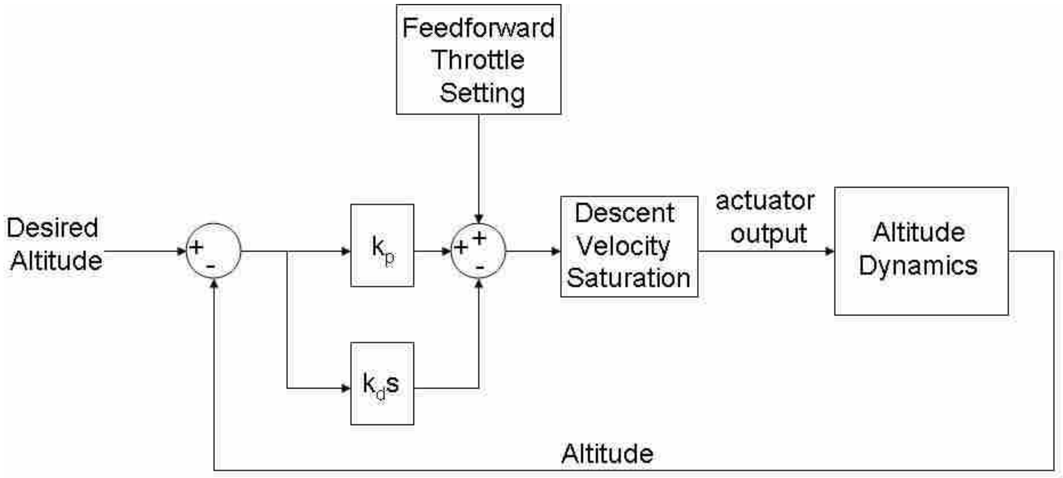


Figure 3.6: Block diagram of the altitude controller used in simulation.

altitude over a period of time. A feedforward throttle setting is applied to the control so that the feedback controller operates about the aircraft's hover flight throttle setting. This allows the controller to perform more smoothly, especially when descending to a desired altitude. To prevent the aircraft from descending too rapidly, which could cause overshoot and premature landing or crashing, we specified a maximum descent rate for the aircraft. When the vertical decent rate of the aircraft was greater than the user specified maximum descent rate, an throttle input of full throttle was applied until the vertical descent rate slowed to an acceptable value.

3.2.2 Hardware Altitude Controller

The controller used in hardware flight testing was very similar to the controller used to control altitude in simulation. As can be seen in Figure 3.7, the main difference is the addition of an integral term in the controller and the absence of a vertical descent rate saturation block. Unlike simulation, the throttle setting required to maintain a specific altitude for an aircraft changes with time in actual flight testing. This is because the thrust output of the helicopter is proportional to the decreasing battery voltage. For this reason, integral control was added to counteract the effects of thrust loss due to decreases in battery voltage.

Analogous to the feedforward throttle setting in the simulation controller, a throttle trim setting was used in the hardware altitude controller. This throttle trim setting was determined by increasing the throttle setting until the aircraft maintained a constant desired altitude. The throttle

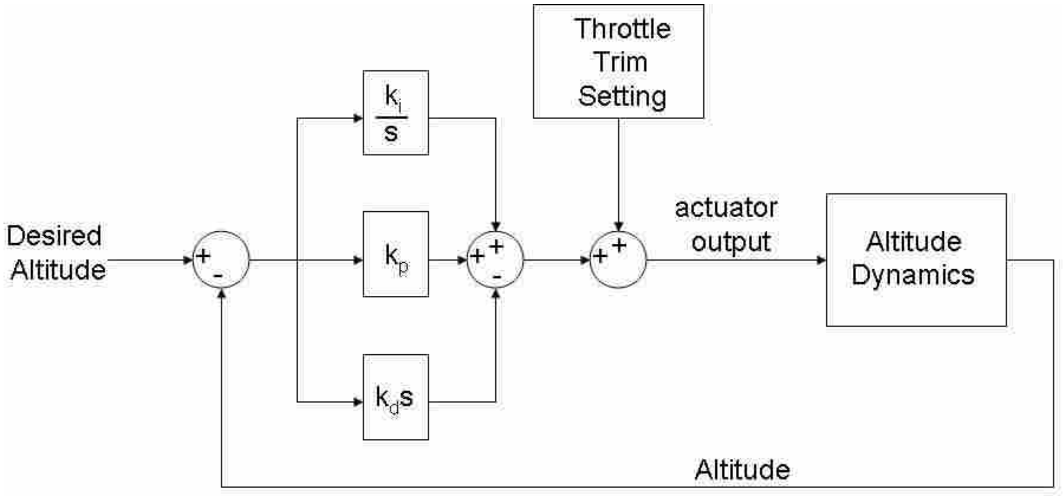


Figure 3.7: Block diagram of the altitude controller used in flight testing.

trim setting was then left constant for the remainder of the flight. The throttle trim setting was reset at the beginning of each flight.

For safety reasons the desired altitude of the aircraft was set by the operator throughout each flight test. This was to prevent damage to the aircraft that could have occurred through premature landings or unexpected behavior in the calculation of a computer derived desired altitude.

3.2.3 Altitude Controller Results

The response of the altitude controller used in simulation and described in Section 3.2.1 is shown in Figure 3.8. The aircraft started at ground level and was initially commanded to 5 meters AGL, upon reaching 5 meters AGL the aircraft was commanded to 10 meters AGL. After reaching 10 meters AGL the aircraft was commanded to descend back to ground level. Notice that the slope of the aircraft's altitude during the descent phase (40-140 seconds) is roughly constant. This is due to the descent velocity saturation method used to prevent the aircraft from becoming uncontrollable by descending too quickly.

The response of the altitude controller used in flight testing described in Section 3.2.2 is shown in Figure 3.9. The helicopter started at ground level with a desired altitude below ground while the rotors were slowly spun up to some initial angular velocity. Once the helicopter rotors were spinning the desired altitude was set to 2 meters, this was later decreased to 1.5, 1.0, 0.5 meters, and a negative altitude corresponding to an desired altitude of zero. Because the desired

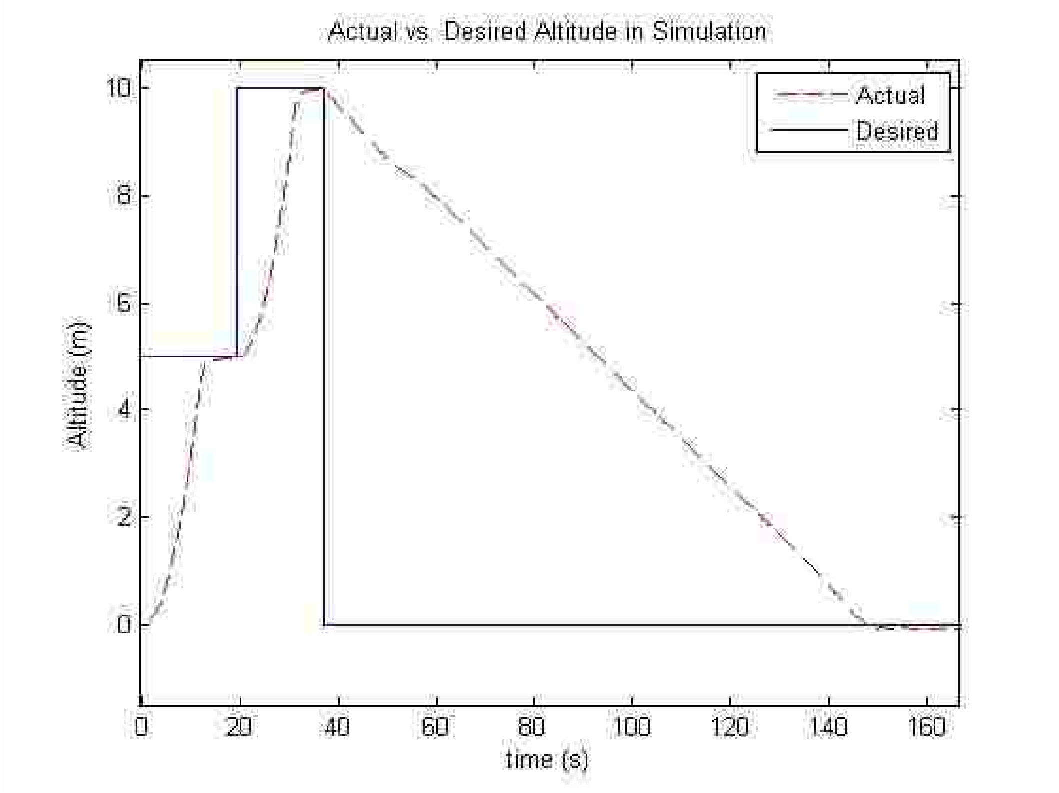


Figure 3.8: Comparison of actual and desired altitudes in simulation using the controller described in Section 3.2.1.

altitude of the aircraft was controlled by the operator, as opposed to an algorithm, no descent velocity saturation techniques were needed. As shown in the figure, the actual altitude of the helicopter closely follows the desired altitude for the aircraft.

3.3 Navigation Controller

The navigation controller uses the current states of the UAV and the desired location of the UAV to determine a desired orientation of the aircraft that will navigate the aircraft to the desired location. The output of the navigation controller is used as the input to the attitude controllers previously described in this chapter. In our tests we were not concerned with the heading of the aircraft being aligned with the motion vector of the aircraft in the North-East plane. For this reason the desired heading of the aircraft was set at a constant of zero degrees off North for both simulation and flight testing.

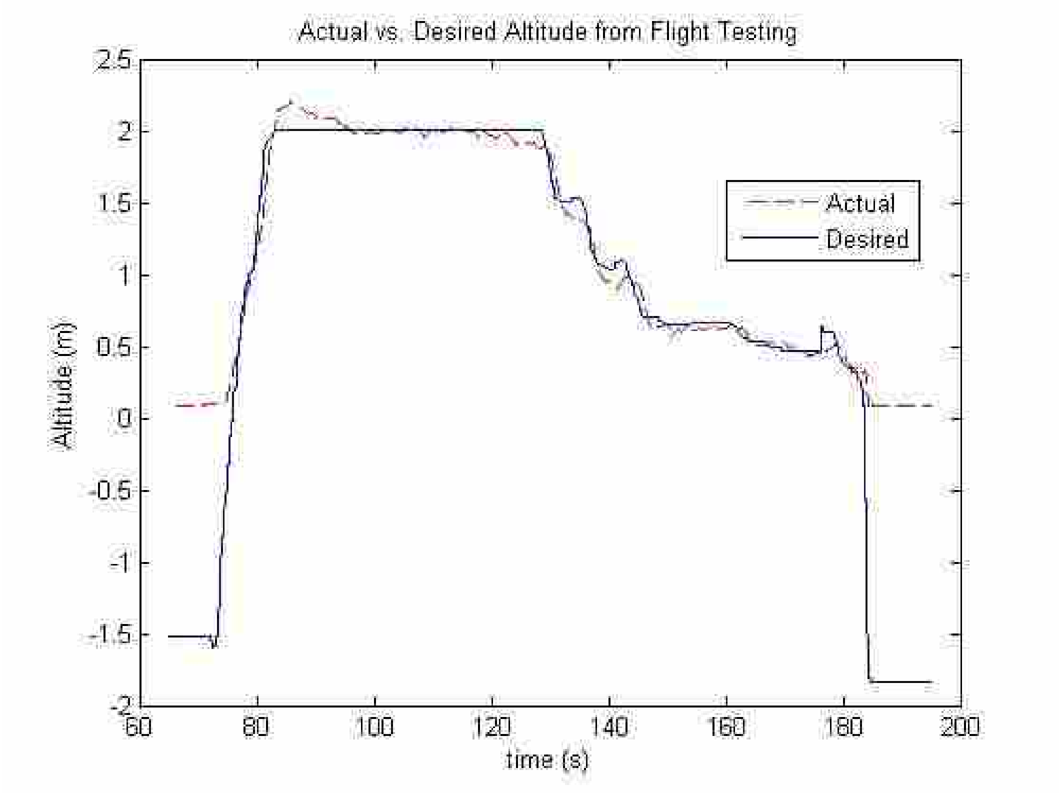


Figure 3.9: Comparison of actual and desired altitudes from flight testing using the controller described in Section 3.2.2.

The navigation controller used in simulation is detailed by Knoebel et al. [5] and is presented in Appendix A. The navigational controller used in hardware is similar to that used in simulation and is presented below in this section. The results of each controller's effectiveness at navigating the helicopter to a desired location is also presented.

3.3.1 Hardware Navigation Controller

Figures 3.10 and 3.11 show the navigation controller used to navigate the aircraft from its current location to a desired location. The hardware navigation controller differs from the navigation controller used in simulation in that it is performed using Euler angles instead of quaternions for reasons explained in section 3.1.2. The blocks labeled 'Pitch Controller' and 'Roll Controller' represent the block diagrams shown in Figures 3.1a and 3.1b respectively.

In Figures 3.10 and 3.11, the *x* and *y* directions are the *x* and *y* directions in the helicopter body frame with the *x*-axis pointing out the nose of the helicopter, the *y*-axis pointing out the right

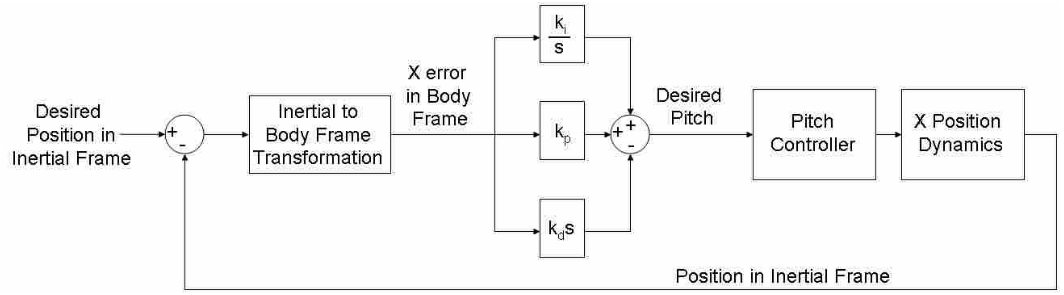

Desired Position in Inertial Frame
+
-
Inertial to Body Frame Transformation
X error in Body Frame
$\frac{k_i}{s}$
k_p
$k_d s$
+
+
-
Desired Pitch
Pitch Controller
X Position Dynamics
Position in Inertial Frame

Figure 3.10: Block diagram of the navigation x direction controller used in flight testing.

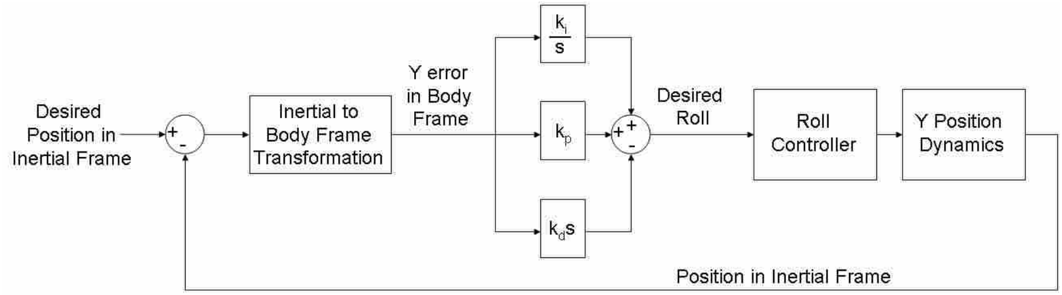

Desired Position in Inertial Frame
+
-
Inertial to Body Frame Transformation
Y error in Body Frame
$\frac{k_i}{s}$
k_p
$k_d s$
+
+
-
Desired Roll
Roll Controller
Y Position Dynamics
Position in Inertial Frame

Figure 3.11: Block diagram of the navigation y direction controller used in flight testing.

side of the helicopter, and the z-axis pointing out the bottom of the helicopter. These x and y directions in the helicopter body frame correspond to the z and y directions in the tailsitter body frame, respectively, for a tailsitter in hover flight.

Operator specified waypoints were used to specify the desired position of the UAV if the ground target was not present in the image frame. When the ground target was detected in the image frame the estimated location of the UAV from vision was used to navigate the UAV onto the centroid of the target. The location of this desired position in the inertial frame was then determined in the aircraft's body frame as

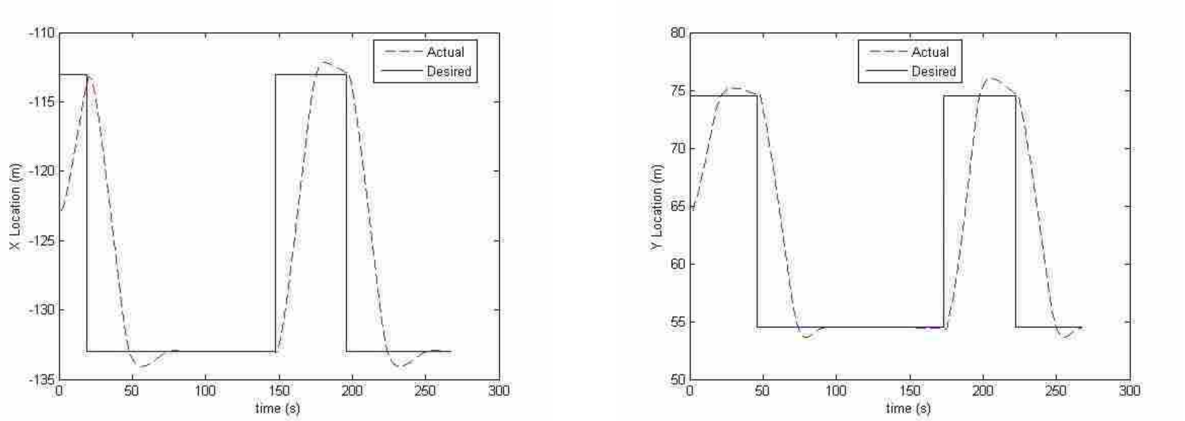
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{body} = R_b^v \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{error} , \tag{3.7}$$

where $[x\ y\ z]^T_{body}$ is the aircraft position error in the helicopter body reference frame, $[x\ y\ z]^T_{error}$ is the error between the desired location and the actual aircraft location in the inertial reference frame, and R_v^b is the rotation matrix between the vehicle coordinate system and the body coordinate system. Using the 3-2-1 aerospace rotation sequence [20] the rotation from the vehicle frame to the body frame is given by

$$R_b^v = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\theta\sin\phi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\theta\cos\phi \end{bmatrix}. \quad (3.8)$$

An alternate definition of the transpose of the matrix in Equation 3.8 is given in Chapter 4 as Equation 4.1 which uses quaternions to define the matrix. There is no difference between the matrix defined in Equation 3.8 and the transpose of the matrix defined in Equation 4.1 besides the fact that the former is defined using Euler angles and the latter is defined using quaternions. The two matrices yield the same direction cosine matrix.

3.3.2 Navigation Controller Results

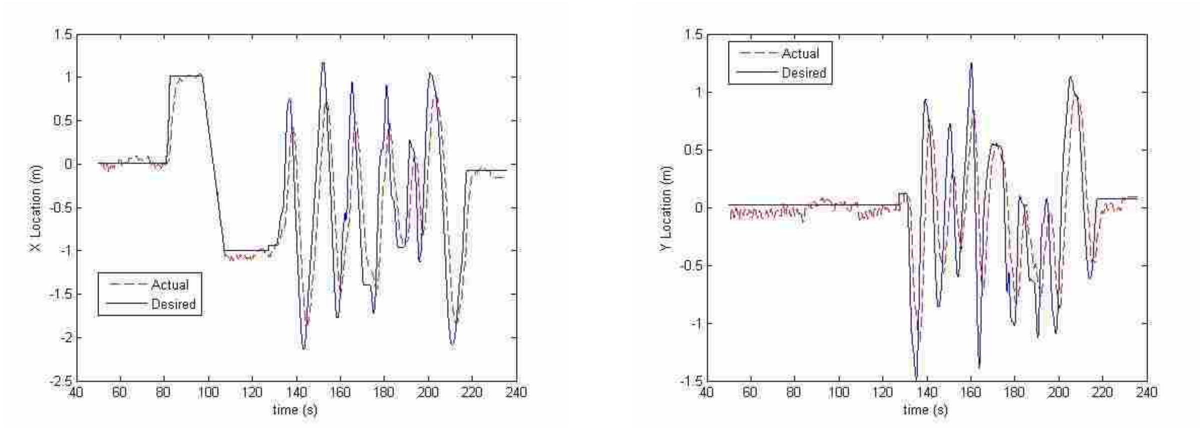


(a) Comparison of the desired and actual x location of the aircraft in simulation.

(b) Comparison of the desired and actual y location of the aircraft in simulation.

Figure 3.12: Desired and actual location of the aircraft in simulation.

To show the effectiveness of the navigational controller used in simulation, the simulated aircraft was commanded to fly waypoints that outlined a 20 meter wide square in the North-East plane. The results of following these waypoints are presented in Figure 3.12. The navigational controller exhibits a rise time of 16.2 seconds to move 20 meters to the next desired waypoint. The controller also exhibits 8 percent of overshoot after having reached the desired location. As Figure 3.12 demonstrates, the navigational controller described in this chapter is capable of tracking a desired input in simulation. This task of tracking a desired position is essential to vision-based landings of VTOL aircraft.



(a) Comparison of the desired and actual x location of the aircraft from flight testing.

(b) Comparison of the desired and actual y location of the aircraft from flight testing.

Figure 3.13: Desired and actual location of the aircraft from flight testing.

Figure 3.13 shows the navigational controller's effectiveness at tracking a reference input. Figure 3.13 shows the data from a flight test where the aircraft was initially commanded to fly in a straight line forwards and backwards corresponding to the time from 40 to 120 seconds, after which the aircraft was commanded to fly in a circular pattern corresponding to the time from 120 to 215 seconds. The controller exhibits a rise time of 3.1 seconds for a step input of 1 meter with no overshoot. There was an average position error of 8 centimeters after the aircraft had reached a constant desired input for the flight shown in Figure 3.13. The navigational controller results show that accuracy in our flight testing setup has a lower bound of approximately 8 centimeters. This is still well within the sub-meter goal that we have for this vision-based method.

CHAPTER 4. RELATIVE POSITION ESTIMATION

For a VTOL UAV to land on a visually detected target, some information regarding its position relative to the landing sight must be known. Using the pixel location of the target in the image and the aircraft's orientation relative to some inertial frame, a vector pointing from the UAV to the target centroid can be calculated by rotating between the different reference frames that are described in the next section. Using the altitude AGL of the aircraft, the vector pointing towards the target centroid can be scaled such that the aircraft's position relative to the target is known in the North and East directions. This chapter introduces the coordinate frames used to perform these calculations which is followed by a brief description of the pinhole camera model. We also introduce the method used to detect the target in the image frame and the computation of the image centroid used for navigation. Finally, the mathematics behind computing the relative position of the UAV is presented followed by our simulation and hardware results.

4.1 Coordinate Frames

Several different coordinate frames are used to determine the relative position of the UAV to a ground target. The *inertial frame* is located at a user-defined home location with its x-axis pointing North, its y-axis pointing East, and its z-axis pointing into the ground, forming a right-handed coordinate system. The *target frame* axes point in the same directions as the inertial frame but with its origin located at the center of the ground target of interest. The *vehicle frame* is also aligned with the inertial frame with its origin at the UAV's center of gravity. The inertial frame, the target frame, and the vehicle frame are all shown in Figure 4.1. The *body frame* shares its origin with the vehicle frame with the x-axis pointing out the nose of the aircraft, the y-axis pointing out the right wing, and the z-axis pointing out the belly of the aircraft, also forming a right handed coordinate system. The body frame of a tailsitter aircraft is shown in Figure 4.2a

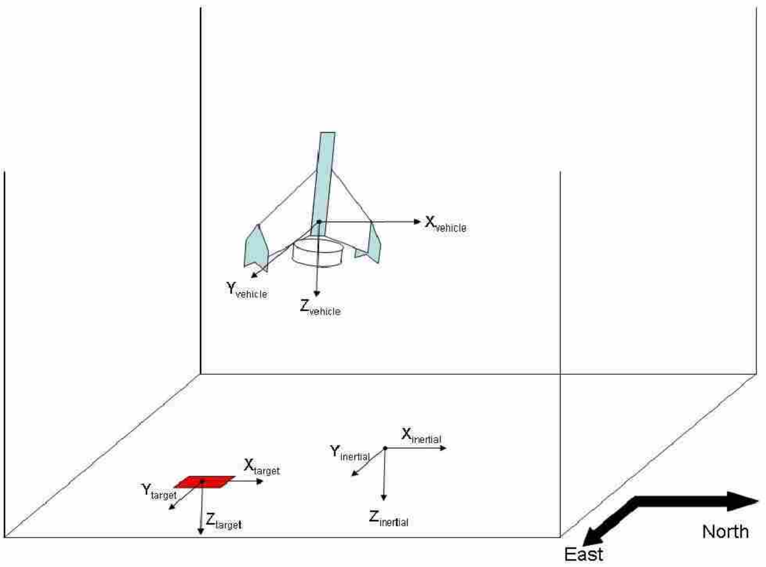
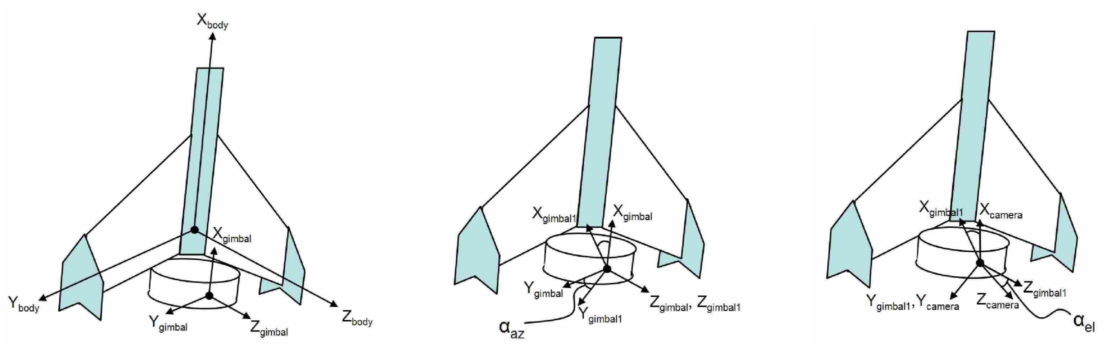

$X_{vehicle}$
$Y_{vehicle}$
$Z_{vehicle}$
$X_{inertial}$
$Y_{inertial}$
$Z_{inertial}$
X_{target}
Y_{target}
Z_{target}
North
East

Figure 4.1: Inertial based reference frames.


X_{body}
X_{gimbal}
Y_{body}
Y_{gimbal}
Z_{gimbal}
Z_{body}
$X_{gimbal1}$
X_{gimbal}
Y_{gimbal}
$Y_{gimbal1}$
$Z_{gimbal}, Z_{gimbal1}$
α_{az}
$X_{gimbal1}$
X_{camera}
$Z_{gimbal1}$
$Y_{gimbal1}, Y_{camera}$
Z_{camera}
α_{el}

(a) Body based reference frames.

(b) Gimbal1 frame and azimuth angle.

(c) Gimbal2 frame and elevation angle

Figure 4.2: Several different reference frames used in the relative position estimation are described.

Despite the fact that no gimbal is used to point the fixed onboard camera, we still denote several gimbal-based frames for ease in defining the orientation of the camera and to be consistent with the work of predecessors [22]. The *gimbal frame* has an origin located at the center of the camera with axes aligned with the body frame axes. The *gimbal-1 frame* is the gimbal frame rotated by the azimuth angle, (α_{az}), about the gimbal frame z-axis. The *gimbal-2 frame* is the gimbal-1 frame rotated by the angle α_{el} about the negative gimbal-1 frame y-axis. The *camera frame* has an origin located at the center of the camera sensor with the x-axis pointing out the right of the camera from the perspective of looking through the camera, the y-axis pointing out

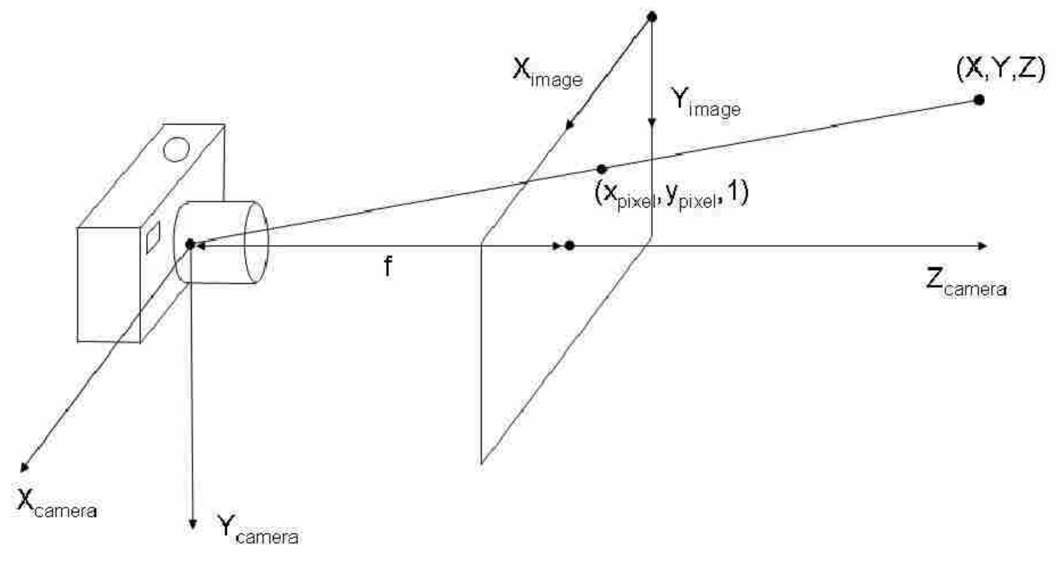


Figure 4.3: Pinhole camera model.

the bottom of the camera, and the z-axis pointing out the lens of the camera. Each of the gimbal coordinate frames and the camera frame are shown in Figure 4.2. The rotation and transformation matrices described in the nomenclature section of this work are used to rotate between these frames as described in later in this section and in Section 4.4.

4.2 Pinhole Camera Model

The pinhole camera model describes the mathematical relationship between a point in three-dimensional space and its projection onto the two-dimensional image plane of an ideal pinhole camera. As can be seen in Figure 4.3, a three-dimensional point is projected onto an imaginary plane at a distance of one focal length from the camera. Points on this image plane are described using the two-dimensional image reference frame whose origin we have defined at the top left-hand side of the image and whose x-axis points out the right of the image and whose y-axis points out the bottom of the image as shown in Figure 4.3. The image reference frame uses units of pixels to describe the location of a point on the image plane. The transformation between the image frame and the camera frame is made possible with the camera calibration matrix K, which describes some of the intrinsic parameters of the camera. The parameters f_x, f_y, c_x, and c_y are obtained from a camera calibration as is performed by the Matlab camera calibration toolbox or using the OpenCV camera calibration example. The terms c_x and c_y are the x and y translations needed to move the image frame origin at the upper right of the image, to the camera frame origin at the center of the image. The parameters f_x and f_y are the conversions from pixels to meters in the x and y directions applied at one focal length from the camera in the camera frame z direction If the camera pixels are perfectly square, then f_x and f_y are the same value. The conversion from the image coordinate frame to the camera coordinate frame is given by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{camera} = K \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{image},$$

where,

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}.$$

This does not however give the actual location of the point of interest in three-dimensional space but rather the location of that point projected onto the image plane at a distance of one focal length from the camera. Because the location of the point of interest has been reduced from 3-dimensions to 2-dimensions when the scene is captured by the camera, the location of the point of interest is only determined up to an unknown scale. In essence we only know a vector pointing from the camera to the object but we don't know the length of that vector. The method of determining the actual three-dimensional location of a point in space is shown in Section 4.4 of this chapter and requires the use of information such as the distance to the target to resolve this unknown scale.

4.3 Target Detection

Vision-based landings are dependent on the ability to autonomously detect and track a target in video streamed from a UAV. Many different methods exist for automatic target recognition, the more complex of which are beyond the scope of this work. Automatic target recognition remains an open research topic with several working solutions that fit different needs. In this work, we utilize a target detection algorithm that can determine the pixel coordinates of the centroid of a target in the image frame.

The method used in this work to determine the centroid of a ground target of interest was simple color segmentation for the color red in simulation and simple thresholding of a black and white image in hardware testing. These methods are demonstrate the vision-guided landing concepts in this work.

4.3.1 Color Segmentation

To detect the target using color segmentation, the image sent from the onboard camera was first converted to the hue, saturation, value (HSV) color space. The image space was then searched for colors that fit between a window of hue and saturation values. This is represented mathematically as

$$P_{i,j} = \begin{cases} 1, \text{ if } H_{\text{min}} < H_{i,j} < H_{\text{max}} \ \& \ S_{\text{min}} < S_{i,j} < S_{\text{max}} \\ 0, \text{ otherwise} \end{cases} ,$$

where $P_{i,j}$ is the value of the binary segmented image at pixel location (i,j), $H_{i,j}$ is the hue value of the original image at location (i,j), and $S_{i,j}$ is the saturation value of the original image at location (i,j). H_{max} and H_{min} define the window of acceptable hue values that classify the color of interest while the terms S_{max} and S_{min} define the window of acceptable saturation values for the color of interest. The advantage of using the HSV color space over the red-green-blue (RGB) or other standard color spaces is that color segmentation in the HSV color space is more robust to changes in lighting conditions. An example of color segmentation for the color red can be seen in the Aviones simulation images shown in Figure 2.1. The center image in Figure 2.1 shows the view from the onboard camera while the image to the right shows the image received from the onboard camera after having been segmented for the color red.

4.3.2 Connected Components

Once the color segmentation or thresholding has been performed yielding a binary image, we group pixels that meet the segmentation/thresholding conditions into connected components.

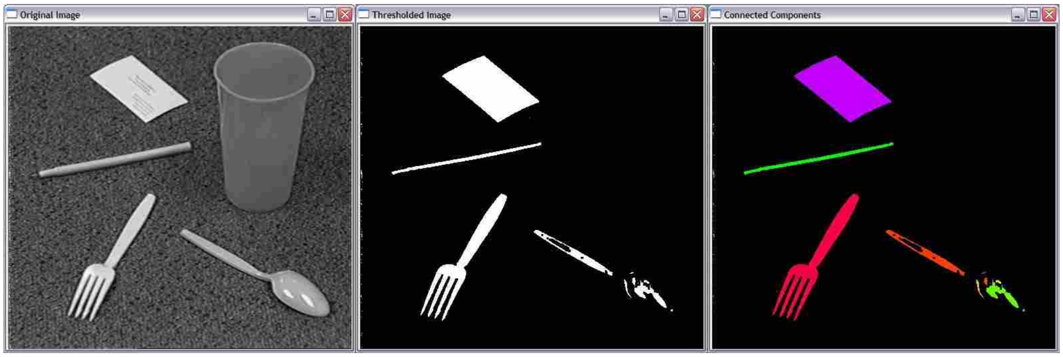


Figure 4.4: Connected components of a thresholded image.

Each connected region of pixels that fit into the window of acceptable colors or light intensity is labeled as a unique connected component. The 8-neighborhood connected component algorithm used in this research was done using the CvContours function of OpenCV, the documentation for CvContours is provided in [19]. The basic methodology of the 8-neighborhood connected component algorithm is described in [23] as Algorithm 8.1. Figure 4.4 shows a visual representation of what the connected components algorithm does. The left most image in the figure shows a black and white image analogous to an image coming from a camera onboard a UAV. The middle image is the original image thresholded to only show lighter colored objects. The right image is colored such that each connected component is a unique color. Note that the spoon in the images in Figure 4.4 did not threshold as a single entity and therefore was composed of several different connected components.

If the largest connected component in the image received from the UAV is larger than a user-defined threshold, then the target is assumed found and the center of mass of the largest connected component is used for navigation. This method of using segmentation or thresholding to find the ground target of interest assumes that no other objects in the image have the same properties as the target. In other words, we are searching for a red target at a location void of any red objects except for the target.

4.3.3 Centroid Computation

The x and y centroids of the largest connected component is calculated as

$$x_{CoM} = \frac{\sum\limits_{i=1}^{k} x_i}{A_{cc}},$$

$$y_{CoM} = \frac{\sum\limits_{i=1}^{k} y_i}{A_{cc}},$$

where k is the number of pixels that makes up the largest connected component and x_i and y_i are the x and y pixel coordinates of the i^{th} components of the largest connected component in the image frame. The term A_{cc} is the area of the largest connected component in units of pixels; this is also the number of pixels that composes the largest connected component.

4.4 Coordinate Transformations

With the pixel coordinates of the centroid of the target and the UAV's altitude AGL and orientation information, the relative position of the UAV to the target in the target frame is calculated as

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = R_b^v R_g^b \left(\left(R_{g1}^g R_{g2}^{g1} R_c^{g2} K^{-1} \begin{bmatrix} X_{\text{pixel}} \\ Y_{\text{pixel}} \\ 1 \end{bmatrix} \right) + \begin{bmatrix} x_{\text{gtrans}} \\ y_{\text{gtrans}} \\ z_{\text{gtrans}} \end{bmatrix} \right),$$

$$\gamma = \frac{AGL}{p_3}, \quad \text{and}$$

$$\begin{bmatrix} X_{\text{UAV}} \\ Y_{\text{UAV}} \\ AGL \end{bmatrix} = \gamma \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}.$$

In these equations, $[v_1\ v_2\ v_3]^T$ is a vector that points to the target centroid from the camera frame origin and γ is a scaling factor that represents the distance from the camera frame origin to the target centroid. The rotation and transformation matrices used above are defined as

$$R_v^b = \begin{pmatrix} 2\eta_{1\mathrm{a}}^2 - 1 + 2\eta_{4\mathrm{a}}^2 & 2\eta_{1\mathrm{a}}\eta_{2\mathrm{a}} + 2\eta_{3\mathrm{a}}\eta_{4\mathrm{a}} & 2\eta_{1\mathrm{a}}\eta_{3\mathrm{a}} - 2\eta_{2\mathrm{a}}\eta_{4\mathrm{a}} \\ 2\eta_{1\mathrm{a}}\eta_{2\mathrm{a}} - 2\eta_{3\mathrm{a}}\eta_{4\mathrm{a}} & 2\eta_{2\mathrm{a}}^2 - 1 + 2\eta_{4\mathrm{a}}^2 & 2\eta_{2\mathrm{a}}\eta_{3\mathrm{a}} + 2\eta_{1\mathrm{a}}\eta_{4\mathrm{a}} \\ 2\eta_{1\mathrm{a}}\eta_{3\mathrm{a}} + 2\eta_{2\mathrm{a}}\eta_{4\mathrm{a}} & 2\eta_{2\mathrm{a}}\eta_{3\mathrm{a}} - 2\eta_{1\mathrm{a}}\eta_{4\mathrm{a}} & 2\eta_{3\mathrm{a}}^2 - 1 + 2\eta_{4\mathrm{a}}^2 \end{pmatrix}, \tag{4.1}$$

$$R_b^g = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ R_g^{g1} = \begin{pmatrix} \cos\alpha_{\mathrm{az}} & \sin\alpha_{\mathrm{az}} & 0 \\ -\sin\alpha_{\mathrm{az}} & \cos\alpha_{\mathrm{az}} & 0 \\ 0 & 0 & 1 \end{pmatrix},\ R_{g1}^{g2} = \begin{pmatrix} \cos\alpha_{\mathrm{el}} & 0 & \sin\alpha_{\mathrm{el}} \\ 0 & 1 & 0 \\ -\sin\alpha_{\mathrm{el}} & 0 & \cos\alpha_{\mathrm{el}} \end{pmatrix},$$

$$R_{g2}^c = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

Because the above rotation matrices are orthonormal (with the exception of K), the following property holds:

$$(R_{\mathrm{f}_1}^{\mathrm{f}_2})^T = R_{\mathrm{f}_2}^{\mathrm{f}_1},$$

where f_1 and f_2 simply serve as placeholders for the previously defined reference frames.

4.5 Results

This position estimation method was tested in the simulation environment and using the hardware setup described in Chapter 2. This section presents the findings of this method from simulation and hardware testing.

In both simulation and hardware flight testing the VTOL UAV started at ground level and used truth position and orientation information to ascend in altitude until the ground target came

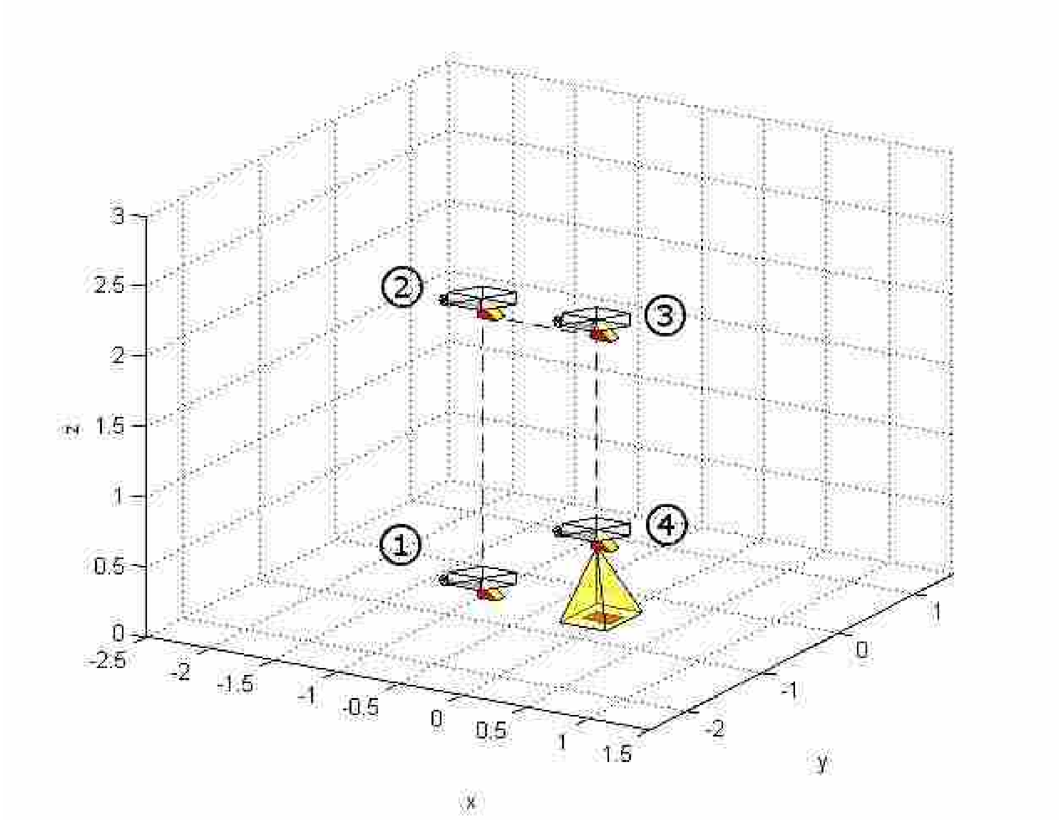
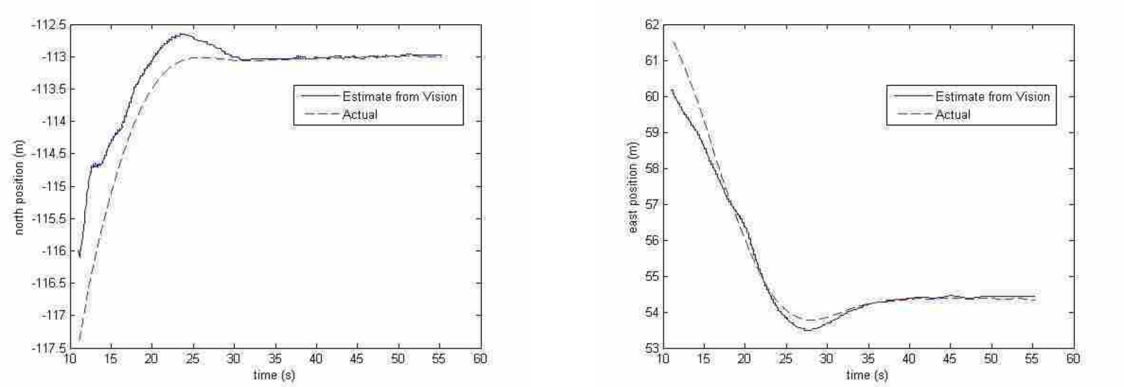


Figure 4.5: Flight path used in simulation and hardware flight testing.

into the onboard camera's view. Once the ground target came into view the vision-based relative position estimation method described in this chapter was used to navigate the UAV over the ground target. Figure 4.5 shows the different phases of the test flights used to validate this research. The UAV uses truth data to navigate from points 1 to 2 shown in the figure. Point 2 is at an altitude such that the ground target is visible to the onboard camera. From point 2 to point 4 vision is used to estimate the UAV position relative to the target.

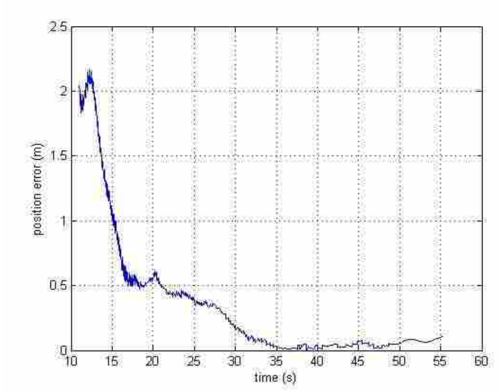
4.5.1 Position Estimation Simulation Results

Figure 4.6 shows the results of the vision-based position estimation algorithm compared to the actual position of the UAV in simulation. Since the position estimation algorithm proposed in this chapter gives the position of the UAV relative to the centroid of the ground target, the location of the centroid of the ground target in the inertial frame was used in comparing the data represented in Figure 4.6.



(a) Comparison of the north position estimate from vision of the UAV and the actual north position in simulation.

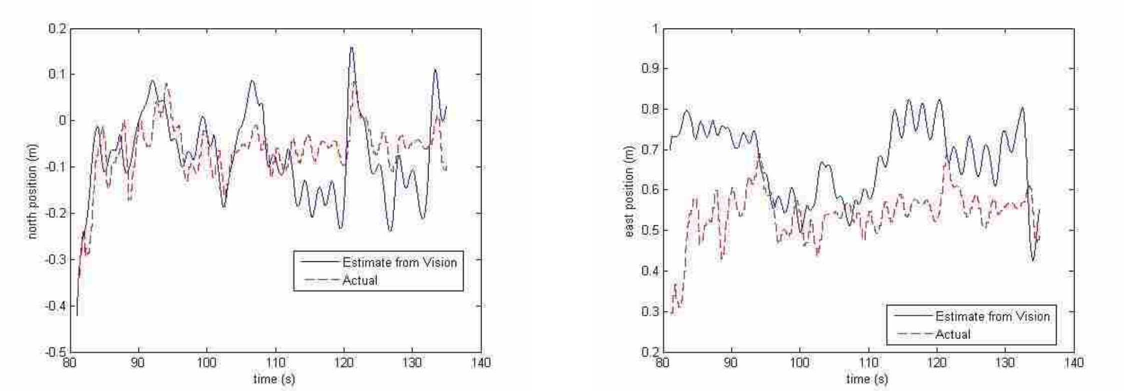
(b) Comparison of the east position estimate from vision of the UAV and the actual east position in simulation.



(c) Estimation error of the vision-based position estimation method in simulation.

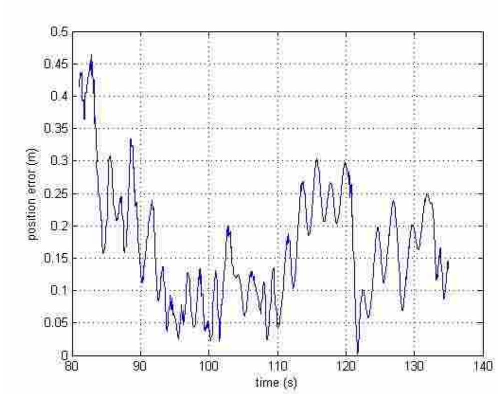
Figure 4.6: Simulation results of the vision-based relative position estimation algorithm.

As can be seen from Figure 4.6c, the estimation error between the estimated position of the UAV and the actual position of the UAV starts out at approximately 2 meters for the given run and then quickly approaches zero. This initial error is attributed to the centroid being located at the edge of the viewable region. As the centroid moves closer to the center of the image the position estimation becomes more accurate. Once the UAV positions itself above the target the estimation error drops to under 8 cm from an altitude of 10 meters.



(a) Comparison of the north position estimate from vision of the UAV and the actual north position.

(b) Comparison of the east position estimate from vision of the UAV and the actual east position.



(c) Estimation error of the vision-based position estimation method

Figure 4.7: Hardware results of the vision-based relative position estimation algorithm.

4.5.2 Position Estimation Hardware Results

Figure 4.7 shows the position estimation results from hardware flight testing using the helicopter platform described in Section 2.2. For the flight test, the helicopter started behind the target at an altitude of 2 meters with the target in the UAV's camera view. The location of the ground target in the image frame and information about the aircraft's attitude and altitude were then used to calculate a relative position between the helicopter and the target. This relative position information was then supplied to the navigational controller described in Section 3.3 and used to position the helicopter over the target. The helicopter was then commanded to descend to altitudes of 1.5, 1.0, and 0.5 meters while still keeping the target in view using the method described in Chapter 5. The position of the UAV estimated from the VICON motion capture system is used as the truth data in the plots shown in Figure 4.7 and has an accuracy of approximately two millimeters.

Table 4.1 shows the relative position estimation results from the ten different flight tests that were performed to test the methods described in this work. The time column in the table is the cumulative time that the target was present in the image frame in order for a position estimate to be made. Position estimates were made for a cumulative flight time of over 26 minutes with an average position error of 22 centimeters. The table also presents the maximum error calculated for each test flight with a maximum error over all test flights being 1.17 meters. The variability in these maximum errors can be attributed to two leading factors, primarily, measurement noise in the VICON system and false target identifications. The VICON MX motion capture system used for attitude and altitude measurements was typically calibrated once in the day before a series of flight tests. An increase in the measurement noise was noticed as the time since a calibration increased. This was attributed to vehicles passing nearby or inside the μAVIARI facilities and due to the pressure change of repeatedly opening and closing the flight facility doors causing a ballooning effect inside the small, sheet-metal lined room.

Because the VICON cameras were mounted to the structure of the flight facility walls, any vibration applied to the structure would potentially change the pose of the cameras. Each camera's pose had been calculated during the calibration process and assumed constant thereafter so any change in the camera's pose after the calibration was complete would add additional error to the aircraft position and attitude estimates. By observing the attitude estimate of the aircraft calculated from the VICON system when the aircraft was at rest, we estimated that the mean attitude noise from the VICON system was approximately 0.0119 degrees. However, attitude errors as high as 1 degree were observed. A 1 degree error in attitude at an altitude of 2 meters would yield a position error of approximately 4 centimeters. The second source of error came about because objects in the camera view frustrum would appear to have a light intensity value similar to that of the target. Therefore objects other than the target could be mistaken for the target and used for relative position estimation. It is easy to see how this could lead to large position estimation errors. Dull black cloth was taped to the floor with black gaffing tape in an attempt to prevent this occurrence and more complex target recognition techniques or heuristic methods could be used in future tests to prevent this from occurring.

Table 4.1: Relative position estimation flight results

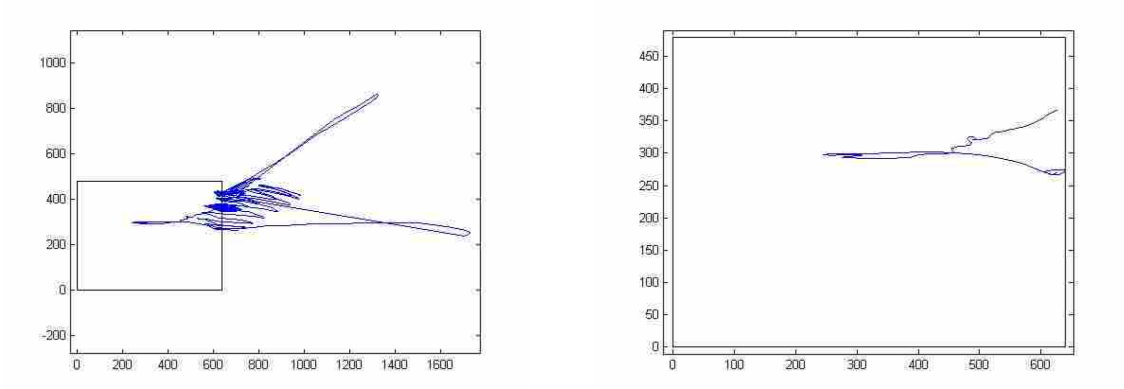
Flight Test	max error (m)	min error (m)	mean error (m)	standard deviation (m)	time (s)
1	0.6035	0.0063	0.1680	0.1424	143.25
2	0.5997	0.0162	0.1470	0.1892	156.10
3	0.5603	0.0175	0.1680	0.1780	92.55
4	0.5780	0.0038	0.2290	0.1987	137.30
5	1.1680	0.0020	0.2517	0.1020	129.30
6	0.4174	0.0100	0.2209	0.0998	133.05
7	0.9599	0.0334	0.2799	0.0817	158.25
8	0.7568	0.0108	0.3275	0.0974	269.15
9	0.4908	0.0023	0.1728	0.1314	135.50
10	0.4956	0.0099	0.2423	0.2065	224.30

CHAPTER 5. DESIRED ATTITUDE SATURATION

With an estimate of the UAV's position relative to the target, the next step becomes moving to the target while still keeping the target in view. To do this, a desired attitude is calculated that will take the UAV from its current location to the desired waypoint using the method described in Chapter 3. Given the current location of the UAV, the desired location of the UAV, the x and y body-frame velocities, and the roll, pitch, and yaw rates, the method discussed in Section 3.3 computes a desired quaternion that will navigate the UAV to the desired location. This chapter details the method used to determine a saturation limit that is applied to the desired attitude so that the ground target remains in view at all times.

To some extent the problem that the ground target is no longer visible when the VTOL UAV tilts to move towards the target is a self-correcting problem. Because the aircraft tilts in a way that will position the UAV over the target when the target goes out of view, the target is likely to come back into view soon after the target leaves the image frame. However, since the relative position estimation of the UAV is dependent on the ground target being in the image frame, it is important that the ground target remain in the image frame so that the UAV's position can be calculated. Additionally, the UAV mission objectives may require that the ground target of interest remain in view at all times such as in an intelligence, surveillance, and reconnaissance (ISR) scenario. Also, as the following example demonstrates, the image-attitude correlation problem is not always self correcting.

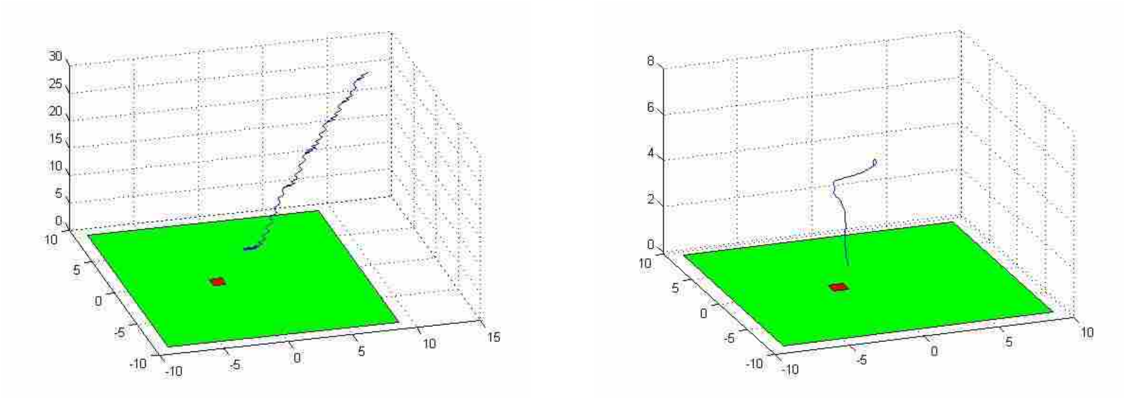
As a motivating example we consider a GPS-denied environment with a 0.2 meter/second cross wind coming from the west. The UAV is initialized at an altitude of five meters with the target in the right-hand side of the image frame. We also make a reasonable assumption that the automatic target recognition algorithm requires one second to acquire the target after which tracking is done with no added lag, each time the target goes out of the image frame it takes another full second to reacquire the ground target in the image frame. When no saturation is applied to the desired attitude, the UAV is not able to keep the target in view and the UAV ascends to reacquire the target, the target centroid in the image coordinate frame is shown in Figure 5.1a. While the target does reappear in the image frame for short periods of time, the 0.2 meter/second wind pushes the UAV away from the target and prevents the target from appearing in the image frame for longer than one second segments. In this scenario the UAV never reacquires the target once it is initially found and simply ascends indefinitely in search of the ground target. The position of the UAV can be seen constantly ascending in this simulated flight test in Figure 5.2a. When saturation is applied to the desired attitude of the UAV, the aircraft initially finds the ground target and the UAV moves in a way that keeps the ground target in view until an altitude of 0.5 meters is reached. The ground target centroid in the image frame can be seen in Figure 5.1b until the the UAV reaches an altitude of 0.5 meters. At 0.5 meters above the target, the ground target is no longer visible from the onboard camera due to the attitude that the aircraft is required to maintain to counter the effect of wind. The position of the UAV as it performs its vision-based landing in this flight test can be seen in Figure 5.2b. This example demonstrates a scenario when the desired attitude saturation technique presented in this chapter is required for precision landings. The absence of the desired attitude saturation method in this scenario leads to an ever ascending UAV that is unable to acquire the ground target of interest and therefore, is unable to estimate its own position.



(a) Location of the ground target centroid in the image frame for a simulated test flight when no desired attitude saturation is applied.

(b) Location of the ground target centroid in the image frame for a simulated test flight when desired attitude saturation is applied.

Figure 5.1: The target centroid in the image frame in a motivating example showing the need for desired attitude saturation, the rectangle defines the onboard camera's view.



(a) Flight path of the UAV for a simulated test flight when no desired attitude saturation is applied.

(b) Flight path of the UAV for a simulated test flight when desired attitude saturation is applied.

Figure 5.2: The UAV position in a motivating example showing the need for desired attitude saturation.

5.1 Desired Attitude Saturation in Quaternions

With the desired quaternion known, the next step becomes ensuring that the orientation of the aircraft at the desired quaternion will still allow the camera to view the ground target. We do this by saturating the error quaternion. The error quaternion is the error between the aircraft orientation quaternion and the desired quaternion, or in other words, the error quaternion rotates the aircraft from its current attitude to the desired attitude. The error quaternion is calculated as

$$q_{\text{error}} = \begin{bmatrix} \eta_{4\text{a}}\eta_{1\text{d}} + \eta_{3\text{a}}\eta_{2\text{d}} + \eta_{2\text{a}}\eta_{3\text{d}} - \eta_{1\text{a}}\eta_{4\text{d}} \\ -\eta_{3\text{a}}\eta_{1\text{d}} + \eta_{4\text{a}}\eta_{2\text{d}} - \eta_{1\text{a}}\eta_{3\text{d}} - \eta_{2\text{a}}\eta_{4\text{d}} \\ \eta_{2\text{a}}\eta_{1\text{d}} + \eta_{1\text{a}}\eta_{2\text{d}} + \eta_{4\text{a}}\eta_{3\text{d}} - \eta_{3\text{a}}\eta_{4\text{d}} \\ \eta_{1\text{a}}\eta_{1\text{d}} + \eta_{2\text{a}}\eta_{2\text{d}} + \eta_{3\text{a}}\eta_{3\text{d}} + \eta_{4\text{a}}\eta_{4\text{d}} \end{bmatrix},$$

and is defined in the body coordinate frame.

If the UAV is allowed to rotate by the error quaternion to the previously calculated desired quaternion, it is possible for the UAV to rotate in a manner that will cause the target to go out of view of the onboard camera. For this reason, the error quaternion is saturated so that the rotation commanded to position the UAV at its desired location, does not send the target out of the camera view. A quaternion can be thought of as a representation of a single rotation about a single axis of

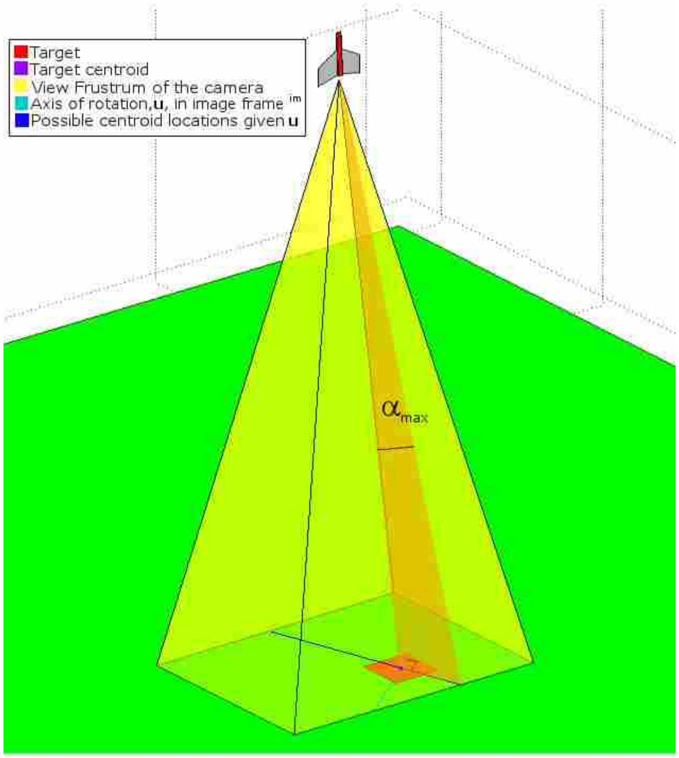

🟥 Target
🟪 Target centroid
🟨 View Frustrum of the camera
🟦 Axis of rotation,**u**, in image frame im
🟦 Possible centroid locations given **u**
α_{max}

Figure 5.3: This figure shows the angle α_{max}, the axis of rotation **u** in the image plane, and the view frustum of a tailsitter in hover flight.

rotation, **u**, by an angle α. We then calculate the maximum allowable angle, α_{max} before the target centroid goes out of view as shown in Figure 5.3. The error quaternion is then recalculated using this α_{max} and the original **u**, so that the resulting rotation does not move the target outside of the camera viewing area.

The vector component of the error quaternion, $\mathbf{u}_{\text{error}}$, is the axis of rotation from q_{aircraft} to q_{desired}. To find α_{max}, the vector component of the error quaternion is projected into the image frame. Similarly, we can project the first three components of the error quaternion, $[\eta_{1e}\ \eta_{2e}\ \eta_{3e}]^T$, into the image frame since this is $\mathbf{u}_{\text{error}}$ scaled by $\sin(\alpha/2)$ and the magnitude of the vector is of no import in the image frame. This transformation is shown as

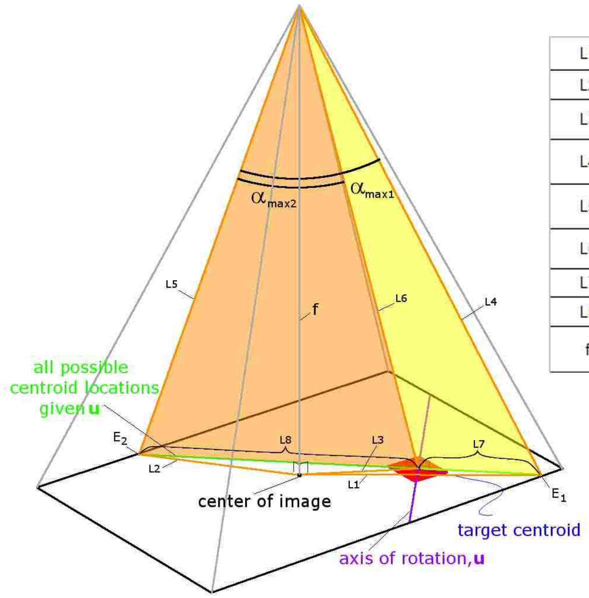
$$\begin{bmatrix} x_{\text{image}} \\ y_{\text{image}} \\ z_{\text{image}} \end{bmatrix} = K R^{c}_{g2} R^{g2}_{g1} R^{g1}_{g} \left(R^{g}_{b} \begin{bmatrix} \eta_{1\text{e}} \\ \eta_{2\text{e}} \\ \eta_{3\text{e}} \end{bmatrix} + \begin{bmatrix} x_{\text{gtrans}} \\ y_{\text{gtrans}} \\ z_{\text{gtrans}} \end{bmatrix} \right). \tag{5.1}$$

The variables x_{image} and y_{image} denote the 2D vector direction of the error quaternion rotated into the image plane. The z_{image} component on the left-hand side of Equation (5.1) is not used to compute the direction of the error quaternion in the image plane which implies that the three-dimensional error quaternion is projected onto the 2D image plane of the camera. By not using the z_{image} component, we are in essence assuming that the aircraft does not roll, or at least rolls very slowly, about its body frame x-axis once the vision-based navigation has commenced. If this assumption holds, then the set of all possible pixel locations of the centroid of the target, given the axis of rotation, is a line perpendicular to the vector $[x_{\text{image}}\ y_{\text{image}}]^T$ that passes through the current centroid of the target. For the practical application of this research, we will assume that the roll rate is small. The set of possible target centroid pixel locations is then a straight line given by

$$y = -\frac{x_{\text{image}}}{y_{\text{image}}} x - \frac{x_{\text{image}}}{y_{\text{image}}} X_{\text{pixel}} + Y_{\text{pixel}}. \tag{5.2}$$

In the undefined case that y_{image} equals zero, which means that the tailsitter is directly over the target in the image frame x direction, then the line perpendicular to $[x_{\text{image}}\ y_{\text{image}}]^T$ is simply $x = X_{\text{pixel}}$. With the line of possible target locations known, the locations where the line defined in Equation 5.2 crosses the image boundaries is calculated. These two locations are called E_1 and E_2 and are shown in Figure 5.4. E_1 and E_2 are calculated by finding where the perpendicular line specified in Equation 5.2, crosses the lines; x = 0, x = *image width*, y = 0, and y = *image height*. The intersections that lie at the border of the viewable area of the camera are the variables E_1 and E_2.

With E_1 and E_2 known, α_{max1} and α_{max2} can be calculated based on the distances shown in Figure 5.4. A short description of each of the distances needed to calculate α_{max1} and α_{max2} is



L1	distance from the center of the image to the point E_1
L2	distance from the center of the image to the point E_2
L3	distance from the center of the image to the centroid of the target
L4	distance from E_1 to the camera frame origin assuming a pinhole camera model
L5	distance from E_2 to the camera frame origin assuming a pinhole camera model
L6	distance from the target centroid to the camera frame origin assuming a pinhole camera model
L7	distance from the target centroid to the point E_1
L8	distance from the target centroid to the point E_2
f	focal length of the camera in pixels using the pinhole camera model, can be approximated as $(f_x + f_y)/2$

Figure 5.4: This figure shows the angles $\alpha_{\text{max}1}$ and $\alpha_{\text{max}2}$, the exit points E_1 and E_2, the axis of rotation $\mathbf{u}$ in the image plane, and the distances needed to calculate $\alpha_{\text{max}1}$ and $\alpha_{\text{max}2}$.

provided in the figure. The law of cosines is used to calculate $\alpha_{\text{max}1}$ and $\alpha_{\text{max}2}$ resulting in

$$\begin{aligned}\alpha_{\text{max}1} &= \arccos\left(\frac{L4^2+L6^2+L7^2}{2L4L6}\right),\\ \alpha_{\text{max}2} &= \arccos\left(\frac{L5^2+L6^2-L8^2}{2L5L6}\right).\end{aligned}$$

The smaller of $\alpha_{\text{max}1}$ and $\alpha_{\text{max}2}$ is the α_{max} used to recompute the error quaternion if α_{max} is less that the original angle of rotation.

The original desired angle of rotation, α_{original}, about the error quaternion can be calculated with the knowledge that the fourth component of the error quaternion is equal to $\cos(\alpha_{\text{original}})$. If α_{original} is greater than α_{max}, then a rotation by the angle α_{original} will cause the target centroid to leave the camera view frustum. For this reason the error quaternion is recalculated using α_{max}. Values for u_{1e}, u_{2e}, and u_{3e} are obtained using the original error quaternion and α_{original}. The saturated error quaternion is then computed as

$$q_{\text{error}} = \begin{bmatrix} u_{1\text{e}} \sin\left(\alpha_{\max}/2\right) \\ u_{2\text{e}} \sin\left(\alpha_{\max}/2\right) \\ u_{3\text{e}} \sin\left(\alpha_{\max}/2\right) \\ \cos\left(\alpha_{\max}/2\right) \end{bmatrix},$$

Since the first three components of the error quaternion are described in the body frame and are scaled by $\sin(\alpha/2)$, the error quaternion axis of rotation can be employed directly to calculate the aircraft actuator commands as

$$\delta_a = k_{\eta 1}\eta_{1e} - k_p p, \tag{5.3}$$

$$\delta_e = -(k_{\eta 2}\eta_{2e} - k_q q), \tag{5.4}$$

$$\delta_r = k_{\eta 3}\eta_{3e} - k_r r. \tag{5.5}$$

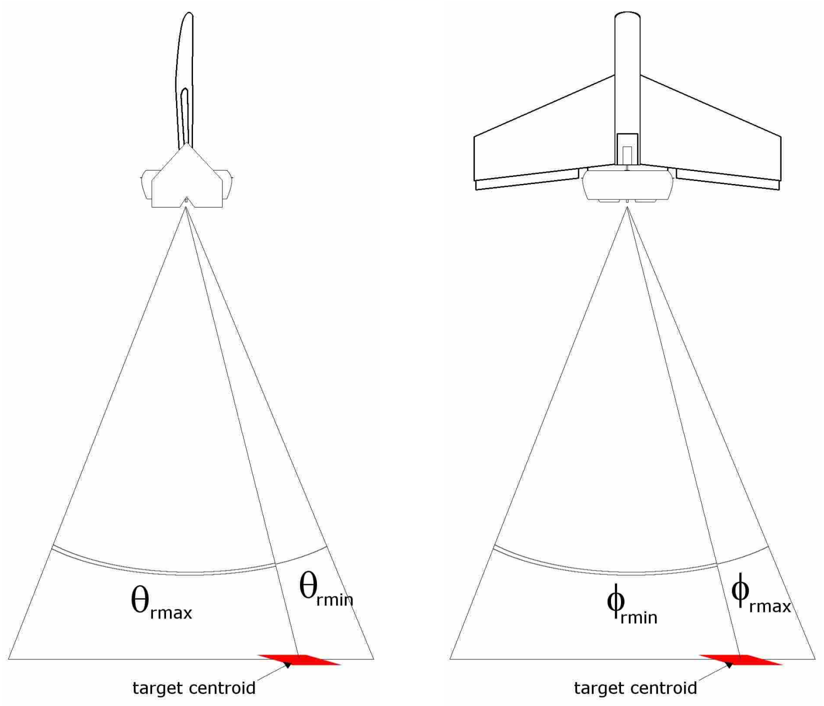
as was initially described in Section 3.1.1. As the UAV nears the desired location, the scaling of $\sin(\alpha/2)$ becomes increasingly smaller, consequently, the commanded actuator input also becomes smaller. Damping is provided by including roll rate (p), pitch rate (q), and yaw rate (r) terms to the actuator calculations. The terms k_{η_1}, k_{η_2}, k_{η_3}, k_{p}, k_{q}, and k_{r} are proportional and derivative gains.

5.2 Desired Attitude Saturation in Euler Angles

Desired attitude saturation can also be calculated for Euler roll and pitch angles directly from the location of the target origin in the image plane. The resulting maximum and minimum roll and pitch angle will be relative to the aircraft's current orientation and not in reference to some inertial coordinate frame.

5.2.1 Pitch Saturation Angles

The maximum and minimum pitch angles relative to the aircraft's attitude, as shown in Figure 5.5, are based on the x location of the target centroid in the image frame, the horizontal field of view of the camera lens, and intrinsic parameters of the camera itself. The relative maximum



(a) Maximum and minimum pitch angles.

(b) Maximum and minimum roll angles.

Figure 5.5: Maximum and minimum euler angles used to keep ground target in view.

and minimum pitch angles that allow the onboard camera to see the ground target of interest are calculated as

$$\begin{aligned} \theta_{\text{rmax}} &= \frac{H_{\text{fov}}}{2} - \tan^{-1}\left(\frac{\frac{I_{\text{width}}}{2} - x_{\text{pixel}}}{f_x}\right), \qquad (5.6) \\ \theta_{\text{rmin}} &= \frac{H_{\text{fov}}}{2} + \tan^{-1}\left(\frac{\frac{I_{\text{width}}}{2} - x_{\text{pixel}}}{f_x}\right). \qquad (5.7) \end{aligned}$$

Where H_{fov} is the horizontal field of view of the camera lens in radians, I_{width} is the width of the image in pixels, x_{pixel} is the centroid of the target in the x direction in units of pixels, and f_x is the x-direction focal length of the camera as derived from a camera calibration. The angles calculated in Equations 5.6 and 5.7 are relative to the current orientation of the aircraft. This means that if the aircraft is at a pitch angle of 13 degrees, for example, and the θ_{rmax} and θ_{rmin} angles are calculated as 12 degrees and 36 degrees, then the angles θ_{max} and θ_{min} would be 25 and -23

degrees respectively. The conversion from $\theta_{\text{rmax}}/\theta_{\text{rmin}}$ and $\theta_{\text{max}}/\theta_{\text{min}}$ is performed as

$$\theta_{\text{max}} = \theta + \theta_{\text{rmax}} \;, \tag{5.8}$$

$$\theta_{\text{min}} = \theta - \theta_{\text{rmin}} \;, \tag{5.9}$$

where θ is the current pitch angle of the aircraft as measured from the inertial coordinate frame. The angles θ_{max} and θ_{min} are then used to saturate the θ_{desired} command derived from the navigational controller.

5.2.2 Roll Saturation Angles

The maximum and minimum roll angles, as shown in Figure 5.5, are based on the y location of the target centroid in the image frame, the vertical field of view of the camera lens, and some intrinsic parameters of the camera itself. The relative maximum and minimum roll angles that allow the onboard camera to see the ground target of interest are calculated as

$$\phi_{\text{rmax}} = \frac{V_{\text{fov}}}{2} - \tan^{-1}\left(\frac{\frac{I_{\text{height}}}{2} - y_{\text{pixel}}}{f_y}\right) \;, \tag{5.10}$$

$$\phi_{\text{rmin}} = \frac{V_{\text{fov}}}{2} + \tan^{-1}\left(\frac{\frac{I_{\text{height}}}{2} - y_{\text{pixel}}}{f_y}\right) \;. \tag{5.11}$$

The term V_{fov} is the vertical field of view of the camera lens in radians, I_{height} is the height of the image in pixels, y_{pixel} is the centroid of the target in the y direction in units of pixels, and f_y is the y-direction focal length of the camera as derived from a camera calibration. Similar to the relative maximum and minimum pitch angles, the angles calculated in Equations 5.10 and 5.11 are relative to the current orientation of the aircraft. The conversion from $\phi_{\text{rmax}}/\phi_{\text{rmin}}$ to $\phi_{\text{max}}/\phi_{\text{min}}$ is

$$\phi_{\text{max}} = \phi + \phi_{\text{rmax}} \;, \tag{5.12}$$

$$\phi_{\text{min}} = \phi - \phi_{\text{rmin}} \;, \tag{5.13}$$

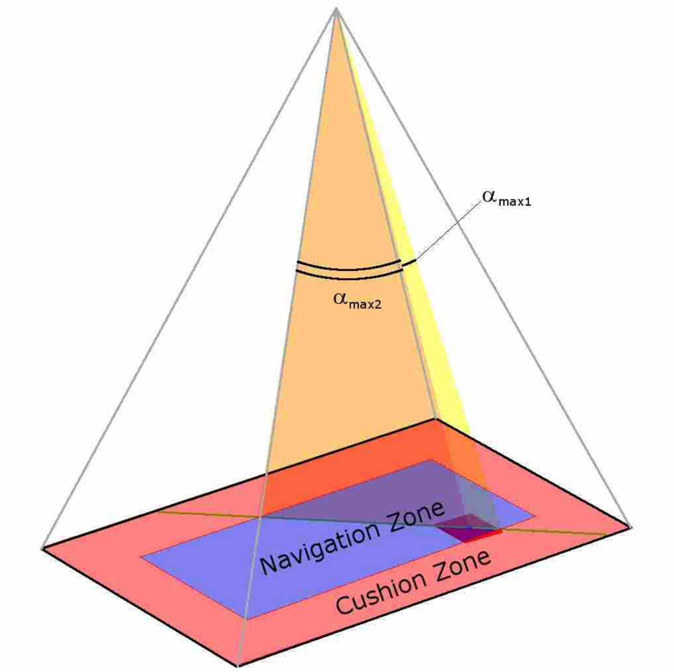


Figure 5.6: This figure shows the cushion and navigation zones and the angles α_{max1} and α_{max2} based on those zones.

where ϕ is the current roll angle of the aircraft as measured from the inertial coordinate frame. The angles ϕ_{max} and ϕ_{min} are then used to saturate the ϕ_{desired} command derived from the navigational controller.

5.3 Navigation and Cushion Zones

A cushion zone around the inside of the camera image is used to prevent disturbances from forcing the target to leave the image frame. This cushion zone also allows the aircraft to continue to navigate towards the target if the target is at the very edge of the image. If the target centroid is located in this cushion zone, the aircraft is commanded to maintain its current x and y positions and gain altitude until the centroid moves out of the cushion zone and into the navigation zone as shown in Figure 5.6. We also used the cushion and navigation zones to calculate α_{max1} and α_{max2} to account for any underdamped response to the control or overshoot that might cause the target to momentarily leave the image frame. The use of a cushion zone is also advantageous when the assumption of no roll rate after the target has been detected is violated. This is because the use of a cushion zone oversaturates the error quaternion and therefore helps keep the target in view if slight rolling occurs. A cushion zone of approximately 5 percent of the image dimensions was found to be adequate for keeping the target in view in simulation.

5.4 Results

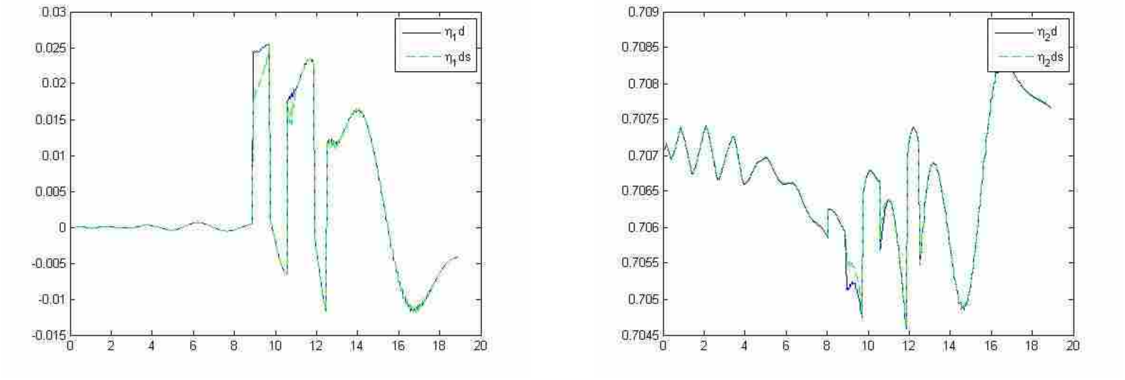
The method used to saturate the desired attitude as described in this chapter was validated in simulation and in hardware to ensure that the ground target used for relative position estimation never left the image frame. A cushion zone of approximately five percent of the image width was also used in simulation and hardware testing to ensure that the target did not go out of view. This section presents the attitude saturation results from simulation and hardware flight testing.

5.4.1 Attitude Saturation Simulation Results

The desired attitude saturation method described in this chapter was tested using the simulation environment described in Section 2.1. The simulated tailsitter was initialized at some location close to the ground target to be used for navigation. It was assumed that the position of the UAV was known for navigation before the ground target came into view. Once the target came into view, only the estimate of the position of the aircraft derived through vision was used for navigation.

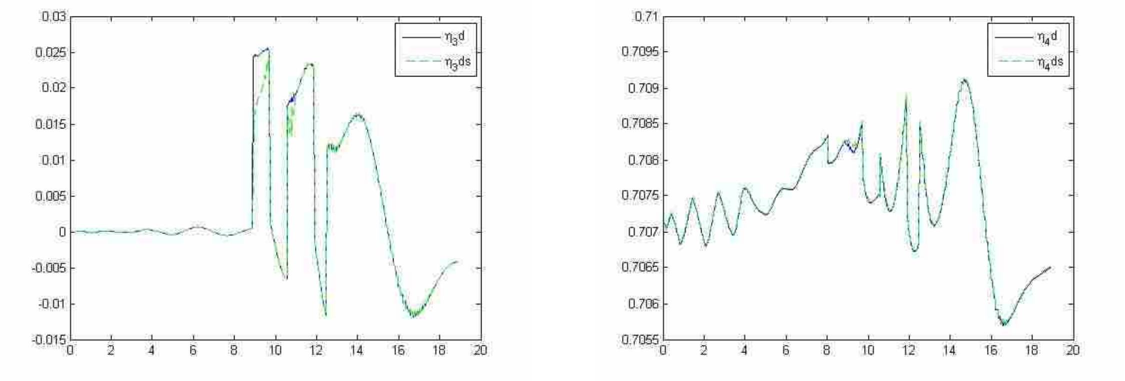
The tailsitter was commanded to gain altitude, maintaining its current position in North and East, until the target came into the onboard camera's view. Once the target came into view, the relative position to the target was calculated as described in Chapter 4 and a desired attitude was calculated. The desired attitude was then saturated using the method described in this chapter if it was determined that the desired attitude would cause the target to go out of frame.

Figure 5.7 shows the results from one of the simulation test runs. Saturation of the desired attitude occurs at 15 and 17 seconds as is apparent from the differences between the desired attitude component (shown in blue) and the saturated desired attitude component (shown in green). When no saturation was applied for the same initial conditions, the target did in fact go out of the field of view of the camera as is shown in Figure 5.8a. When saturation was applied, the target never went out of view once it entered the image as is shown in Figure 5.8b.



(a) Comparison of the η_1 component of the desired (d), and saturated desired (ds) quaternions.

(b) Comparison of the η_2 component of the desired (d), and saturated desired (ds) quaternions.



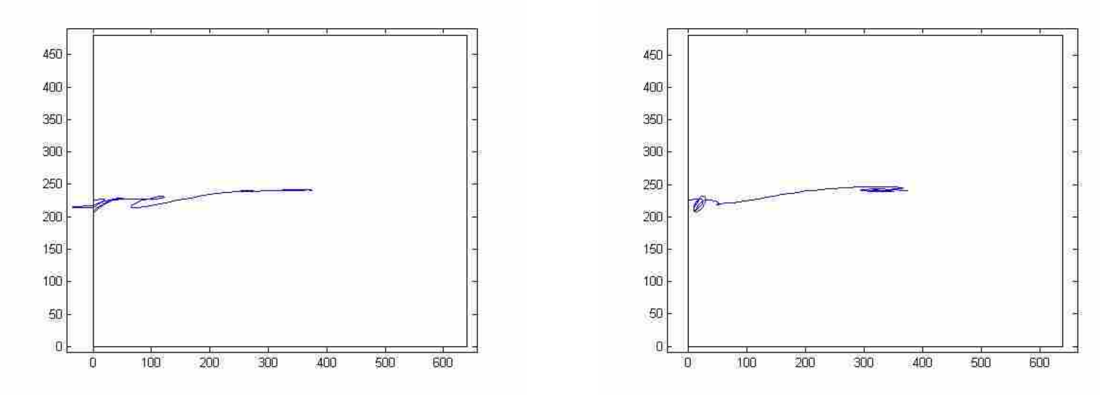
(c) Comparison of the η_3 component of the desired (d), and saturated desired (ds) quaternions.

(d) Comparison of the η_4 component of the desired (d), and saturated desired (ds) quaternions.

Figure 5.7: Comparison of the desired, and saturated desired quaternion components from simulation.

5.4.2 Attitude Saturation Hardware Results

The method of desired attitude saturation described in this chapter was also tested using the flight test setup described in Section 2.2. The helicopter took off from a location about one meter from the ground target and gained altitude until the target came into view. Position estimates from the VICON motion capture system were initially used to control the aircraft until the target came into view. Once the target was in view, the position of the UAV was estimated using vision as described in Chapter 4 and the attitude saturation techniques described in this chapter were used to keep the target in view. The UAV then navigated over the target, using both the vision-based position estimation and attitude saturation techniques, and descended until it reached the target.



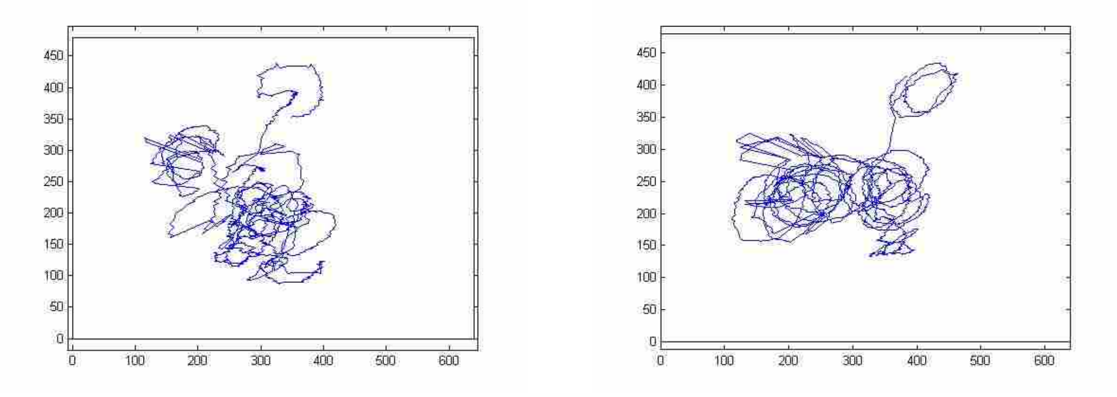
(a) Location of the target centroid in the image frame when no saturation is applied to the desired attitude.

(b) Location of the target centroid in the image frame when saturation is applied to the desired attitude to keep the target centroid in view.

Figure 5.8: Location of the target centroid in the image frame in simulation, the rectangle defines the onboard camera's view.

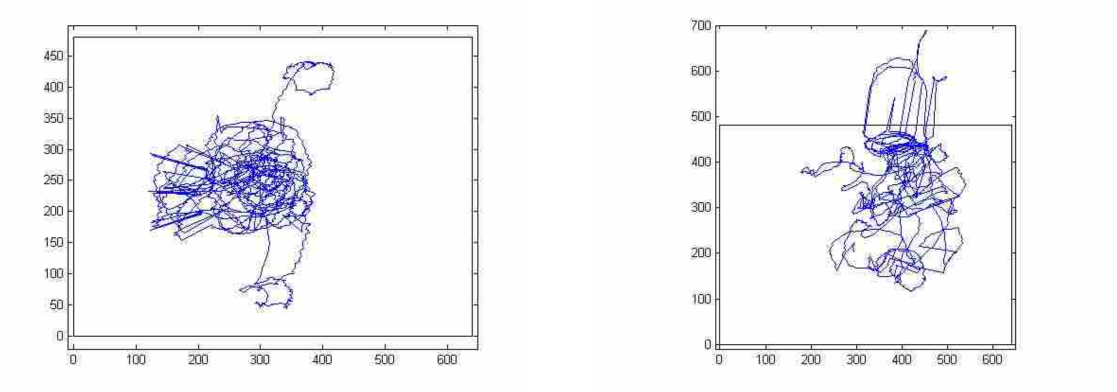
Figure 5.9 shows the location of the target centroid in the image frame for all ten hardware test flights. These ten test flights show the flight data from the instance that the vision-based controller was turned on with the ground target in view of the onboard camera until the aircraft descended to an altitude of 0.5 meters over the target. These ten test flights also implemented the desired attitude saturation technique described in this chapter to keep the ground target in view for the duration of the flight. As can be seen in Figures 5.9d, 5.9f, 5.9i, and 5.9j, the target did not always stay in the image frame even with the use of the desired attitude saturation technique described in this chapter. However, as the flight data in Figure 5.10 shows, the ground target centroid would have never gone out of view if the UAV had tracked the commanded attitude value. Figure 5.10 shows the location of the target centroid in the image frame if the UAV had achieved the desired attitude commanded to it. The implications of this figure are important because it shows that while the target centroid did not always stay in the image frame, this UAV was never commanded to an attitude that would force the target centroid out of the image frame. Therefore, we can attribute the instances in which the target centroid left the image frame to the UAV's inability to track a desired attitude with enough precision to prevent to target from leaving the image frame.

In hardware flight testing the target would often go out of view and not be regained at altitudes below 18 inches. This is attributed to the fact that the oscillatory behavior of the helicopter as seen in Figures 3.4a and 3.4b, has a greater effect on what is seen in the image frame at lower



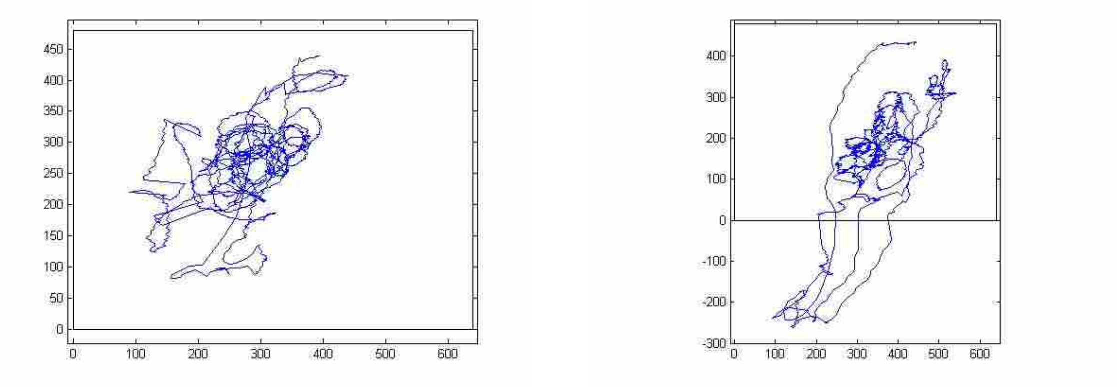
(a) Test flight 1

(b) Test flight 2.



(c) Test flight 3.

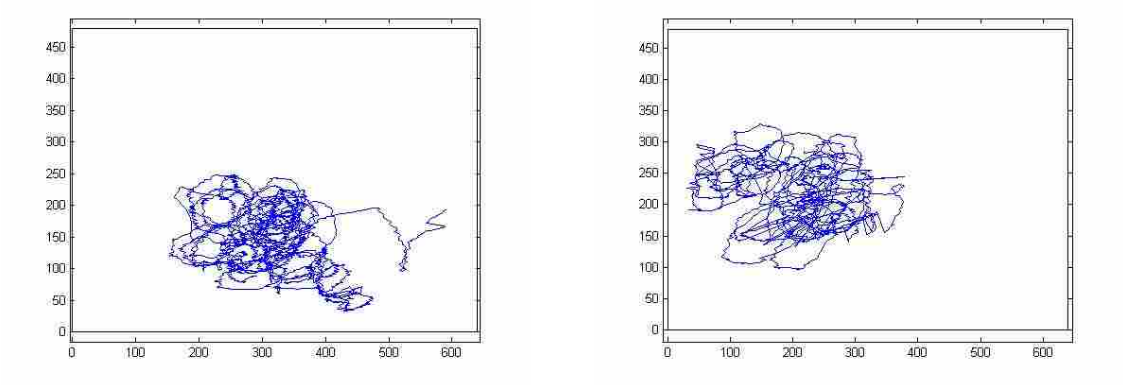
(d) Test flight 4.



(e) Test flight 5.

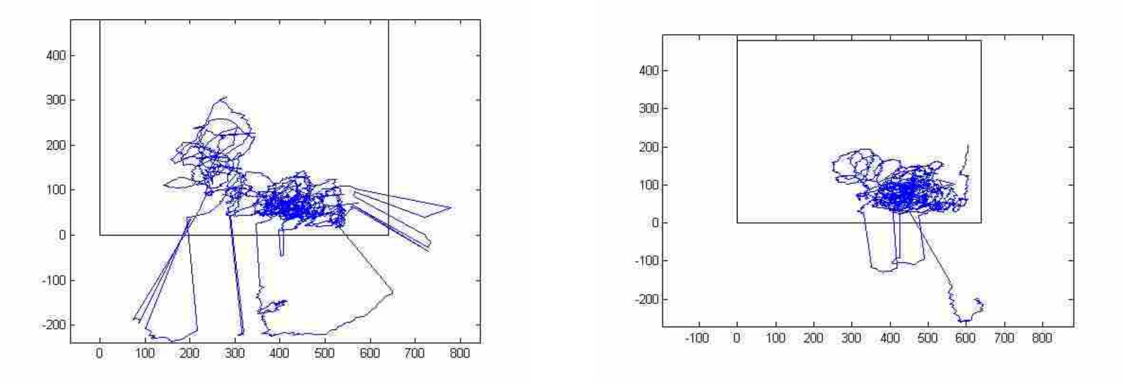
(f) Test flight 6.

Figure 5.9: Location of the target centroid in the image frame from the different test flights, the rectangle defines the onboard camera's view.



(g) Test flight 7.

(h) Test flight 8.



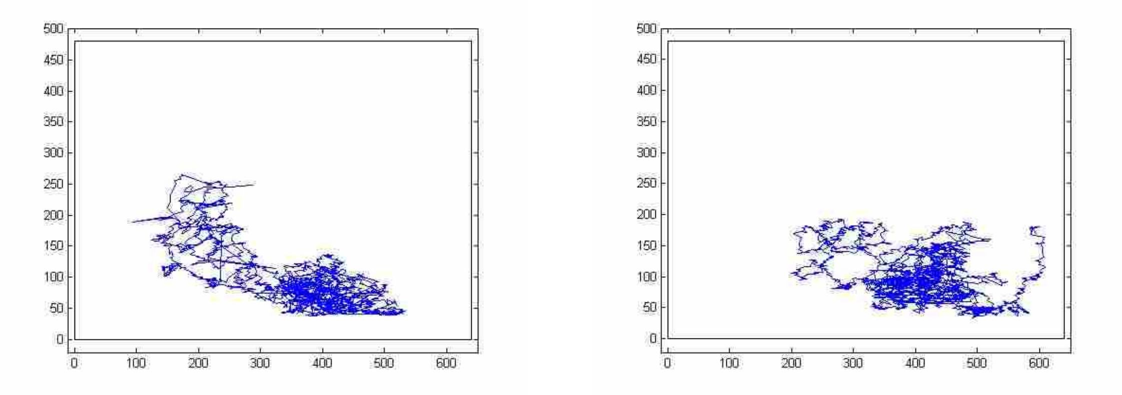
(i) Test flight 9.

(j) Test flight 10.

Figure 5.9: Location of the target centroid in the image frame from the different test flights, the rectangle defines the onboard camera's view.

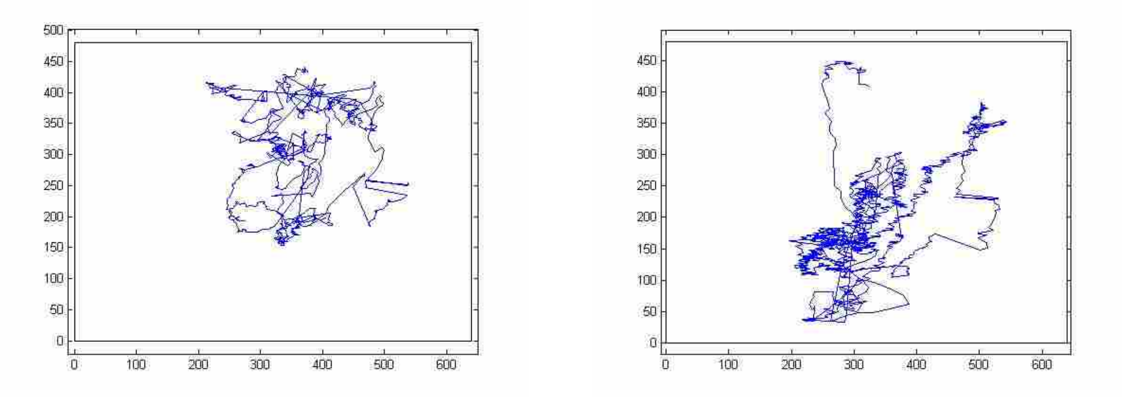
attitudes. Ground effects from flying close to the ground could also account for this occurrence. For this reason, the plots shown in Figure 5.9 only present flight data down to 0.5 meters altitude AGL.

Figure 5.11 shows an example of when desired attitude saturation was used to keep the target in the image frame in flight testing. The desired pitch angle as calculated from the navigational controller is shown as a red dashed line while the minimum and maximum pitch angles are shown in green. Saturation of the desired pitch angle occurs when the desired pitch angle is not between the maximum and minimum pitch angles. At times when the desired pitch angle is not in the window of acceptable pitch angles the nearest saturation value is used in its place. Saturation of the desired pitch angle can be seen in Figure 5.11 at times beginning at 83 seconds and occurring periodically throughout the figure. For the data shown, the pitch angle experiences 11 saturation



(a) Test flight 4.

(b) Test flight 6.



(c) Test flight 9.

(d) Test flight 10.

Figure 5.10: Location of the target centroid in the image frame from the different test flights if the UAV had achieved the desired attitude, the rectangle defines the onboard camera's view.

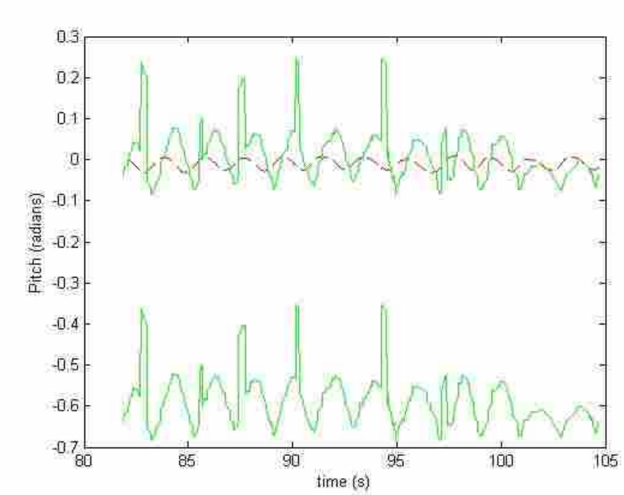


Figure 5.11: This figure shows the the desired attitude in red as calculated from the navigational controller and the maximum and minimum desired attitude limits in green, saturation occurs when the desired attitude is outside of the saturation window.

occurrences and the figure ends in a state of saturation. Using this window of acceptable pitch and roll values enabled the helicopter to navigate over the ground target while preventing a desired attitude to be commanded that would cause the target to move outside the image frame.

Table 5.1 shows the saturation occurrences for all ten test flights as well as the cumulative time that the desired attitude was saturated. A noticeable difference can be seen between tests one through six and tests seven through ten where much fewer saturation occurrences are observed. This is attributed to the compiling method used to run the vision navigation algorithm in C++ for these tests. Specifically tests seven through ten were compiled in debug mode while tests one through six were compiled in release mode. This means that the vision programs in tests one through six were running much faster and many more saturation occurrences are seen as a consequence.

We did not collect vision-based landing flight test data with the desired attitude saturation algorithm turned off. However, we see from Table 5.1 that each flight test had several instances when a desired attitude would have been commanded that would have sent the target centroid out of the image frame if the attitude saturation algorithm had not been used.

Table 5.1: Attitude saturation flight results

Flight Test	saturation occurrences	time(s)
1	27	9.6
2	24	3.75
3	13	5.35
4	16	2.6
5	22	9.35
6	24	5.2
7	3	0.75
8	4	0.25
9	6	0.5
10	4	3.45

CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS

This chapter discusses the conclusions that we have reached as a result of the research presented in this work. Our recommendations for future related research and potential improvements to the research presented are also given.

6.1 Conclusions

The objective of this research was to perform vision-based landings of a tailsitter aircraft with sub-meter accuracy using only an onboard camera, aircraft orientation information and the aircraft's altitude AGL. A method of calculating the position of the UAV relative to a visually detected target was presented using only the information described above, and was shown to have an average accuracy of 22 centimeters in hardware flight testing.

One of the main difficulties encountered in this research was the problem of keeping the ground target in the camera's image frame while still tracking desired attitudes to navigate over the target. For this reason, one focus of this research has been in calculating the maximum and minimum components of the aircraft's attitude that still allow the ground target to be visible to the UAV's onboard camera. A method was presented to find the maximum and minimum attitude components of the aircraft that would allow the UAV to rotate until the target centroid was at the edge of the viewing area. This method was presented in both quaternion and Euler angle attitude representations. These maximum and minimum attitude values were then used to saturate the desired attitude so that the aircraft was not commanded to an attitude that would move the target out of the viewable image frame. The vision-based landing methods presented in this paper were tested in simulation and hardware flight testing of an autonomously controller RC helicopter. In six of the ten flight tests the UAV navigated over the target and descended to an altitude of 0.5 meters using vision without causing the target centroid to go out of the camera view. In the remaining four test flights the target did go out of view of the onboard camera and was eventually regained and the vision-based landing procedure resumed until the UAV reached an altitude of 0.5 meters above the target. Upon further investigation into these four test flights we found that the UAV was never commanded to an attitude that would have caused the target to leave the image frame. Instead the target left the image frame due to the inability of the UAV to precisely track a desired attitude.

As mentioned in Chapter 1, the research presented in this work could have direct applications in both the civilian and military marketplaces. With the research and conclusions that we have presented it is easy to see that VTOL UAVs already have the technology at hand to perform precision drop missions with submeter accuracy, perch-and-stare missions from a platform with a sufficiently large base, and precision landings with submeter accuracy. The ability to land with such accuracy allows the UAV to land in a variety of GPS denied environments such as urban canyons and thickly forested regions, or at locations where GPS's accuracy is insufficient such as on a ship deck. Much of the research that we have performed acts as a foundation for further research that could extend the capabilities of unmanned VTOL aircraft and precision landings even further.

6.2 Recommendations for Future Work

There are several aspects of our work that could be extended to further the capabilities of unmanned VTOL flight. In this section we will discuss possible research directions that we feel would yield valuable results.

The method we have presented for vision-based landings is dependent on an estimate of the aircraft's attitude that we assume is known. It is likely that this estimate would come from an onboard inertial measurement unit (IMU). Using more complicated target geometry and computer vision techniques similar to those presented in [24] and [14, 15], it is possible to solve for the full pose of the aircraft from vision if there is sufficient information known about the ground target's geometry. This reduces the amount of hardware required for a UAV to navigate autonomously and therefore allows for smaller and lighter weight UAVs to perform autonomous tasks. This method would however seriously increase the computational workload. A more complex target pattern or geometry could also allow the UAV to navigate all the way to the ground without the target completely filling the image frame therefore losing some position accuracy.

The research we have presented has been in landing a VTOL UAV on a static ground target. The task of landing a VTOL UAV on a moving target has unique challenges. Much like the problem of having to tilt the UAV towards the target and therefore forcing the target out of view described in Chapter 5, landing on a moving target is further complicated by the target itself moving outside of the image frame. This scenario would lead to smaller attitude saturation angles and further limit the aircraft's ability to track the moving target. Further research could look into gimballing the onboard camera to track or land on a moving target or to enable vision-based navigation.

Future research could also be performed in bounding the position error at times when the ground target goes out of view with the use of a Kalman filter or extended Kalman filter. Information from optic flow sensed in the image frame, or from rate gyros and accelerometers from the onboard IMU, could be used to approximate the UAV's movement since the last instance that the ground target had been acquired in the image frame. This technique would aid in reacquiring the ground target if the UAV ever lost sight of the ground target. Vision-based estimates of position could also be combined with GPS and IMU data in a Kalman filter to obtain a more reliable position estimate.

When performing the maximum and minimum attitude calculations in Chapter 5, we assumed that the aircraft did not roll or rolled very slowly. Further research could address keeping the ground target in view of an onboard camera while allowing for rolling of the tailsitter aircraft.

Finally, when performing the position estimation of the UAV relative to the target in Chapter 4, a flat earth assumption was made. In several situations the approximation that the ground over which the UAV is flying is flat, holds for the small window of the Earth that we are considering. At other times, however, such as flying over an urban environment with tall buildings, this assumption does not hold. If the target of interest is located in a highly non-planar scene, the landing method that we have proposed may not work. Further research could address vision-based landings in such environments.

REFERENCES

[1] Ransone, R., 2002. “A photo history of VSTOL aircraft and their contributions.” In *AIAA X-Plane Symposium*.

[2] Beard, R. W., and McLain, T. W. Guidance and control of autonomous miniature air vehicles.

[3] Sabatini, R., and Palmeri, G., 2008. Differential global positioning systems (DGPS) for flight testing Tech. Rep. ISBN 978-92-837-0041-8, NATO Research and Technology Organisation, October.

[4] Knoebel, N. B., Osborne, S. R., Matthews, J. S., Eldredge, A. M., and Beard, R. W., 2006. “Computationally simple model reference adaptive control for miniature air vehicles.” In *American Control Conference*.

[5] Knoebel, N. B., Osborne, S. R., Snyder, D. O., McLain, T. W., Beard, R. W., and Eldredge, A. M., 2006. “Preliminary modeling, control, and trajectory design for miniature autonomous tailsitter.” In *AIAA Conference on Guidance, Navigation, and Control*.

[6] Knoebel, N., 2007. “Adaptive quaternion control of a miniature tailsitter UAV.” Master’s thesis, Brigham Young University.

[7] Barber, D. B., McLain, T. W., Taylor, C., and Beard, R. W., 2007. “Vision-based landing of fixed-wing miniature air vehicles.” In *2007 AIAA Infotech@Aerospace Conference*.

[8] Edwards, B., Archibald, J., Fife, W., and Lee, D., 2007. “A vision system for precision mav targeted landing.” In *Proc. International Symposium on Computational Intelligence in Robotics and Automation CIRA 2007*, pp. 125–130.

[9] Hubbard, D., Morse, B., Theodore, C., Tischler, M., and McLain, T., 2007. “Performance evaluation of vision-based navigation and landing on rotorcraft unmanned aerial vehicle.” In *IEEE Workshop on Applications of Computer Vision 2007*.

[10] Hintze, J., Christian, D., Theodore, C., Tischler, M., McLain, T., and Montgomery, J., 2004. “Automated landing of a rotorcraft UAV in a non-cooperative environment.” In *American Heliopter Society International 60th Annual Forum*.

[11] Theodore, C., Hubbard, D., Dai, W., , and Takahashi, M., 2005. “Full mission simulation of a rotorcraft UAV in a non-cooperative environment.” In *American Helicopter Society 61st Annual Forum*.

[12] Theodore, C., Hubbard, D., Ansar, A., and Whalley, M., 2006. “Flight trials of a rotorcraft unmanned aerial vehicle landing autonomously at unprepared sites.” In *American Helicopter Society 62nd Annual Forum*.

[13] S.Saripalli, Montgomery, J., and Sukhatme, G., 2003. "Visually guided landing of an unmanned aerial vehicle." pp. 371–380.

[14] Shakernia, O., Ma, Y., Koo, T. J., Hespanha, J., and Sastry, S. S., 1999. "Vision guided landing of an unmanned air vehicle." In *38th Conference on Decision and Control*.

[15] Sinopoli, B., Micheli, M., Donato, G., and Koo, T. J., 2001. "Vision based navigation for and unmanned aerial vehicle." In *IEEE International Conference on Robotics and Automation*.

[16] Sharp, C. S., Shakernia, O., and Sastry, S. S., 2001. "A vision system for landing an unmanned aerial vehicle." In *IEEE International Conference on Robotics and Automation*.

[17] How, J. P., Teo, J., and Michini, B., 2008. "Adaptive flight control experiments using RAVEN." In *14th Yale Workshop on Adaptive and Learning Systems*, pp. 205–210.

[18] `http://aviones.sourceforge.net/`.

[19] `http://opencv.willowgarage.com/wiki/`.

[20] Kuipers, J. B., 2002. *Quaternions and Rotation Sequences*. Princeton University Press.

[21] Shuster, M., 1993. "A survey of attitude representations." *Journal of Astronautical Sciences,* **41**(4).

[22] Barber, D. B., Redding, J. D., McLain, T. W., Beard, R. W., and Taylor, C. N. "Vision-based target geo-location using a fixed-wing miniature air vehicle." *J. Intell. Robotics Syst.,* **47**(4), pp. 361–382.

[23] Sonka, M., Hlavac, V., and Boyle, R., 2008. *Image Processing, Analysis, and Machine Vision.*, 3 ed. 7 Thomson Learning, Toronto, Ontario, Canada.

[24] Lu, C.-P., Hager, G. D., and Mjolsness, E., 1998. Fast and globally convergent pose estimation from video images.

[25] Phillips, W. F., 2004. *Mechanics of Flight*. Wiley, New Jersey.

APPENDIX A. DERIVATION OF THE DESIRED QUATERNION

For completeness, the derivation of the desired quaternion in hover flight is shown as taken from Ref. [5]. The notation used to represent the different components of a quaternion is

$$\overline{\eta}_{(\cdot)} = \begin{pmatrix} \eta_{(\cdot)} \\ \eta_{4(\cdot)} \end{pmatrix},$$

where,

$$\eta_{(\cdot)} = \begin{pmatrix} \eta_{1(\cdot)} \\ \eta_{2(\cdot)} \\ \eta_{3(\cdot)} \end{pmatrix},$$

and the symbol $(\cdot)$ is the location where the name of the specific quaternion is given. Using this notation, the equation for the desired quaternion is

$$\overline{\eta}_{\text{desired}} = \overline{\eta}_{\text{correction}} \otimes \overline{\eta}_{\text{vertical}} \,, \tag{A.1}$$

where η_{vertical} is the vertical quaternion which points straight up in the vehicle frame and is defined as

$$\overline{\eta}_{\text{vertical}} = \begin{pmatrix} 0 \\ \sqrt{2}/2 \\ 0 \\ \sqrt{2}/2 \end{pmatrix}.$$

The $\otimes$ symbol denotes quaternion multiplication and is defined as

$$\overline{\eta}'' = \overline{\eta}' \otimes \overline{\eta} = \begin{pmatrix} \eta_4 \eta' + \eta_4' \eta - \eta' \times \eta \end{pmatrix}. \tag{A.2}$$

The × symbol in Equation (A.2) simply denotes the cross product of two vectors.

The term $\overline{\eta}_{\text{correction}}$ in Equation (A.1) is the primary correction quaternion that describes the rotation needed for x-y position tracking in the inertial frame and is

$$\overline{\eta}_{\text{correction}} = \overline{\eta}_{\text{cv}} \otimes \overline{\eta}_{\text{cp}} . \tag{A.3}$$

The term $\overline{\eta}_{\text{cp}}$ from Equation (A.3) is the correction quaternion based on the position error and is defined in as

$$\eta_{\text{cp}} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} \frac{(x-x_d)}{\|e_p\|} \\ \frac{(x-x_d)}{\|e_p\|} \\ 0 \end{pmatrix} \quad \text{and} \quad \eta_{4\text{cp}} = \gamma \parallel e_p \parallel .$$

The terms x_d and y_d refer to the desired position of the aircraft in the inertial frame, γ is a gain, and $\parallel e_p \parallel$ is the norm of the position error and is defined as

$$\parallel e_p \parallel = \sqrt{(x-x_d)^2 + (y-y_d)^2} .$$

The term, $\overline{\eta}_{\mathrm{cv}}$, from Equation (A.3) is the correction quaternion based on body frame velocities and provides damping, it is defined as

$$\eta_{\mathrm{cv}} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} \frac{w}{\sqrt{v^2+w^2}} \\ \frac{v}{\sqrt{v^2+w^2}} \\ 0 \end{pmatrix} \quad \text{and} \quad \eta_{4\mathrm{cv}} = \zeta \sqrt{v^2+w^2}\ .$$

The terms v and w refer to the body frame velocities of the aircraft in the y-axis and z-axis directions respectively. The term ζ is a gain.

APPENDIX B. SIMULATION FORCES, TORQUES, AND EQUATIONS OF MOTION

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 B.1. The actual values used in calculations with brief descriptions are shown in Table B.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.

B.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 of [6]), 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):

$$\omega_p = C_{\omega_p 0} + C_{\omega_p,E} E + C_{\omega_p,\delta_t} \delta_t. \tag{B.1}$$

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 $C_{\omega_p,\delta_t} = 7.28$.

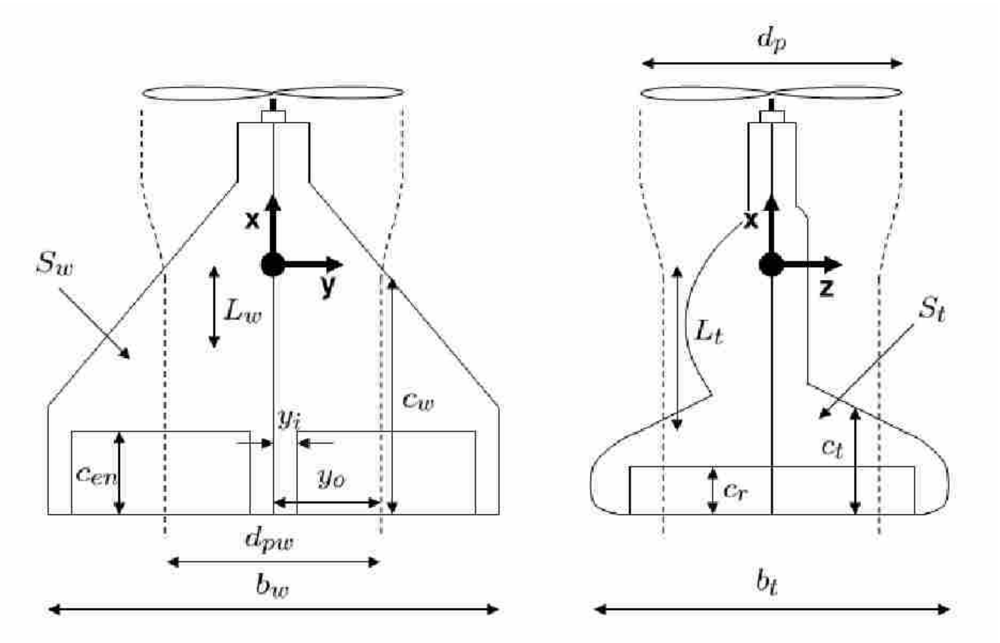


Figure B.1: Tailsitter geometry is shown with nomenclature.

Table B.1: Table of physical parameters used in simulation

Variable	Value	Description
S_w	0.203	Wing area, m^2
S_t	0.151	Tail area, m^2
c_w	0.230	Wing cord, m
c_t	0.120	Tail cord, m
c_{en}	0.120	Elevon cord, m
c_r	0.120	Rudder cord, m
b_w	1.0	Wing span, m
b_t	0.280	Tail span, m
L_w	0.038	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	0.130	y distance from center of gravity to outer edge of elevon, m
y_i	0.010	y distance from center of gravity to inner edge of elevon, m
d_p	0.280	Propeller diameter, m
d_{pw}	0.280	Propeller wash diameter, m
g	9.81	Gravity, m/s^2
μ	1.307	Mass, kg
ρ	1.27	Air density, kg/m^2
J_{xx}	0.315	x-axis moment of inertia, $kg{\cdot}m^2$
J_{yy}	0.200	y-axis moment of inertia, $kg{\cdot}m^2$
J_{zz}	0.058	z-axis moment of inertia, $kg{\cdot}m^2$
J_{xz}	0.000	cross-product of inertia, $kg{\cdot}m^4$

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 of [6]), given as

$$V_p = C_{V_p,\omega_p}\omega_p, \tag{B.2}$$

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:

$$V_{pw} = V_p - V. \tag{B.3}$$

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

$$F_{x-p} = \frac{1}{2}\rho A_p(V_p^2 - V^2). \tag{B.4}$$

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

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

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

B.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:

$$\begin{aligned}
F_{x-pw} &= 0.0, \\
F_{y-pw} &= -F_r(V_{pw}, \delta_r), \\
F_{z-pw} &= -F_e(V_{pw}, \delta_e), \\
l_{pw} &= M_a(V_{pw}, \delta_a), \\
m_{pw} &= -F_e(V_{pw}, \delta_e)L_{en} + M_e(V_{pw}, \delta_e), \\
n_{pw} &= F_r(V_{pw}, \delta_r)L_r + M_r(V_{pw}, \delta_r).
\end{aligned}$$

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.

B.2.1 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 produces 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

$$(\Delta\widetilde{C}_{Lw})_{\delta_a} = \pm\widetilde{C}_{Lw,\alpha}\widetilde{\varepsilon}_{en}\delta_a, \tag{B.6}$$

where $\widetilde{}$ 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

$$C_{\ell,\delta_a} = -\frac{2\widetilde{C}_{Lw,\alpha}}{S_w b_w}\int_{y_i}^{y_o} y\widetilde{\varepsilon}_{en}c_w\,dy. \tag{B.7}$$

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

$$C_{\ell,\delta_a} = -\frac{\widetilde{C}_{Lw,\alpha}}{S_w b_w}\widetilde{\varepsilon}_{en} c_w (y_o^2 - y_i^2). \tag{B.8}$$

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

$$M_a = \tfrac{1}{2}\rho V_{pw}^2 \left(\delta_a \widetilde{C}_{Lw,\alpha}\widetilde{\varepsilon}_{en} c_w \left(y_o^2 - y_i^2\right)\right), \tag{B.9}$$

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

$$\widetilde{C}_{Lw,\alpha} \approx 2\pi, \tag{B.10}$$

and from thin airfoil theory (see for instance Ref. [25]) the elevon efficiency is

$$\widetilde{\varepsilon}_{en} \approx 1 - \frac{\sigma_{en} - \sin(\sigma_{en})}{\pi} \tag{B.11}$$

with

$$\sigma_{en} = \cos^{-1}\left(2\frac{c_{en}}{c_w} - 1\right). \tag{B.12}$$

B.2.2 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

$$C_{z,\delta_e} = \frac{2\widetilde{C}_{Lw,\alpha}}{S_w}\int_{y_i}^{y_o} \widetilde{\varepsilon}_{en} c_w \, dy. \tag{B.13}$$

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

$$C_{z,\delta_e} = \frac{2\widetilde{C}_{Lw,\alpha}}{S_w}\widetilde{\varepsilon}_{en} c_w (y_o - y_i) \tag{B.14}$$

or

$$F_e = \rho V_{pw}^2 \left(\delta_e \widetilde{C}_{Lw,\alpha} \widetilde{\varepsilon}_{en}\right) c_w (y_o - y_i) \tag{B.15}$$

with the wing lift slope and elevon efficiency defined in equations (B.10) and(B.19), respectively.

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

$$M_e = \rho V_{pw}^2 \widetilde{C}_{m,\delta e} \delta_e c_w (y_o - y_i) \tag{B.16}$$

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

$$\widetilde{C}_{m,\delta e} = \frac{\sin(2\sigma_{en}) - 2\sin(\sigma_{en})}{4} \tag{B.17}$$

with σ_{en} defined in Equation (B.12).

B.2.3 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 B.1, we find the force produced by the rudders is

$$F_r = \tfrac{1}{2}\rho V_{pw}^2 \left(\delta_r \widetilde{C}_{Lt,\alpha} \varepsilon_r\right) c_t d_s. \tag{B.18}$$

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

$$\widetilde{\varepsilon}_r \approx 1 - \frac{\sigma_r - \sin(\sigma_r)}{\pi} \tag{B.19}$$

with

$$\sigma_r = \cos^{-1}\left(2\frac{c_r}{c_t} - 1\right). \tag{B.20}$$

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

$$M_r = -\tfrac{1}{2}\rho V_{pw}^2 \widetilde{C}_{m,\delta r}\delta_r d_s c_t, \tag{B.21}$$

where

$$\widetilde{C}_{m,\delta r} = \frac{\sin(2\sigma_r) - 2\sin(\sigma_r)}{4}. \tag{B.22}$$

B.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 B.2.

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

$$\begin{aligned} F_{x-lev} = \frac{1}{2}\rho V^2 S_w \Big(& -\cos\alpha C_{D0} - \sin\alpha C_{L0} - \cos\alpha C_{D,\alpha}|\alpha| - \sin\alpha C_{L,\alpha}\alpha \\ & -\cos\alpha C_{D,q}\frac{c_w}{V}|q| - \sin\alpha C_{L,q}\frac{c_w}{V}q - \cos\alpha C_{D,\delta_e}|\delta_e| - \sin\alpha C_{L,\delta_e}\delta_e\Big), \end{aligned} \tag{B.23}$$

$$F_{y-lev} = \frac{1}{2}\rho V^2 S_w \left(C_{Y0} + C_{Y,\beta}\beta + C_{Y,p}\frac{b_w}{2V}p + C_{Y,r}\frac{b_w}{2V}r + C_{Y,\delta_a}\delta_a + C_{Y,\delta_r}\delta_r \right), \tag{B.24}$$

$$\begin{aligned} F_{z-lev} = \frac{1}{2}\rho V^2 S_w \Big(& -\sin\alpha C_{D0} - \cos\alpha C_{L0} - \sin\alpha C_{D,\alpha}|\alpha| - \cos\alpha C_{L,\alpha}\alpha \\ & -\sin\alpha C_{D,q}\frac{c_w}{V}|q| - \cos\alpha C_{L,q}\frac{c_w}{V}q - \sin\alpha C_{D,\delta_e}|\delta_e| - \cos\alpha C_{L,\delta_e}\delta_e\Big), \end{aligned} \tag{B.25}$$

Table B.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

and the torques similarly as

$$l_{lev} = \frac{1}{2}\rho V^2 S_w \frac{b_w}{2}\left(C_{l0} + C_{l,\beta}\beta + C_{l,p}\frac{b_w}{2V}p + C_{l,r}\frac{b_w}{2V}r + C_{l,\delta_a}\delta_a + C_{l,\delta_r}\delta_r\right), \tag{B.26}$$

$$m_{lev} = \frac{1}{2}\rho V^2 S_w c_w \left(C_{m0} + C_{m,\alpha}\alpha + C_{m,q}\frac{c_w}{V}q + C_{m,\delta_e}\delta_e\right), \tag{B.27}$$

$$n_{lev} = \frac{1}{2}\rho V^2 S_w \frac{b_w}{2}\left(C_{n0} + C_{n,\beta}\beta + C_{n,p}\frac{b_w}{2V}p + C_{n,r}\frac{b_w}{2V}r + C_{n,\delta_a}\delta_a + C_{n,\delta_r}\delta_r\right). \tag{B.28}$$

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

$$-\alpha_{stall} \leq \alpha \geq \alpha_{stall}. \tag{B.29}$$

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

$$F_{x-lev} = \frac{1}{2}\rho V^2 S_w \Big(-\cos\alpha C_{D0} - \cos\alpha C_{D,\alpha}|\alpha| \Big), \tag{B.30}$$

$$F_{y-lev} = \frac{1}{2}\rho V^2 S_w \Big(C_{Y0} + C_{Y,\beta}\beta \Big), \tag{B.31}$$

$$F_{z-lev} = \frac{1}{2}\rho V^2 S_w \Big(-\sin\alpha C_{D0} - \sin\alpha C_{D,\alpha}|\alpha| \Big), \tag{B.32}$$

and the torque model is simply

$$l_{lev} = 0.0, \tag{B.33}$$

$$m_{lev} = 0.0, \tag{B.34}$$

$$n_{lev} = 0.0. \tag{B.35}$$

B.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

$$\dot{u} = rv - qw + g(2\eta_1\eta_3 - 2\eta_4\eta_2) + \frac{F_x}{\mu}, \tag{B.36}$$

$$\dot{v} = -ru + pw + g(2\eta_2\eta_3 + 2\eta_4\eta_1) + \frac{F_y}{\mu}, \tag{B.37}$$

$$\dot{w} = qu - pv + g(\eta_4\eta_4 - \eta_1\eta_1 - \eta_2\eta_2 + \eta_3\eta_3) + \frac{F_z}{\mu}, \tag{B.38}$$

and

$$\begin{aligned}\dot{p} = {} & \frac{J_{xz}(J_{xx} - J_{yy} + J_{zz})}{J_{xx}J_{zz} - J_{xz}^2}pq - \frac{J_{zz}(J_{zz} - J_{yy}) + J_{xz}^2}{J_{xx}J_{zz} - J_{xz}^2}qr \\ & + \frac{J_{zz}}{J_{xx}J_{zz} - J_{xz}^2}l + \frac{J_{xz}}{J_{xx}J_{zz} - J_{xz}^2}n,\end{aligned} \tag{B.39}$$

$$\dot{q} = \frac{J_{zz} - J_{xx}}{J_{yy}}pr - \frac{J_{xz}}{J_{yy}}(p^2 - r^2) + \frac{1}{J_{yy}}m, \tag{B.40}$$

$$\begin{aligned}\dot{r} = {} & \frac{J_{zz}(J_{zz} - J_{yy}) + J_{xz}^2}{J_{xx}J_{zz} - J_{xz}^2}pq - \frac{J_{xz}(J_{xx} - J_{yy} + J_{zz})}{J_{xx}J_{zz} - J_{xz}^2}qr \\ & + \frac{J_{xz}}{J_{xx}J_{zz} - J_{xz}^2}l + \frac{J_{zz}}{J_{xx}J_{zz} - J_{xz}^2}n.\end{aligned} \tag{B.41}$$

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

$$\dot{\eta}_1 = \frac{1}{2}(\eta_4 p - \eta_3 q + \eta_2 r), \tag{B.42}$$

$$\dot{\eta}_2 = \frac{1}{2}(\eta_3 p + \eta_4 q - \eta_1 r), \tag{B.43}$$

$$\dot{\eta}_3 = \frac{1}{2}(-\eta_2 p + \eta_1 q + \eta_4 r), \tag{B.44}$$

$$\dot{\eta}_4 = \frac{1}{2}(-\eta_1 p - \eta_2 q - \eta_3 r), \tag{B.45}$$

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

$$\dot{x} = u(\eta_4\eta_4 + \eta_1\eta_1 - \eta_2\eta_2 - \eta_3\eta_3) + v(2\eta_2\eta_1 - 2\eta_4\eta_3) + w(2\eta_3\eta_1 + 2\eta_4\eta_2), \tag{B.46}$$

$$\dot{y} = u(2\eta_1\eta_2 + 2\eta_4\eta_3) + v(\eta_4\eta_4 - \eta_1\eta_1 + \eta_2\eta_2 - \eta_3\eta_3) + w(2\eta_3\eta_2 - 2\eta_4\eta_1), \tag{B.47}$$

$$\dot{z} = u(2\eta_1\eta_3 - 2\eta_4\eta_2) + v(2\eta_2\eta_3 + 2\eta_4\eta_1) + w(\eta_4\eta_4 - \eta_1\eta_1 - \eta_2\eta_2 + \eta_3\eta_3). \tag{B.48}$$