



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**INTEGRATION OF CONTROL ALGORITHMS FOR
QUADROTOR UAV'S USING AN INDOOR SENSOR
ENVIRONMENT**

by

Bryan D. Watts

September 2011

Thesis Advisor:

Isaac Kaminer

Second Reader:

Vladimir Dobrokhodov

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 2011	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE Integration of Control Algorithms for Quadrotor UAV's Using an Indoor Sensor Environment			5. FUNDING NUMBERS	
6. AUTHOR(S) Bryan D. Watts				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number n/a.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) This thesis develops an architecture that facilitates the design and indoor testing of control algorithms implemented onboard quadrotor UAV's using an ultra-wideband (UWB) indoor positioning solution from Ubisense. Initially, details are provided on basic quadrotor dynamics, the setup of the indoor sensor environment, and the communication scheme. A thorough analysis is conducted on the accuracy and estimation lag of Ubisense UWB sensors for providing indoor position information to the quadrotor. Once this framework is established, the focus is placed on design and experimental validation of the altitude hold control algorithm. The observer used is a discrete Kalman filter that minimizes the covariance of position and acceleration measurement inputs to produce a smooth estimation of states (position, velocity and acceleration). These estimated states are then fed into a modified PD plus Integral controller to produce quadrotor thrust commands for given altitude step commands. Results indicate that the technology used is capable of maintaining a UAV's altitude within an error margin of +/-13.3 cm, but the relatively slow update rate of the Ubisense system limits the possibility of more complex and aggressive maneuvers.				
14. SUBJECT TERMS UAV, Quadrotor, Autonomous, Control, Ultra-Wideband Radio Frequency Identification, Kalman Filter, Altitude Hold, Ubisense			15. NUMBER OF PAGES 65	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**INTEGRATION OF CONTROL ALGORITHMS FOR QUADROTOR UAV'S
USING AN INDOOR SENSOR ENVIRONMENT**

Bryan D. Watts
Lieutenant, United States Coast Guard
B.S., U.S. Coast Guard Academy, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
September 2011**

Author: Bryan D. Watts

Approved by: Isaac I. Kaminer

Vladimir N. Dobrokhodov

Knox T. Millsaps
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis develops an architecture that facilitates the design and indoor testing of control algorithms implemented onboard quadrotor UAV's using an ultra-wideband (UWB) indoor positioning solution from Ubisense. Initially, details are provided on basic quadrotor dynamics, the setup of the indoor sensor environment, and the communication scheme. A thorough analysis is conducted on the accuracy and estimation lag of Ubisense UWB sensors for providing indoor position information to the quadrotor.

Once this framework is established, the focus is placed on design and experimental validation of the altitude hold control algorithm. The observer used is a discrete Kalman filter that minimizes the covariance of position and acceleration measurement inputs to produce a smooth estimation of states (position, velocity and acceleration). These estimated states are then fed into a modified P-D plus Integral controller to produce quadrotor thrust commands for given altitude step commands. Results indicate that the technology used is capable of maintaining a UAV's altitude within an error margin of ± 13.3 cm, but the relatively slow update rate of the Ubisense system limits the possibility of more complex and aggressive maneuvers.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	QUADROTOR DYNAMICS	1
A.	INTRODUCTION.....	1
B.	QUADROTOR DESIGN.....	2
C.	DYNAMICS.....	2
II.	IMPLEMENTATION OF THE SENSOR ENVIRONMENT	7
A.	INTRODUCTION.....	7
B.	ULTRA-WIDEBAND RADIO FREQUENCY IDENTIFICATION	7
C.	UBISENSE INTRODUCTION.....	8
1.	Tags	8
2.	Sensors	9
3.	Software	10
D.	UBISENSE CALIBRATION	11
E.	UBISENSE ACCURACY.....	13
F.	UBISENSE SOFTWARE FILTERS.....	15
G.	SPEED OF RESPONSE	17
III.	EXPERIMENTATION SETUP	19
A.	COMMUNICATION SCHEME	19
1.	Ubisense Data	20
2.	MOOS	20
B.	FLIGHT TESTING APPARATUS	21
IV.	NAVIGATION AND CONTROL	23
A.	INTRODUCTION.....	23
B.	DISCRETE KALMAN FILTER.....	23
C.	DISCRETE KALMAN FILTER RESULTS.....	26
D.	CONTROLLER DESIGN.....	28
E.	P-D CONTROLLER ANALYSIS	28
F.	PID CONTROLLER	31
V.	FLIGHT TESTING	35
A.	INTRODUCTION.....	35
B.	CONTROLLER TUNING	35
C.	FLIGHT TEST RESULTS.....	36
D.	CONCLUSIONS	39
VI.	LIMITATIONS AND FUTURE WORK.....	41
A.	PROJECT LIMITATIONS	41
B.	PROPOSED TECHNOLOGY IMPLEMENTATION	41
C.	FUTURE WORK.....	43
	LIST OF REFERENCES.....	45
	INITIAL DISTRIBUTION LIST	47

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1.	AscTec Hummingbird Quadrotor (After [1])	1
Figure 2.	Quadrotor Orientation (From [1]).....	3
Figure 3.	Quadrotor Accelerations (After [4])	4
Figure 4.	Ubisense Sensor/Tag Measurement Diagram (From [6]).....	10
Figure 5.	Ubisense Location Engine Configuration Display	11
Figure 6.	Unfiltered Ubisense X-Position	14
Figure 7.	Unfiltered Ubisense Z-Position.....	15
Figure 8.	Static Information Filtering Normalized Position.....	16
Figure 9.	Information Filtering Normalized Position.....	17
Figure 10.	Ubisense Speed of Response for Step Input	18
Figure 11.	Communication Scheme	19
Figure 12.	Quadrotor Z-axis Isolation Frame.....	22
Figure 13.	Discrete Kalman Filter Implementation (From [10]).....	24
Figure 14.	DKF Estimated Altitude Gains	26
Figure 15.	Unfiltered Signal vs. DKF Signal	27
Figure 16.	Unfiltered Signal vs. DKF Signal (Larger Scale)	27
Figure 17.	Controller Design Flowchart.....	28
Figure 18.	PD Closed Loop System Root Locus & Bode Plot.....	30
Figure 19.	PD Controller Block Diagram	30
Figure 20.	PID Compensators Using High and Low Integral Gains.....	32
Figure 21.	Quadrotor Hovering Autonomously	35
Figure 22.	UAV Altitude from First Successful Flight Test	37
Figure 23.	Second Test Flight	37
Figure 24.	Third Test Flight	38
Figure 25.	MaxBotix Ultrasonic Range Finder (After [13])	42

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	Effect on System Characteristics by Increasing Gains	36
Table 2.	Transient and Steady State Response Characteristics.....	39

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AoA – Angle of Arrival

CF – Complementary Filter

DHCP – Dynamic Host Configuration Protocol

DKF – Discrete Kalman Filter

IMU – Inertial Measurement Unit

LOS – Line of Sight

MOOS – Mission Oriented Operating Suite

PoE – Power over Ethernet

RFID – Radio Frequency Identification

TDoA – Time Difference of Arrival

UAV – Unmanned Aerial Vehicle

UDP – User Datagram Protocol

UWB – Ultra-Wide Band

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I am grateful for the dedication of Professor Kaminer and Professor Dobrokhodov throughout my education and research. I am fully aware of the patience that was required to impart control theory to a student such as myself. Thanks are also in order for Jefferey Wurz and Venanzio Cichella for their continuous assistance and willingness to fill in my gaps of knowledge. Finally, to my gorgeous wife, Maggie, thank you for your loving support and continuous study snacks.

THIS PAGE INTENTIONALLY LEFT BLANK

I. QUADROTOR DYNAMICS

A. INTRODUCTION

A quadrotor is a type of helicopter with several changes incorporated into the design to create an ideal platform for conducting agile, autonomous maneuvers. Currently, quadrotor UAVs (Unmanned Aerial Vehicles) are a popular vessel for testing new control algorithms for several reasons: they are lightweight, small in size, inexpensive, and highly maneuverable. Aside from testing, they also have many valuable real-world applications such as surveillance, search and rescue, and payload transport. The Ascending Technologies Hummingbird is the specific quadrotor model used throughout my experimentation; see Figure 1. It is uniquely manufactured mainly for the purposes of classroom instruction and research. This is precisely the application that is researched in this paper. The objective is to design a robust indoor testing environment for the quadrotor that can be utilized to fly a successful altitude hold algorithm. Once this is completed, the same architecture may be used for further research. In the next section, a description of the basic construction and dynamics of the Hummingbird quadrotor will be provided.



Figure 1. AscTec Hummingbird Quadrotor (After [1])

B. QUADROTOR DESIGN

The Hummingbird model (along with many other standard quadrotor UAV's) consists of four independent propellers fixed to four corners of the vehicle. Each propeller is powered by its own DC brushless motor and connected by a cross configuration with a central mounting platform [1]. This frame is fabricated from a light-weight, carbon-fiber and balsa wood sandwich material. The central platform holds the Lithium-polymer rechargeable battery, Inertial Measurement Unit (IMU), stabilization control circuitry, and multi-function power board. The IMU for the Hummingbird is known as the AscTec AutoPilot. In addition to self-stabilization circuitry, the autopilot also contains three accelerometers and three gyros which measure all the vehicle's accelerations within the body frame of reference. The accelerometers determine the inertial acceleration in each axis of movement, and the gyros provide the angular accelerations of every Euler angle. The power board is of an ideal design since it is capable of providing differing voltage values to the autopilot, DC motors, and any additional components installed by the user.

Additionally, each quadrotor comes with an essential remote control transmitter. These devices are necessary even when only using the autonomous flight mode. Ascending Technologies has established that the quadrotor will be unable to fly without an activated transmitter for safety reasons. This requirement is a safety measure that provides the opportunity for a user to resume manual control of the quadrotor in the event that its continued autonomous flight could cause injury to the vehicle or others around it. Therefore, it is prudent for a new UAV pilot to first practice the manual controls of the quadrotor before attempting autonomous flight, especially indoors.

C. DYNAMICS

Unlike a normal helicopter, the quadrotor does not have variable-pitch rotors to adjust levels of thrust. Instead, it maneuvers by alternating rotational speeds of each rotor. These rotational speeds translate to individual rotor thrusts. The fact that every

motor is controlled by a separate speed controller circuit is essential to the maneuverability of the quadrotor. The front and rear rotors both rotate in a counter-clockwise direction at all times, and the left and right rotors rotate in a clockwise direction (Figure 2). When the rear rotor rotates at a different speed than the forward rotor, this creates a moment, and consequently a pitch angle, that moves the quadrotor forward (or backward) [2]. Similarly, when the right rotor rotates at a different speed than the left rotor, this creates a roll angle that results in lateral movement. Yaw rotation is produced by any inequality in the sum of the moments produced by each motor. For example, if the left and right motors are collectively producing greater thrust than the forward and rear motors, then the quadrotor will rotate in a clockwise direction. However, the majority of the turning in continuous flight can be executed by rolling simultaneously while pitching. Finally, upward or downward movement is experienced by the total thrust created by all four rotors. If the sum of the individual thrusts is greater than the weight of the vehicle (hovering thrust), then it will certainly ascend. Likewise, a total thrust that is less than the hovering thrust will result in descending altitude. Figure 3 illustrates the accelerations achieved from each of these rotor speed manipulations.

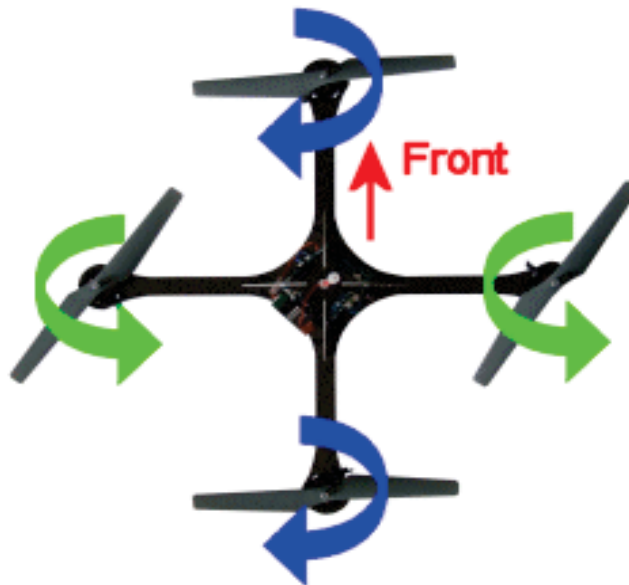


Figure 2. Quadrotor Orientation (From [1])

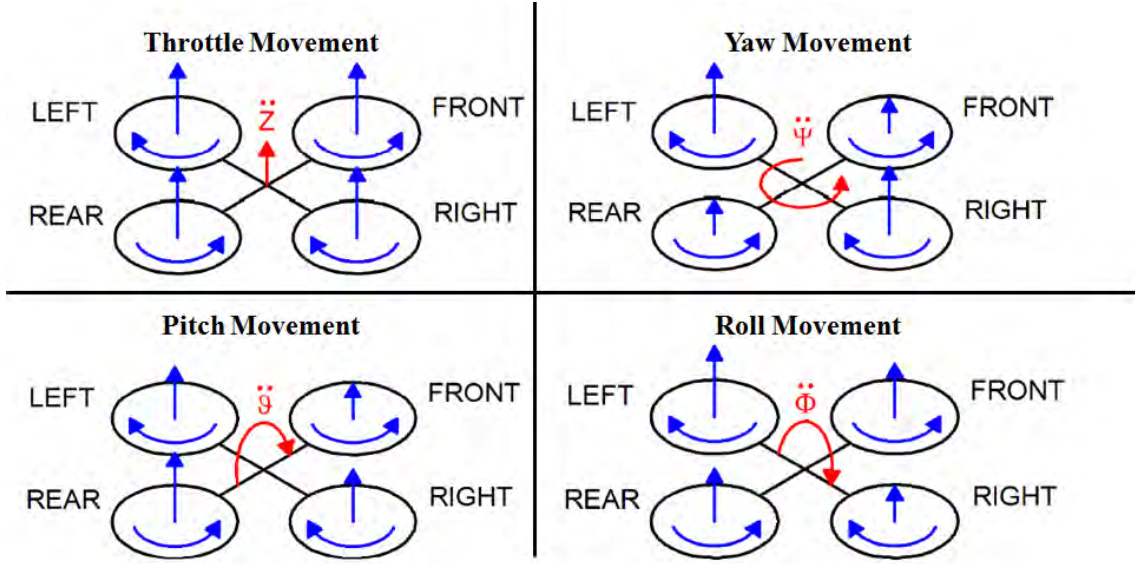


Figure 3. Quadrotor Accelerations (After [4])

Since the focus of experimentation in this paper is limited to an altitude hold control in the vertical channel, the previously mentioned total thrust is the main consideration. (Full descriptions of quadrotor dynamics, including all 6 degrees of freedom, can be found in [2], [3] or [4]). In order to create motion in the z direction only, it is also necessary to maintain the thrust of each individual rotor at the same value. As described above, any differences in propeller speeds result in an unwanted Euler angle. In Chapter III, I will describe the setup used to ensure isolation of the quadrotor in the vertical channel. For the controller, it is important to note that the output provided, as the input to the physical system, will only be total thrust. This total thrust is the sum of the individual thrusts:

$$F = f_1 + f_2 + f_3 + f_4 \quad (1.1)$$

Additionally, this total thrust is found in the only pertinent equation of motion modeled in our system:

$$m\ddot{z} = F \cos \theta \cos \varphi - mg \quad (1.2)$$

Equation 1.2 comes directly from Newton's Second Law of motion where the sum of the forces includes the upward thrust of the quadrotor and the downward force due to gravity, which is assumed constant for our purposes. Additionally, we assume the effects of drag are negligible on the quadrotor. The thrust from the quadrotor, $F \cos \theta \cos \varphi$, includes the total thrust of the rotors and the cosines of the pitch and roll angles respectively. In order to simplify (1.2) into a linear equation, it is possible to implement the following substitution for F:

$$F = \frac{r_1 + mg}{\cos \varphi \cos \theta} \quad (1.3)$$

In this expression, r_1 represents the input of the physical model (thrust). When (1.3) is substituted into the original equation of motion (1.2), gravitational force and the cosine terms are cancelled out. This results in a linear equation:

$$m\ddot{z} = r_1 \quad (1.4)$$

Next, by taking the Laplace Transform of (1.4), an equation in the frequency domain is achieved:

$$ms^2 Z(s) = R_1(s) \quad (1.5)$$

Finally, this can be simplified by assuming a mass of the quadrotor of 1 kilogram. This is an acceptable simplification since I will later account for the true mass of the quadrotor by experimentally choosing a nominal thrust value that keeps the UAV hovering. This results in the following transfer function for the physical system:

$$\frac{Z(s)}{R_1(s)} = \frac{1}{s^2} \quad (1.6)$$

Clearly, this model has been simplified to a mere double integrator with characteristics that will be further investigated in Chapter IV.

II. IMPLEMENTATION OF THE SENSOR ENVIRONMENT

A. INTRODUCTION

The initial step necessary for flying the quadrotor autonomously indoors is setting up a sensor environment that is capable of providing accurate and fast navigation data for the position of the vehicle. This task begins with the installation of robust sensors (Ubisense) throughout the indoor environment. Once the sensors are calibrated effectively and can transmit position data, it will be possible to correlate this data with z-acceleration telemetry from the quadrotor's IMU to provide required input states to the control model. Receiving the best measurements possible from these sensors is crucial for the success of indoor UAV flight.

B. ULTRA-WIDEBAND RADIO FREQUENCY IDENTIFICATION

Ubisense achieves its position determination abilities through use of Ultra-Wideband (UWB) Radio Frequency Identification (RFID). This technology is highly effective over a multitude of applications due to its low manufacturing cost, high localization accuracy within short to medium ranges, high time resolution, and the safety provided by its low power transmission [5]. Another benefit of using any RF localization technology is its ability to penetrate through obstacles and propagate across long distances. In contrast, an optical localization system, such as Vicon, has a different set of pros and cons. It is limited by its need for line-of-sight (LOS) conditions and its sensitivity to sunlight, but it provides a higher degree of accuracy than most RF systems. Consequently, optical systems are ideal for indoor, close-range environments such as the lab where we conducted the flight tests for this paper. Unfortunately, optical localization technology is significantly more expensive and was unattainable during the time of my testing. Therefore, UWB RFID is the next best option, and the following sections will investigate the calibration and performance of the Ubisense system in particular.

C. UBISENSE INTRODUCTION

Ubisense is a company that developed an Ultra-Wideband RFID system that boasts an accuracy of 15 cm for real time positions [6]. Ubisense has the only UWB localization system that is certified in both the United States and Europe. The majority of its clients use the technology for keeping track of inventory and/or personnel. Only a small percentage has purchased the system to track UAV's, but its equipment does have the capability for this application. The following sections describe the individual components of the Ubisense system.

1. Tags

The research bundle purchased by the Naval Postgraduate School includes four sensors, multiple tracking tags, and software for coordinating all information. The available tags consist of both "compact tags" and "slim tags." Both of these varieties have similar capabilities, but the slim tag is longer and includes buttons to mark location events in the software. Since the slim tags are slightly larger, and since there was no immediate use for marking location events, the compact tags were the only ones used in this experimentation. Additionally, Ubisense specifies that the compact tags are more efficient for mounting on top of objects whereas slim tags function better on the sides of objects. Each compact tag is approximately 4 by 4 cm in dimension, and is powered by a single 3V coin cell battery. These tags are able to go into sleep mode to conserve power and are reenergized when motion is detected. The compact tag emits a UWB signal between 6 and 8 Gigahertz to be received by the sensors. It also is able to send its telemetry data over a 2.4 GHz channel. This data includes its battery life status and provides the opportunity to command new update rates dynamically. The tag's maximum update rate is 33.75 Hz, but the user is only able to receive one position fix for every four tag updates. Therefore, the functional update rate is approximately 10 Hz. Slower update rates can be commanded, but for the purposes of tracking a rapidly moving UAV, the fastest available is used at all times. In the quadrotor experimentation, a single tag is attached to the top of the quadrotor, via Velcro, to ensure optimal exposure to each of the sensors.

2. Sensors

Every one of the four Ubisense sensors includes an array of antennas and UWB receivers. This allows two-way communications between the sensors and the tags. Consequently, the sensors can send and receive telemetry information from the tags, and also receive tag position measurements over UWB signals. These sensors are intended for fixed installation in a square or rectangular configuration to maximize coverage. Mounting instructions specify that they should be fixed to a solid structure at a height that allows each sensor to face the floor at the center of the room using mild angles of pitch. Additionally, all of the sensors receive power and send information via Power-over-Ethernet (PoE) cables. They also must be connected to each other using separate network cables to correlate timing between the slaves and the designated master. Ubisense is able to provide time stamps with accuracy better than 10 microseconds. The designated master sensor serves a dual role; it keeps accurate time for each position fix and also collects the measurement data from each of the slaves to then calculate the position. Any of the sensors is able to serve as a master or a slave.

The sensors all take UWB measurements of the tag through use of both Angle-of-Arrival (AoA) and Time-Difference-of-Arrival (TDoA) methods (as portrayed in Figure 4). The AoA method measures both the azimuth and elevation angles of the incoming UWB pulse from an energized tag and uses simple trigonometry to find a fix. The TDoA method is where the time difference between pulses received from any two sensors is correlated to help plot a fix. A fix can be achieved from the combination of two pieces of information (two AoA's or one TDoA and one AoA). Therefore, in theory, only two functional sensors are necessary to plot a fix. However, each additional sensor increases accuracy of a given fix since the master sensor can correlate all pieces of information to minimize error. Finally, it is important to note that four sensors make up a single "cell." If additional sensors are available, RFID can be extended to multiple spaces or rooms.

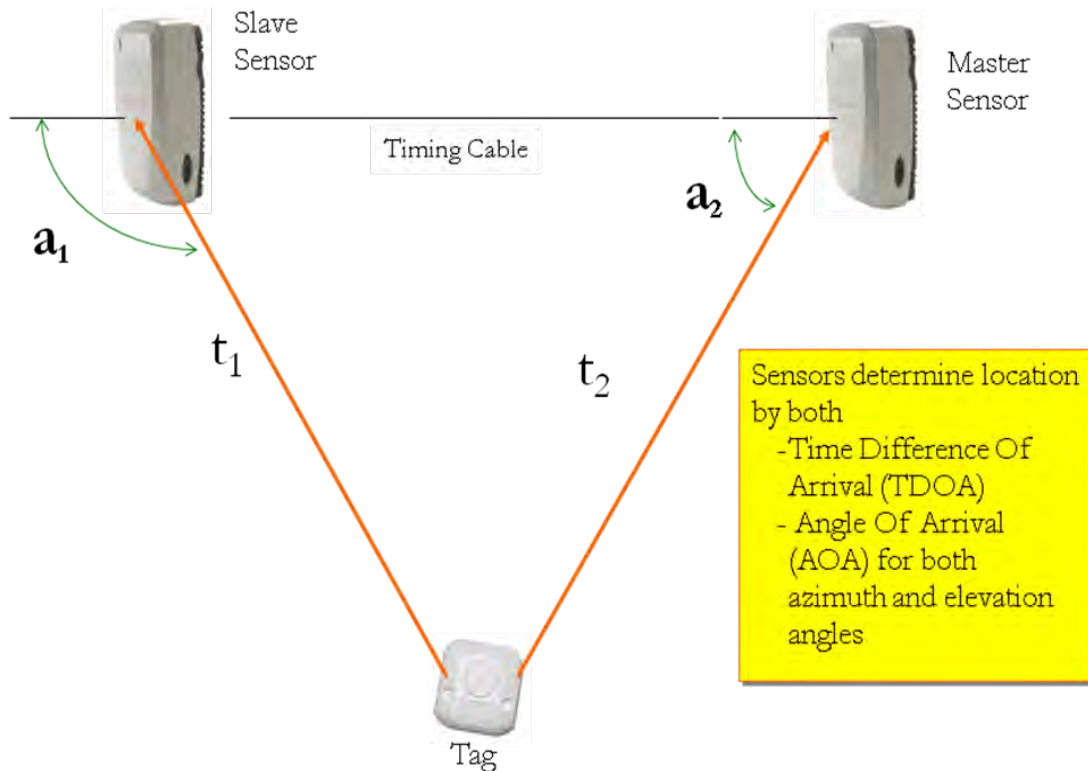


Figure 4. Ubisense Sensor/Tag Measurement Diagram (From [6])

3. Software

The PC laptop used to run the Ubisense Location Platform must first be configured to communicate via Dynamic Host Configuration Protocol (DHCP). This allows all the position data sent over Ethernet from the master sensor to be collected in the same network location. The main application provided by Ubisense is known as the “Location Engine Configuration.” This is the principle control center for initializing and monitoring the sensor environment. The Location Engine Configuration is where the locations and orientations of each of the sensors are designated, and where the user can define functionality of the sensors and tags. It is also where the user runs sensor calibrations and monitors (in 2-D or 3-D) the position of each registered tag.

Figure 5 shows a typical display of the sensor map with a single tag location fix plotted. The green lines coming from each of the four sensors represent the AoA for the tag shown in the middle of the room. Each of the crosshairs plotted in the image

represent a precisely measured position on the floor of the lab. These are used to calibrate the sensors, as will be discussed in the next section.

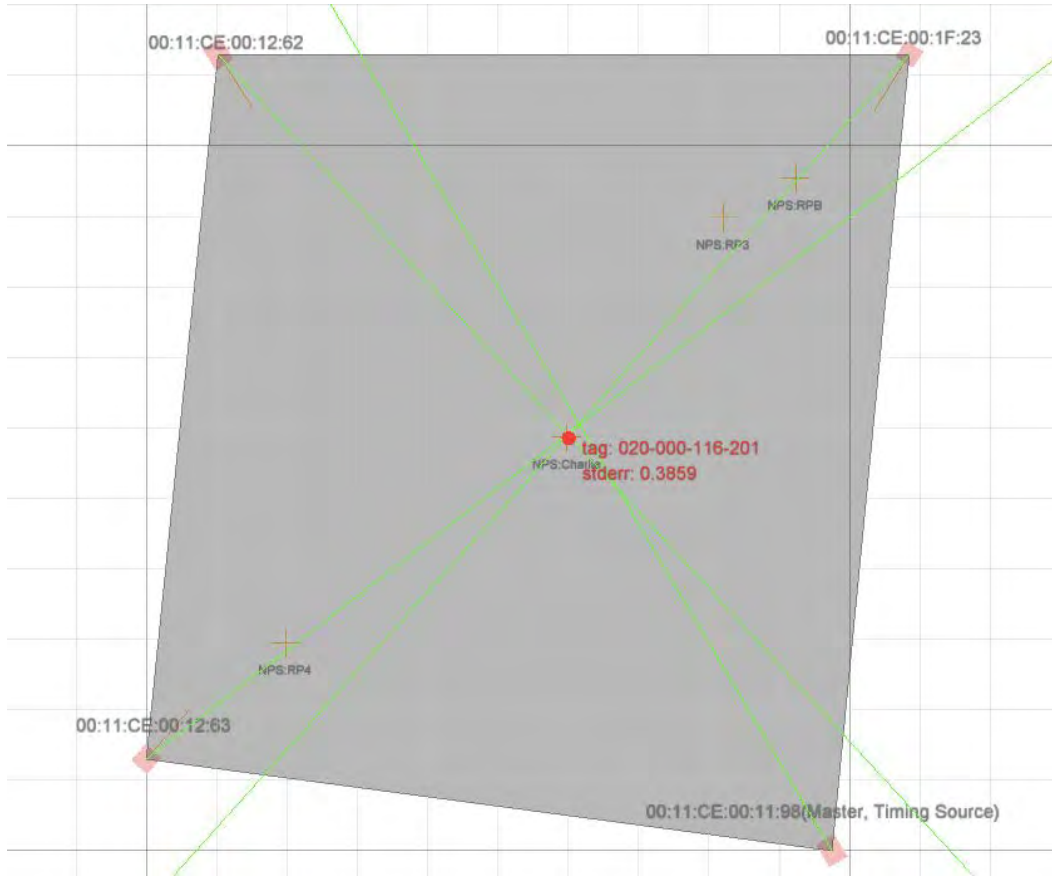


Figure 5. Ubisense Location Engine Configuration Display

D. UBISENSE CALIBRATION

In order to provide the best position information possible, it is necessary to run orientation and timing calibration processes for each installed sensor. Before this can be accomplished, the exact locations and approximate orientations of the sensors must be measured. The locations of the sensors were determined by referencing them from a chosen point of origin. In our lab, this origin was chosen as the point on the floor directly below one of the sensors (bottom left sensor shown in Figure 5). Then, using a laser

range-finder, each sensor was measured in meters (x, y, and z-coordinates) from the origin. These coordinates were then entered into the software, along with approximated elevation and azimuth angles. When the angles were measured, it was also necessary to ensure each sensor was level about its roll axis.

The final prerequisite to running the calibrations was to plot exact coordinates for designated locations on the floor of the room. Each of these locations were permanently marked and labeled directly onto the concrete floor of the lab. The main point used was located in the center of the area of coverage and was designated as “Charlie.” The orientation and cable calibrations are included in the Ubisense software. For the Orientation Calibration, all that is required is to place a single sensor in the known location, pick a sensor to calibrate, choose the sensor’s known position from a list of points entered earlier, and run the calibration [7]. The process will compare the approximated azimuth and elevation angles of the selected sensor to more accurate values after measuring multiple AoA’s from the sensor. It will then provide the user with the adjusted angles that can either be rejected or accepted for the system. The Cable Calibration can also be conducted for each individual sensor with the additional initialization requirement of choosing another sensor for timing comparison. For best results, each slave should have their timing calibrated against the master’s timing. This calibration results in a suggested timing offset, which can then be accepted to improve the accuracy of the system. Timing offset values can also be calculated using the Equidistant Calibration. This requires the tag to be measured in a location that is equidistant to two of the sensors. Since this setup is slightly more complicated and reaches the same goal, the Cable Calibration is sufficient.

Once the appropriate calibrations have been completed, and at least one tag has been registered and energized, the system should be ready to track a tag. By viewing the screen shown previously in Figure 5, the user should be able to clearly see a red dot in the location of the tag. Ideally, this dot will remain in one place if it is truly static. If the green lines representing the AoA from each sensor are not lining up with the location of the tag, then it may be necessary to recalibrate the sensor in error before proceeding. When the tag’s update rate is set to its “one every four time slots” designation (fastest

available), then the x, y, and z position coordinates should be displayed in the lower left of the screen at a rate of almost ten times per second. In the next section, a thorough analysis of the accuracy of Ubisense, with and without filters, is conducted.

E. UBISENSE ACCURACY

The Ubisense software includes several pre-programmed filters that can be used in different applications. This section investigates the accuracy of data using no filters. Further analysis with filters is included near the end of this chapter. Results were plotted from position data for each of the x, y, and z-coordinates, as well as a normalized position magnitude from the origin. Each recorded position is correlated against the assumed known position from the laser rangefinder. This data was plotted using MATLAB with the known position included as a constant value (data was collected by a local intern named Robert Salerno).

Included in Figure 6 is a sample of one of the plots used to measure the x-coordinate from Ubisense over a time of 150 seconds. The dotted line in the center is the known position, and the dotted lines bounding the plot represent standard deviation. This indicates that the maximum average error for unfiltered Ubisense x-positions is 0.13 meters. Since no filters have yet been implemented, this result shows that better accuracy than the Ubisense-rated 15 cm can be achieved (at least in the horizontal plane). However, this position was taken in the center of the lab's area of coverage where accuracy is maximized. The degree of error tends to increase exponentially as the tag is moved toward the borders of coverage. The vertical channel quadrotor flights were conducted mainly near this center point to maximize control effectiveness in the experimentation. The error documented in Figure 6 is nearly identical to the error of the y-position, but there is a significant difference with the z-position.

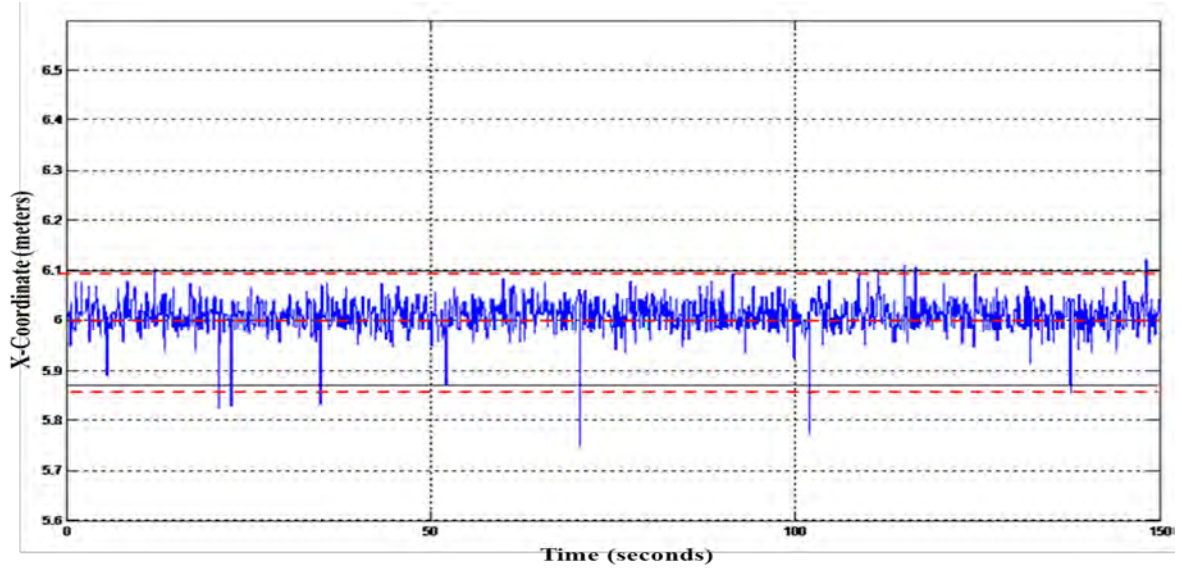


Figure 6. Unfiltered Ubisense X-Position

Next, it is necessary to investigate the system's performance in the vertical channel since this is initially the primary mode of interest. Unfortunately, the results in this scenario degrade significantly. Figure 7 shows this result for the same central position, but the tag has been placed on a wooden block 0.4 meters high to avoid the extra noise encountered near the concrete floor. Here the error is observed to be approximately 0.3 meters; more than twice the error in the x and y coordinates. This is actually an expected degradation in performance since indoor localization sensors encounter the same geometrical difficulties of GPS satellites. Satellites always experience increased error in altitude measurements when the satellites in view of the receiver are all close to the horizon. Since the Ubisense sensors have been installed only 4.5 meters high, they fall victim to the same phenomenon. This error could be mitigated by elevating the sensors, but space constraints limited this possibility. Additionally, the further away the sensors are from the tag, the less capable they are of receiving accurate UWB signals.

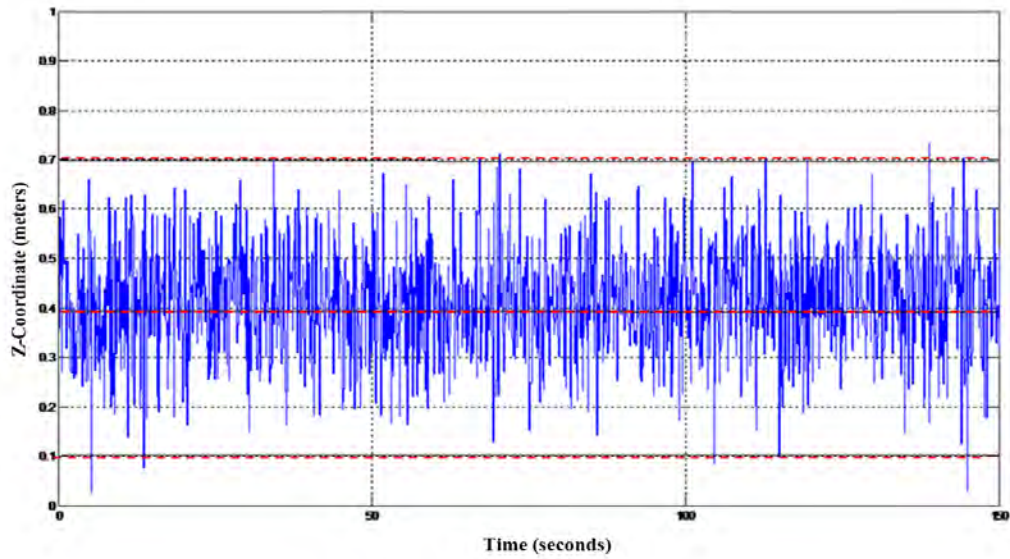


Figure 7. Unfiltered Ubisense Z-Position

The altitude hold control problem is a serious challenge for any indoor UAV flight testing. Researchers have encountered continuous setbacks in looking for a device that can be used to provide reliable altitude data. For outdoor flight, the significant error given by GPS has prompted companies like Ascending Technologies to install a barometric altimeter to use air pressure measurements to approximate a linear relationship with altitude. Unfortunately, these pressure readings are highly nonlinear in practice and do not provide the degree of precision required to fly in a relatively small indoor space. Therefore, attempting to take the Ubisense altitude data and process it into a usable format is a crucial first step in autonomous indoor flight. Without a reliable altitude hold control, more complex path-following algorithms in three dimensions would be nearly impossible.

F. UBISENSE SOFTWARE FILTERS

As mentioned previously, Ubisense software includes four, fully-developed filters that are designated for specific applications. These include Information Filtering, Fixed Height Information Filtering, Static Information Filtering, and Static Fixed Height Information Filtering. Clearly, the two filters that mention “fixed height” are not applicable to an altitude hold control because the fixed height designation makes the

assumption that the tag will only be moving horizontally. This completely contradicts our intended use. Therefore, I have investigated the effects of Information Filtering and Static Information Filtering only. As can be inferred, the Static Information Filtering models Gaussian Noise on position, whereas the Information Filtering models Gaussian Noise on velocity. Based on their described applications, one might assume that the static filtering should not be considered since it does not account for velocity. However, in an ideal scenario, if the altitude hold is working moderately well, the velocity is almost negligible anyway. Therefore, it is still beneficial to compare the two filters.

Similar tests to the ones conducted for the no-filter conditions were replicated with each of the provided filters. A normalized position, incorporating x, y and z positions, will now be observed to measure the direct distance from the designated origin to the known location. The performance of the Static Information Filtering can be seen in Figure 8.

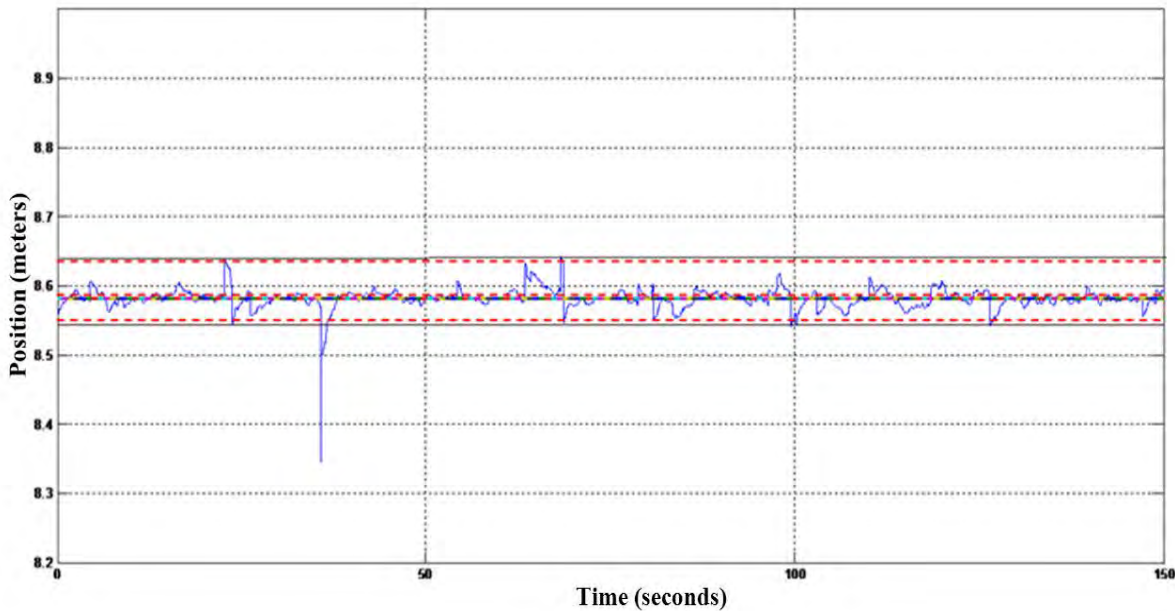


Figure 8. Static Information Filtering Normalized Position

Here, the error has been reduced to 0.06 meters, which is a significant improvement. There is still one wayward spike showing, but this noise will be further processed by the observer design in Chapter IV. Next, the performance of the Information Filtering at the same position can be seen in Figure 9.

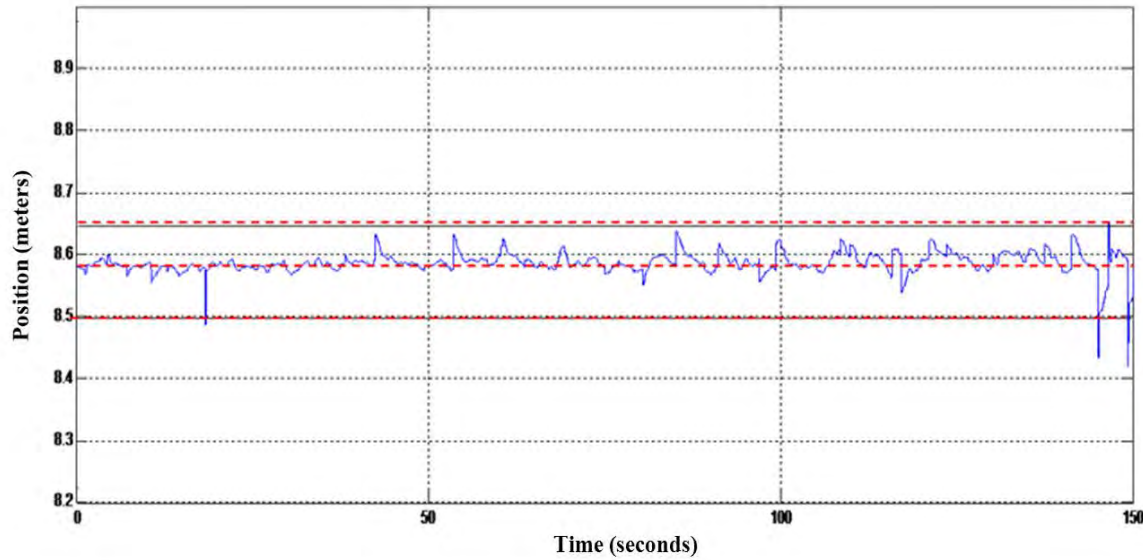


Figure 9. Information Filtering Normalized Position

This presents an error of 0.08 meters. The obvious flaw in this last test is that the tag being measured is static and has no velocity. Therefore, this is an ineffective display of how the Information Filtering would minimize velocity noise. Conducting such a test would require dynamically moving the tag in a pre-measured path. The difficulty involved with setting up such an experiment did not seem worth the benefits gained from observing a more realistic performance. Therefore, since the two filtering methods both produced small errors with relatively small difference, and based on the expectation of added filtering of velocity noise, it was decided to work with the Information Filtering alone.

G. SPEED OF RESPONSE

Finally, a rudimentary test was conducted to gain an approximation for the actual speed of response of Ubisense with a step input change in altitude. To investigate this

result, the Ubisense tag was fixed in such a way that it could only move in the z-direction. Then, while continuous measurements were plotted, the tag was swiftly raised to a height one meter above its original altitude. The gradual adjustment of Ubisense to this new altitude is displayed in Figure 10. A speed of response time constant of two seconds resulted from measuring the time elapsed for the Ubisense z-position to achieve 70% of the input. This lag presents the most significant difficulty encountered with the Ubisense system. The delay must be incorporated into the final model by ensuring that the controller (described in Chapter IV) has a speed of response that is not faster than this time constant. If the controller is faster, then the transient oscillation of Ubisense will be propagated throughout the model. This will result in a slower settling time for the system, but ultimately, this transient characteristic is less important than the steady state accuracy of the altitude hold.

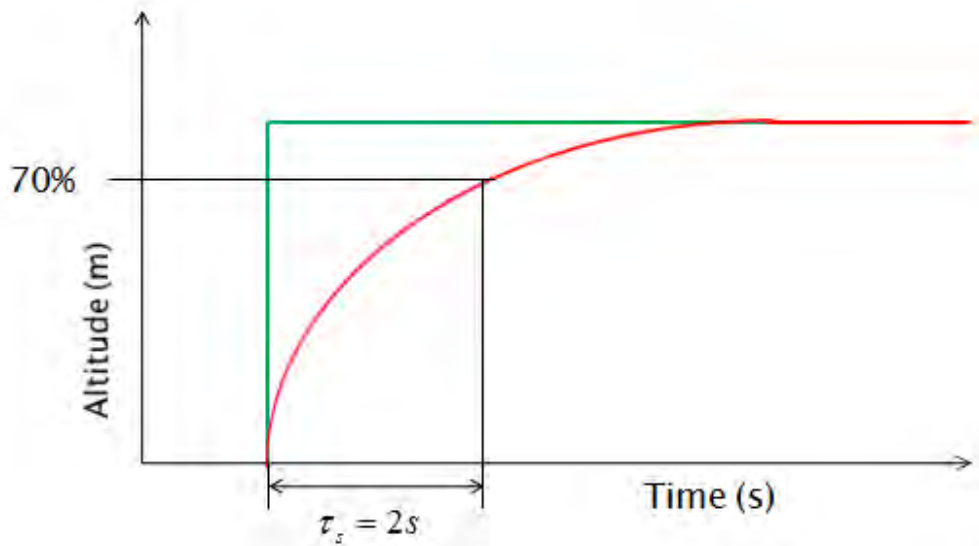


Figure 10. Ubisense Speed of Response for Step Input

III. EXPERIMENTATION SETUP

A. COMMUNICATION SCHEME

An effective flight of the quadrotor indoors depends upon the active communication between all measurement systems, commanded inputs and the control model. The architecture employed consists of Ubisense sensors, tags and software, the quadrotor IMU and speed controllers, a Simulink control model, and a Linux-based telemetry collecting program known as MOOS. This detailed network can be visualized in Figure 11, and will be described in the subsequent sections.

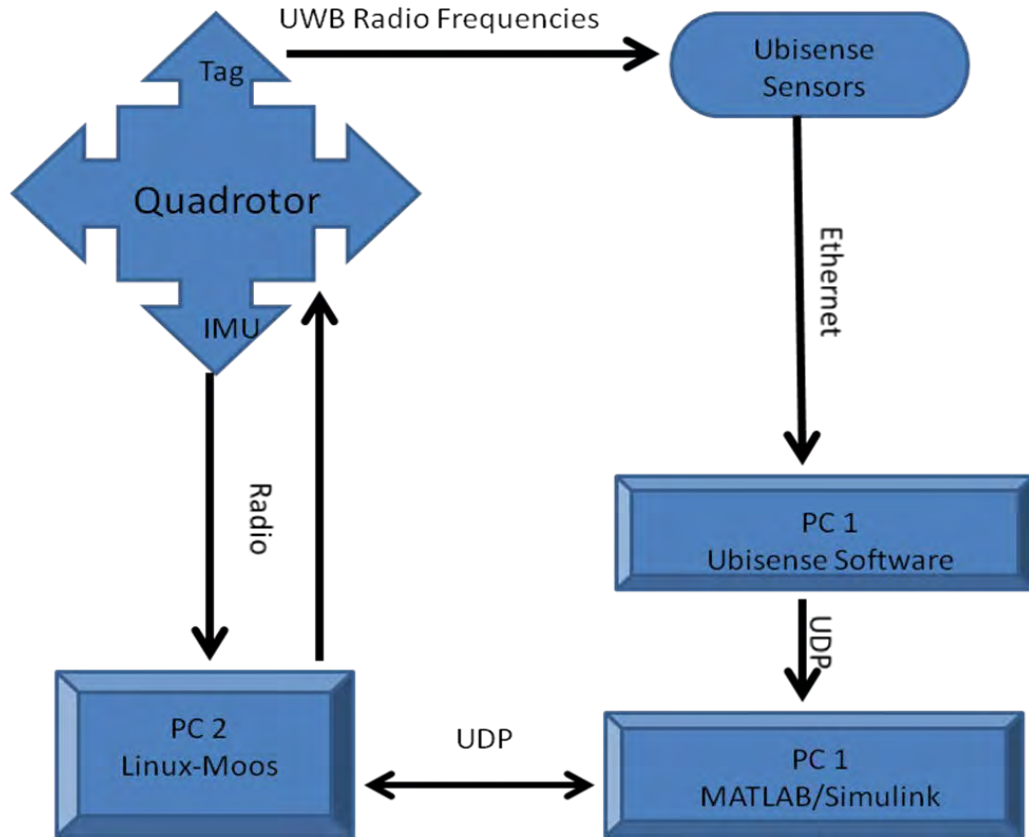


Figure 11. Communication Scheme

1. Ubisense Data

Descriptions of how position information is gathered from the tag, calculated by the sensors, and then sent to the first PC have already been presented in Chapter II. Therefore, the first new communication link of interest in Figure 11 is the transfer of position data from Ubisense software to a Simulink model. It seems as though it should be a simple task to deliver information over UDP (User Datagram Protocol) between two programs on the same computer, but this is not a direct process by any means. We requested a specific program from Ubisense, and then Jeffrey Wurz (Unmanned Systems Laboratory Technician) was able to modify a few lines of the C# code to allow data compilation into UDP. Once the data was available in UDP, it was simple to collect and unpack the information using block-sets within Simulink.

2. MOOS

MOOS, Mission Oriented Operating Suite, is a Linux-based program that runs on the second computer as a means of sending and receiving telemetry data and commands to and from the quadrotor. It is cross-platform software that is run in the programming language of C++ and is used mainly for research in robotics [8]. The MOOS software includes a main database, “MOOSDB,” which can compile and disperse information from specifically programmed applications.

Two applications for use with the quadrotor were created here at the Naval Postgraduate School by Research Associate Theodore Masek. The first, labeled “iAscTechQuadRotor,” registers to receive vehicle control messages and also publishes data from the quadrotor to MOOSDB using a ZigBee radio. The ZigBee Alliance manufactures a tiny radio that can plug into the USB port of a computer and broadcast and receive data from the quadrotor in the 2.4 GHz frequency band [9]. Among the values collected are height (via barometric altimeter); magnetic heading; angle of roll, pitch and yaw; velocity of roll, pitch and yaw; acceleration of roll, pitch and yaw; speed in x, y, and z; acceleration in x, y, and z; and multiple values of GPS information if GPS is activated (latitude, longitude, horizontal accuracy, vertical accuracy, etc.). Despite the availability of plentiful information, only the z acceleration is used for this

experimentation. The provided speed in the z direction is a value based only on the measured acceleration and is without units. It was found to be unreliable and is therefore unused in the observer or controller design. Additionally, the altitude data provided by the barometric altimeter was also found to be unreliable, for the same reasons discussed in Chapter II. Therefore, Ubisense was the sole method of providing dependable altitude data. This data is correlated with the z-accelerations continuously updating in MOOS to give an estimated altitude in the system observer (described in Chapter IV).

The name of the second created application is “pSimulinkBridge.” As the name suggests, it serves as a two-way communication bridge between MOOSDB and Simulink. It registers for vehicle information to forward via UDP to Simulink and also receives control commands to submit back to the database. These control commands are submitted in the form of percentages of maximum total thrust, pitch angle, roll angle and yaw angle before they are received by MOOSDB and sent back over the ZigBee radio to the quadrotor.

B. FLIGHT TESTING APPARATUS

In order to properly isolate flight in only the vertical channel, it was necessary to construct some means of restricting movement of the quadrotor in any horizontal plane. This was simply done by building a wooden frame, two meters high, with two tensioned steel cables running down the middle. The quadrotor in use was then fixed with two brackets allowing its movement up and down the length of the cables without movement in any other directions. This frame had the added bonus of portability so that it could be used in any position within the coverage of the Ubisense system. This allowed for execution of test flights in areas of optimal, as well as sub-optimal, coverage. A 30 cm high foam block was fixed underneath the quadrotor as a makeshift landing platform which protected against damage upon rough descents. A similar strip of foam was attached to the top of the frame to guard the propellers and motors upon rapid ascents.

Once complete, this frame allowed approximately 1.5 meters of unrestricted movement along the z-axis. This space was sufficient for adequately testing the altitude hold controls implemented in Chapter IV. In order to give the quadrotor enough room

above and below a commanded altitude, a standard height of 1.2 meters was used for a majority of the tests. This extra room provided a buffer for oscillations encountered in the transient before steady-state stability was achieved. A photo, along with a basic drawing of the frame described, is shown in Figure 12.

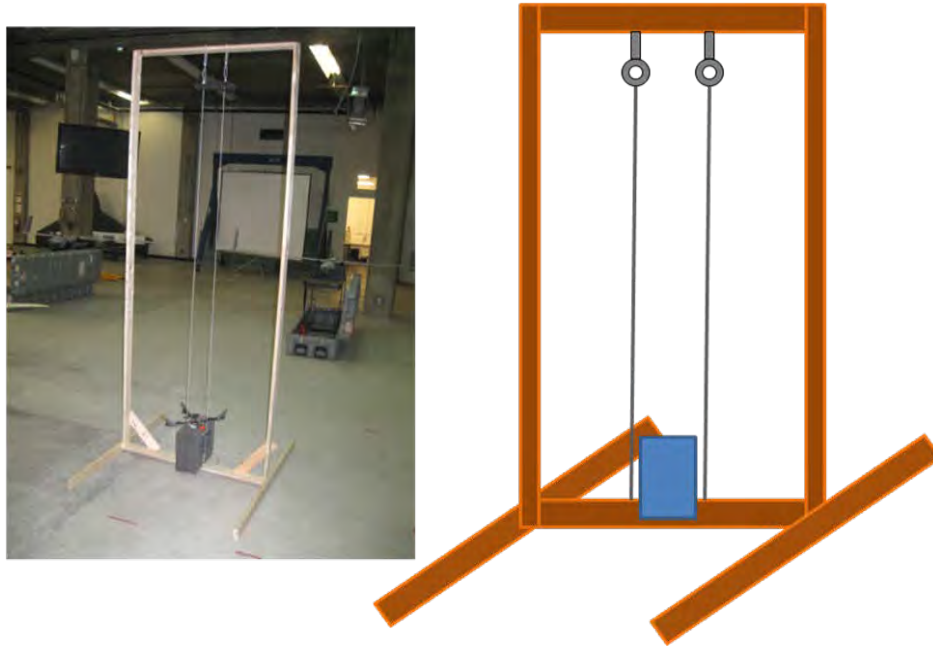


Figure 12. Quadrotor Z-axis Isolation Frame

IV. NAVIGATION AND CONTROL

A. INTRODUCTION

Receiving a noise-free and accurate feedback signal is essential to the closed-loop controls involved in UAV flight testing. It is apparent that the altitude measurements received via Ubisense are not ideal due to a lack of precision with the z coordinate and a slow update rate. There are several possible filter configurations available that can use the position data and integrate it with the translational acceleration received from the IMU of the quadrotor. For our purposes, we attempted implementing both a simple complementary filter (CF) and a Discrete Kalman Filter (DKF). Ultimately, we settled on the DKF since it provided more reliable results.

B. DISCRETE KALMAN FILTER

The DKF has several advantages over the complementary filter depending on its application. Using the same inputs of measured z -position and acceleration, it can recursively predict estimated values for the states of the quadrotor (position, velocity and acceleration) [10]. Instead of tuning gains by trial and error to achieve satisfactory performance using the CF technique, the DKF will continuously solve for the ideal gains, based on matrices that represent the process noise covariance and measurement noise covariance of the system. Essentially, these two matrices (labeled Q and R respectively) are the only parameters that require tuning for the observer after the state and control input matrices have been appropriately designated.

Figure 13 shows a standard implementation of the Kalman Filter in Simulink. The top loop determines the a priori estimation by using the system's state and input matrices (A and B) and then uses the same a priori estimation, state measurements, output matrix (C), and calculated Kalman gain to provide estimated states. The entire bottom loop of this structure is only used to produce the Kalman gains that are fed to the upper loop to determine estimated states. This gain is calculated such that the trace of the a priori error covariance matrix is minimized. As can be seen, the noise covariance matrices are used in the bottom loop.

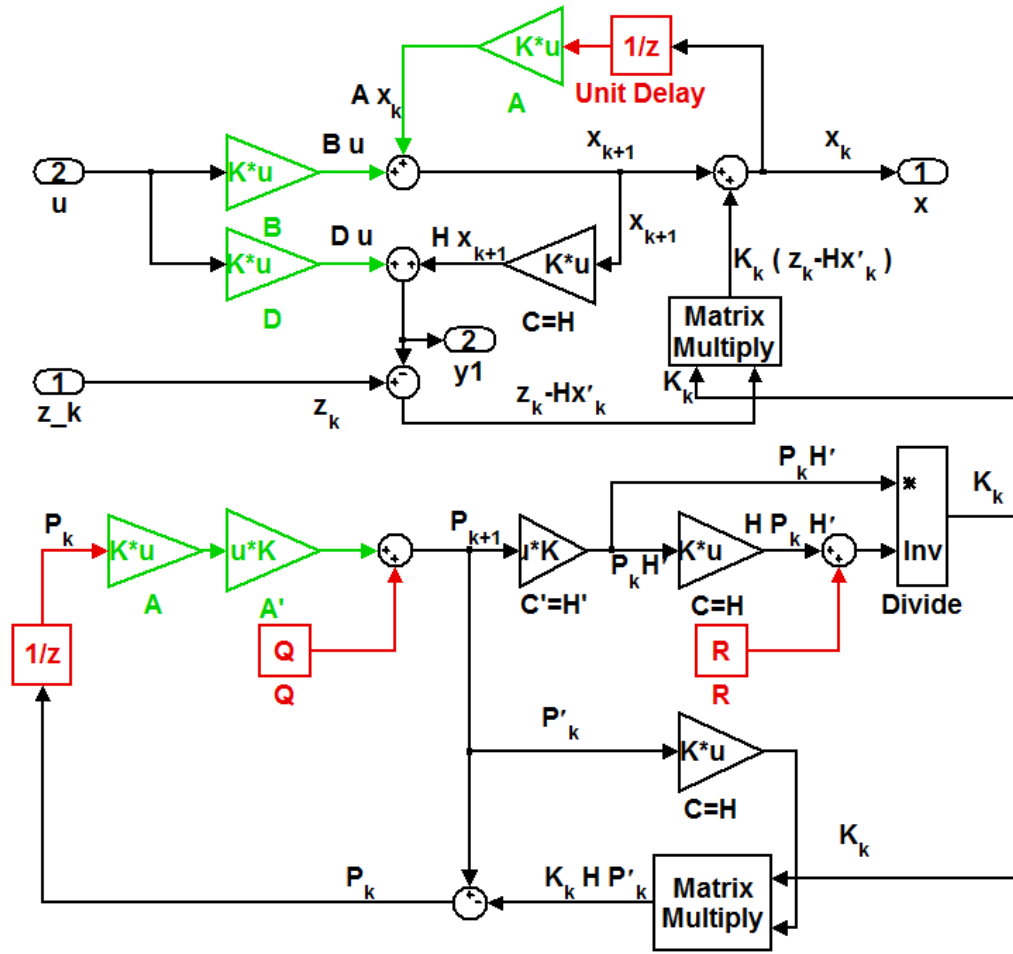


Figure 13. Discrete Kalman Filter Implementation (From [10])

The measurement noise covariance matrix, R , is the easier of the two matrices to determine. It is a two by two matrix due to there being two estimated states (acceleration and position). There is no codependent error caused from the measured states since the IMU accelerometer measurements have no effect on the Ubisense measurements and vice versa. Therefore, the goal is only to estimate values for the main diagonal; representing the individual variances of each state. The position variance was determined by picking an appropriate standard deviation based on the results of the Ubisense measurements. Since a mean error using set Information Filtering from Ubisense was found to be 0.08 meters, we rounded up the position standard deviation to 0.1, which is then squared to achieve the variance. Rounding up was acceptable since a higher assigned variance

indicates less confidence in the measurements of the system and is consequently a more conservative choice. The acceleration standard deviation was also set to the same value as the position due to experimental observations.

With the measurement noise variances set, the process noise covariance matrix, Q , is the only set of values left for tuning the DKF. This is appropriate since the process noise is very difficult to observe in any accurate way. Therefore, it was simply a matter of trial and error to find values for the main diagonal that achieved acceptable observer performance. The variance values chosen are an order of magnitude lower than the R matrix since it is a safe assumption that the process noise is significantly lower than the noise in Ubisense and the accelerometer. Also, Q is a three by three matrix since the additional state of velocity is pertinent. Equations 4.1 and 4.2 show the finalized values of the R and Q matrices:

$$R = \begin{bmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{bmatrix} \quad (4.1)$$

$$Q = \begin{bmatrix} 0.01^2 & 0 & 0 \\ 0 & 0.01^2 & 0 \\ 0 & 0 & 0.01^2 \end{bmatrix} \quad (4.2)$$

A discrete Kalman filter's gains should always converge to constant values for time invariant systems. Our system is indeed time-invariant, so this property should hold true. Therefore, the Kalman gains that correspond to the estimated altitude only are plotted in Figure 14. It is important to notice that all three gains converge relatively fast. The slowest gain converges in just under 0.15 seconds. This observation demonstrates that the gains are essentially constant, and no significant loss of performance is realized from this time delay.

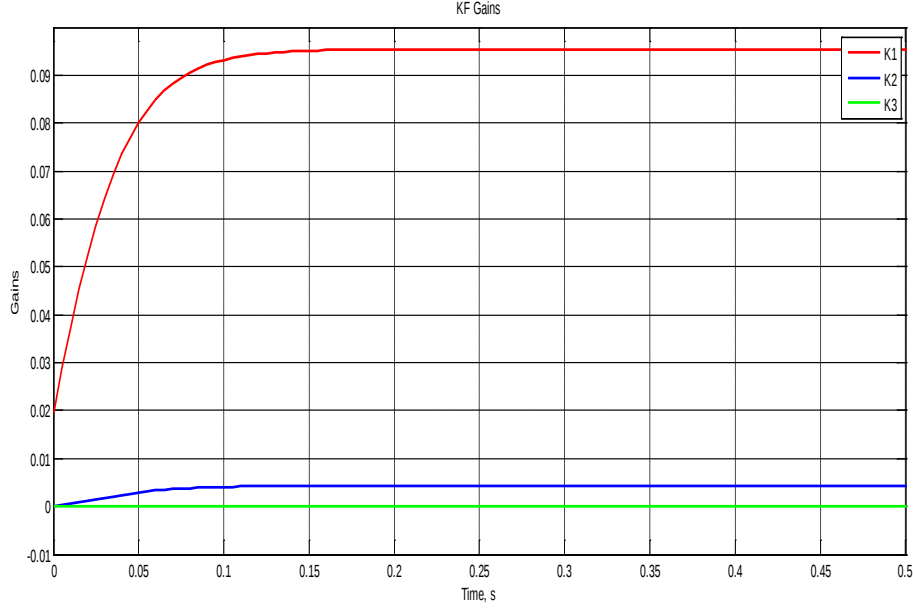


Figure 14. DKF Estimated Altitude Gains

Within our Simulink model, we implemented an “Enable” trigger that allows the DKF to process data for a few seconds before the actual controller is switched on. Therefore, even though convergence rates are fast, any negative effects from this delay are not experienced by the controller’s response. Due to the Kalman Filter’s more precise nature of gain estimation through minimizing error covariance, and the low gain convergence time, the DKF was an apt choice for the specific variety of model observer.

C. DISCRETE KALMAN FILTER RESULTS

In order to observe the effectiveness of the DKF implementation, the quadrotor was flown manually, ascending and descending, while both the z-position from Ubisense and the z-position estimated from the DKF were plotted together. An example of one such plot is shown in Figures 15 and 16. Both Figures include results from the same flight, but Figure 16 is zoomed in closer. Figure 15 demonstrates the observer’s ability to track the input signal reasonably well. Figure 16 emphasizes the ability of the DKF to filter the Ubisense signal, and also clearly represents the slight lag in the estimated state.

This is the singular negative cost to using an observer in the model. However, this lag is significantly smaller than the step size of each Ubisense tag update (10 Hz) and the benefits of the DKF greatly outweigh this cost.

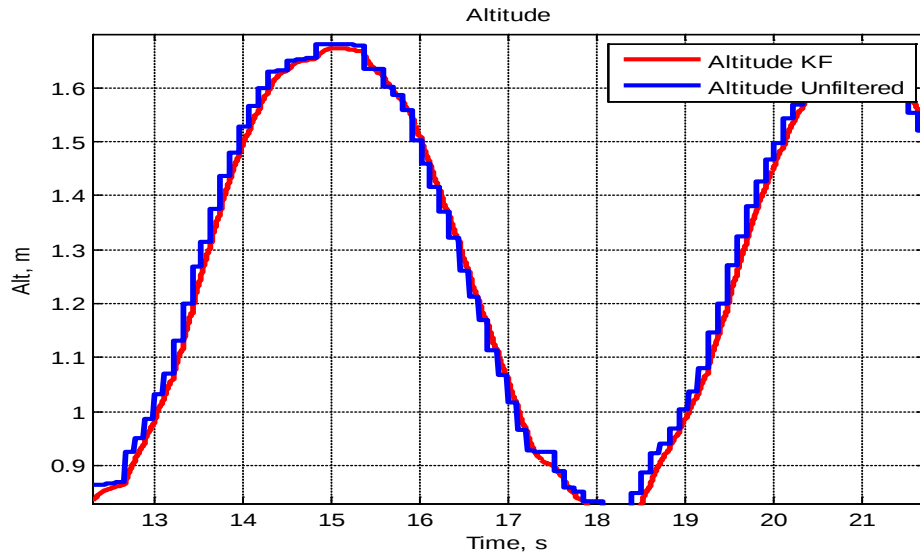


Figure 15. Unfiltered Signal vs. DKF Signal

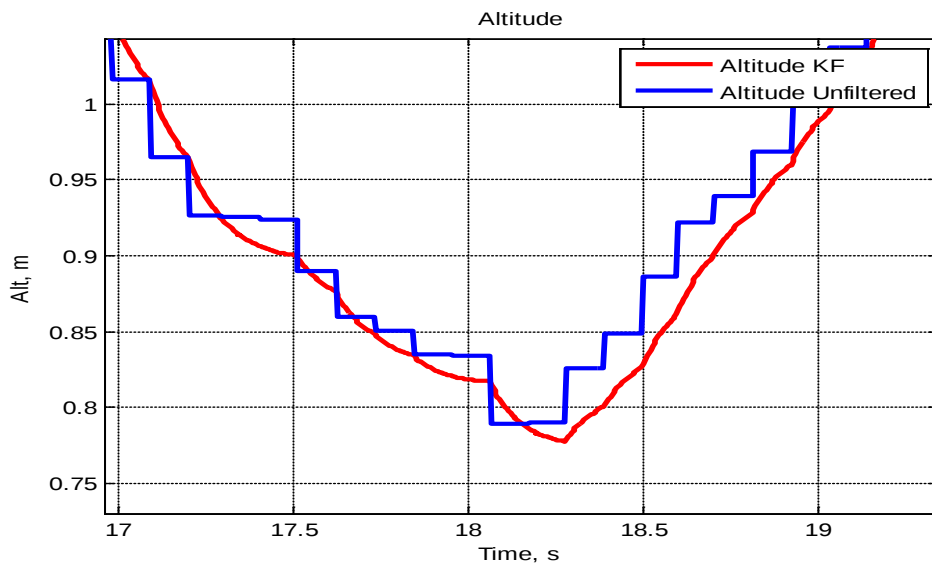


Figure 16. Unfiltered Signal vs. DKF Signal (Larger Scale)

D. CONTROLLER DESIGN

After deciding upon an effective observer to estimate the measured states of the physical system, it is now necessary to discuss the process for developing a controller. The goal of the controller is to use the observer's states to command the total quadrotor thrust necessary to bring the UAV to a commanded altitude quickly, safely, and with minimum steady state error. From Chapter I, it was determined that our initial physical model was simplified to a double integrator. The process for creating a compensator that ensures both stability and zero steady state error for this model is illustrated in Figure 17. The specific steps shown will each be described in the following sections.

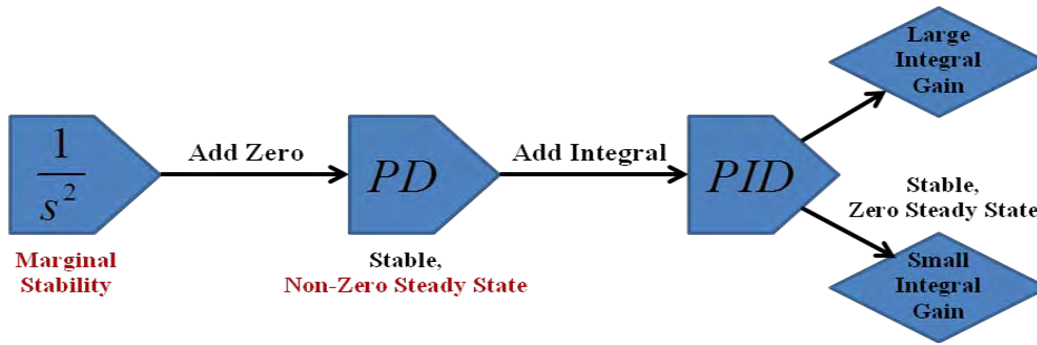


Figure 17. Controller Design Flowchart

When plotted on a root-locus diagram, the double integrator model contains two locus branches that run vertically up and down the imaginary axis. Therefore, regardless of chosen gain values, the system will always be marginally stable. A compensator must now be implemented to improve stability.

E. P-D CONTROLLER ANALYSIS

To stabilize this system, we first attempt a PD (Proportional Derivative) Controller. This will introduce a zero into the original physical model. A typical PD compensator multiplies the transfer function by the following expression [11], [12]:

$$K_p + K_d s \quad (5.1)$$

Above, K_p is the proportional gain and K_D is the derivative gain. These values can be initialized and tuned later to improve performance and stability. To assist in choosing the appropriate zero, equation 5.1 can be rearranged as shown in 5.2:

$$K_D \left(s + \frac{K_p}{K_D} \right) \quad (5.2)$$

It is now plain to see that the zero for the PD compensator is equal to the ratio of the proportional gain to the derivative gain. The closed loop system with the PD compensator can be viewed in the root locus and Bode plot combination displayed in Figure 18. In this example, the root locus shows a circle that is tangent to the imaginary axis and, consequently, marginal stability is mostly avoided in favor of absolute stability. However, it is important to note that the phase margin decreases for higher frequency values of the zero introduced by the PD. Therefore, the phase margin decreases for greater values of proportional gain and for lesser values of the derivative gain. From a pure stability point of view, a greater value (positive) of phase margin indicates greater stability and robustness. This consideration will be useful later when picking initial gain values for the controller. The desired settling time of the system can also be utilized to help determine these values. The applicable block diagram representing the PD compensator's implementation into the model is shown in Figure 19. This gives the closed loop transfer function for the system shown in equation 5.3.

$$\frac{Z(s)}{Z_c(s)} = \frac{K_p + K_D s}{s^2 + K_D s + K_p} \quad (5.3)$$

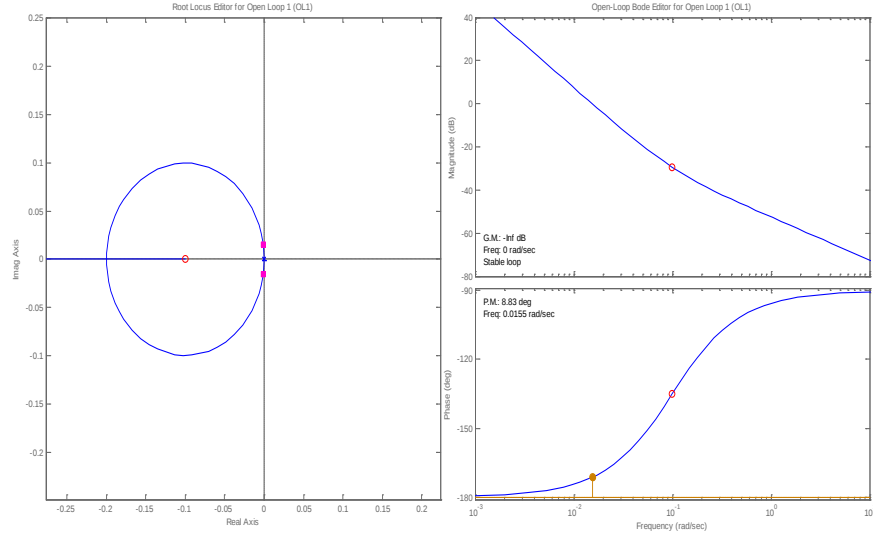


Figure 18. PD Closed Loop System Root Locus & Bode Plot

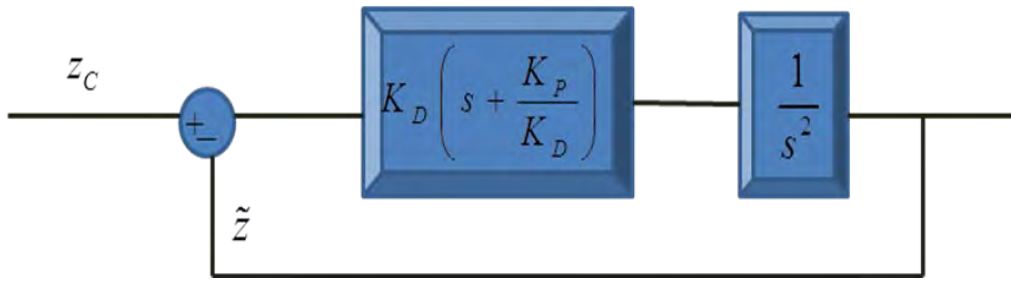


Figure 19. PD Controller Block Diagram

Next, it is useful to find the numeric representation of the PD's transfer function that will satisfy some design criteria. MATLAB's SISO Toolbox provides the helpful functionality of specifying the aforementioned design criteria on the root locus plot. From the final section in Chapter II, it was determined that the controller should have a slower settling time than the two second delay encountered from Ubisense. Designating a two second settling time in SISO Toolbox marks a vertical line orthogonal to the real axis that marks that settling time. Closed loop poles to the right of that line indicate that the system will theoretically settle slower than two seconds. In order to have another

parameter to determine the value of the compensator, I also set a requirement of maximum overshoot less than 18 percent. This constrained the usable area for the plot even further to provide an appropriate damping ratio. A specific compensator was calculated once each criterion was satisfied, along with a phase margin greater than 45 degrees for stability. This gives a compensator transfer function as shown in equation 5.4:

$$C(s) = 10.744(1 + 0.43s) = 4.6(s + 2.33) \quad (5.4)$$

This indicates an open loop zero at -2.33. Therefore, from the format of (5.2), this compensator gives a derivative gain of 4.6 and a proportional gain of 10.76. It also results in a damping ratio of 0.707, which is a standard optimally under-damped value. Ideally, this open loop transfer function is an appropriate starting point for tuning gains throughout actual flight testing. However, the assumption that the physical system is only a double integrator was a significant simplification and will certainly require different parameters before performance of the quadrotor can be acceptable. Additionally, the compensator is still insufficient as will be explained in the following section.

F. PID CONTROLLER

The difficulty with settling for a PD compensator is that the derivative term amplifies noise and can seriously degrade the steady state performance of the system. Now, to move to the final step of the controller design flow chart from Figure 17, it is necessary to add an integral term. This integral term is the only piece of the compensator that can keep track of the error history and eliminate the steady state error over time. Without it, there would be a constant bias on the altitude error which could lead to serious problems in autonomous maneuvers. Now, when a step input is introduced for the commanded altitude, the system will maintain stability and theoretically converge to zero error since it is a Type 1 system versus Type 0.

This is now a complete PID controller. The task of tuning each of the three gains will be described in Chapter V, but for now, it is helpful to investigate a starting gain for the integral term. A simple qualitative analysis is to compare the effects of a high integral gain and a low integral gain on both transient and steady state performance. This is achieved through use of yet another root-locus plot. Figure 20 illustrates this comparison on the same plot.

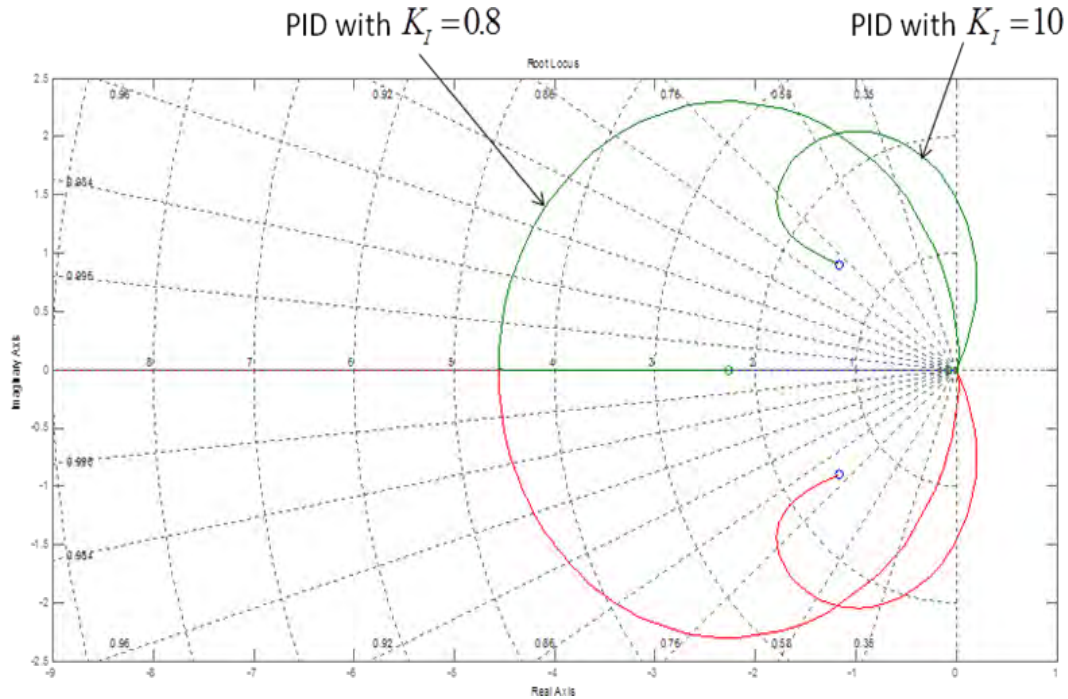


Figure 20. PID Compensators Using High and Low Integral Gains

The root-locus of the controller with a high integral gain has a large region of instability (locus branches to the right of the imaginary axis). Consequently, this compensator is unacceptable for the system due to its lack of robustness. In contrast, the root-locus of the controller with low integral gain predominately occupies territory in the left-half plane and only has an insignificantly small region of instability. This region (which can only be viewed if diagram is zoomed in near the origin) can be avoided through placement of the closed loop poles. Therefore, when choosing a starting integral

gain in the next chapter, it is prudent to choose a small value. A small value can be defined as an order of magnitude lower than the derivative and proportional gains. In Figure 20, 0.8 was a reasonable estimation.

THIS PAGE INTENTIONALLY LEFT BLANK

V. FLIGHT TESTING

A. INTRODUCTION

This chapter represents the culmination of the efforts in this thesis. The entire process will be proven repeatable and ready for continued research if it is possible to execute a flight test indoors using data from the Ubisense RFID system (Chapter II), a robust communication structure (Chapter III), and an appropriate observer and controller (Chapter IV). The outcome of this process is described in the following sections. A photo of the quadrotor holding altitude autonomously is shown in Figure 21.



Figure 21. Quadrotor Hovering Autonomously

B. CONTROLLER TUNING

The next step is to adjust each of the gains in the controller to optimize the quadrotor's altitude hold control. Appropriate starting gains, before tuning, have been estimated in the previous chapter. An additional parameter of significance is the nominal

thrust value. This should be set as a percentage of total thrust that effectively keeps the quadrotor hovering at a single altitude while in manual control.

The majority of the flight tests served the purpose of adjusting gains and nominal thrust from their initial values. Initially, we were able to gauge the responses of the quadrotor with certain degrees of gain adjustment in order to understand the appropriate ranges of values to use for the tuning process. From that point forward, it was simply a matter of using pre-established knowledge of effects of increasing each gain to arrive at desirable transient and steady state characteristics. Table 1 shows a list of effects of increasing each of the three gains in a PID controller. Not all of these characteristics were affected experimentally in the same way described by the table, but it was certainly a helpful tool for the process.

Table 1. Effect on System Characteristics by Increasing Gains

Response	Rise Time	Overshoot	Settling Time	S-S Error
K_P	Decrease	Increase	NT	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	NT	Decrease	Decrease	NT

In Table 1, “NT” indicates no tangible effect. Using these adjustments as a guide, it was then a matter of trial and error to achieve a flight test that had a quick rise time, and settling time while minimizing overshoot and steady state error. Ultimately, this tuning led to the values of 1.4 for proportional gain, 5.7 for derivative gain, 0.8 for integral gain and a nominal (hover) thrust input of 45% of the total thrust available to the quadrotor. Results using these values are discussed in the following section.

C. FLIGHT TEST RESULTS

With the optimal gains, several successful flight tests were conducted with acceptable altitude holding results. Results from three of these tests were chosen to

represent some varying responses with the same gain values. The variations could be due to factors such as differing amounts of remaining battery power, air resistance and friction between the steel wires and the quadrotor's brackets. However, it is more likely that these variations were caused mostly by the dynamic performance of Ubisense measurements at any given time. Each of the following plots (Figures 22–24) demonstrates a satisfactory response to the commanded altitude unit-step input.

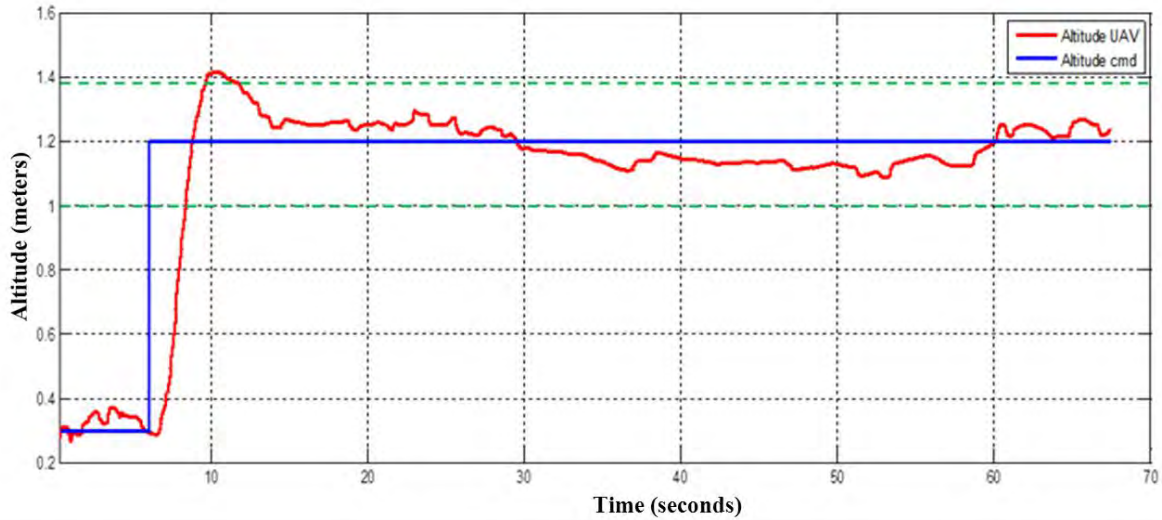


Figure 22. UAV Altitude from First Successful Flight Test

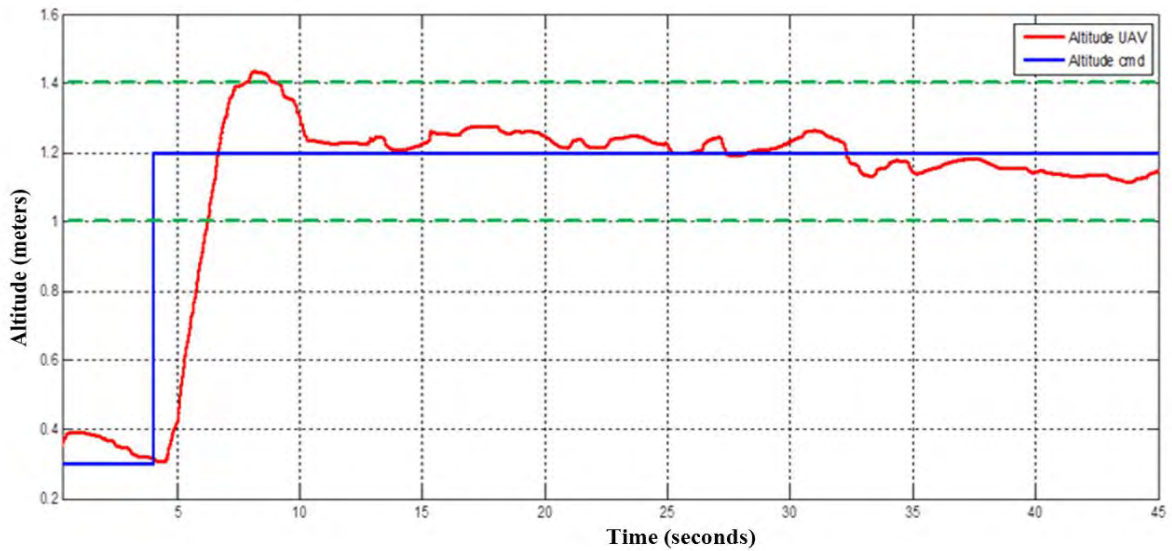


Figure 23. Second Test Flight

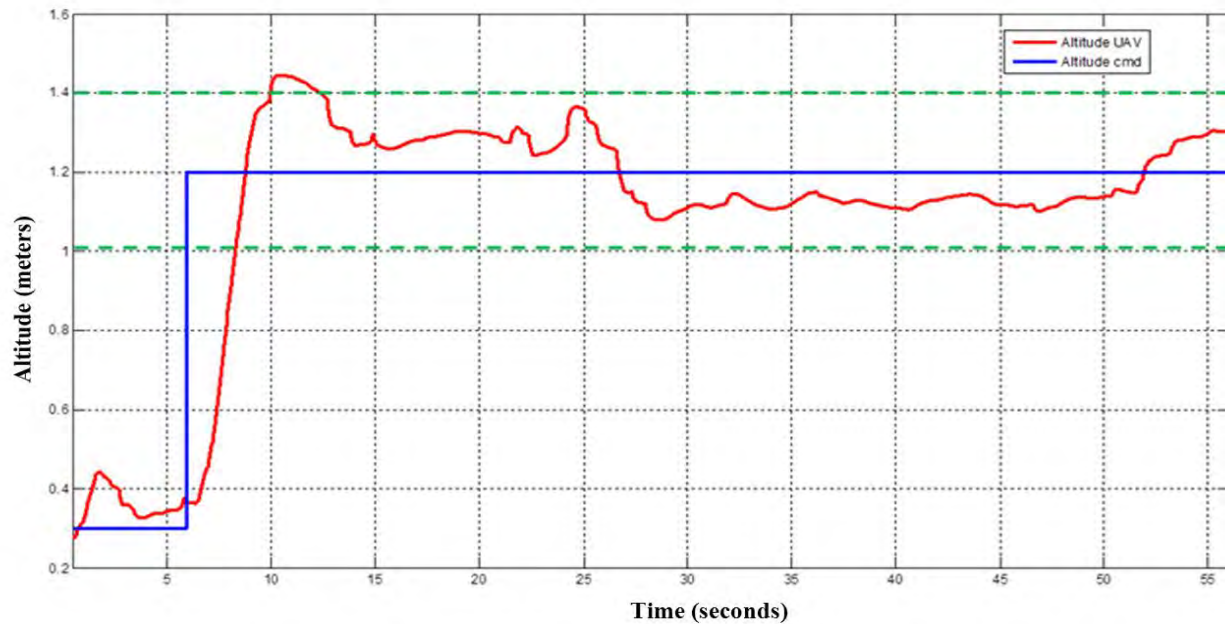


Figure 24. Third Test Flight

In each plot, the commanded altitude is shown in blue, the green dotted lines represent the bounds of an error region targeted for ± 20 cm, and the red line is the measured altitude of the quadrotor over time. Initially, the quadrotor is at rest on the 0.3 meter landing platform and is then commanded to 1.2 meters after switching into autonomous flight. The UAV then ascends above the mark (thereby defining the overshoot margin) and proceeds in time to its steady state altitude; eventually cancelling the altitude error. The flights shown here are only about 60 seconds long, but, with these settings, the quadrotor was able to remain well within 20 cm of the target altitude for as long as the battery lasted in each attempt. The individual and averaged performance characteristics are shown for the three recorded flights in Table 2.

Table 2. Transient and Steady State Response Characteristics

	Rise Time	Settling Time	Max % Overshoot	Steady State Error
1st Flight	0.94 s	5.48 s	17.83 %	+/- 0.110 m
2nd Flight	1.67 s	5.12 s	11.67 %	+/- 0.085 m
3rd Flight	1.96 s	6.45 s	20.42 %	+/- 0.165 m
Averages	1.52 s	5.68 s	16.64 %	+/- 0.120 m

An average rise time less than 1.6 seconds is a good result for a UAV to climb 0.9 meters. Of course, a faster rise time usually leads into an excessive overshoot, but 16.64% is reasonable in this scenario since it only correlates to the quadrotor departing its commanded altitude by 21 cm. The settling time was calculated by the vehicle's arrival within an error less than 20 cm. This was chosen since the error of the Ubisense system is designated as 15 cm by the company, and considering additional noise in the system, 20 cm is a sensible upper and lower bound on error. It is of interest that the settling time of 5.68 seconds is an additional 3.68 seconds slower than the 2 second lag of Ubisense. While it is true that we did aim to have the settling time exceed 2 seconds, 5.68 seconds is a bit excessive for an average response. It would depend on future applications of this algorithm to determine whether it would be worth increasing the maximum percentage overshoot in order to cut down on the settling time. I would submit that this performance would be sufficient for pursuing algorithms to test controls for quadrotor pitch and roll, but it has not achieved the level of robust control necessary for testing the coordination of multiple vehicles or collision avoidance.

D. CONCLUSIONS

The data collected in the previous section proves the successful execution of an altitude hold algorithm for a quadrotor UAV using indoor position sensors. The hummingbird was able to maintain its altitude with a margin of error less than the Ubisense-rated 15 cm. However, this was only completed under ideal conditions using

the vertical channel isolation frame. The architecture created holds promise for further research, but there still were some serious difficulties and limitations that will be addressed in the final chapter.

VI. LIMITATIONS AND FUTURE WORK

A. PROJECT LIMITATIONS

As can be inferred from the previous chapters, the majority of limitations encountered throughout this research were involved with the Ubisense RFID system. Primarily, the delay in response time of the system significantly hindered improvements in the transient response of the quadrotor. The resulting slow settling time will cause added difficulty once another student/researcher can begin to implement controls in three dimensions. A delay of this magnitude compounds position error in any complex maneuvers attempted.

Also, despite producing an acceptable degree of accuracy in many flight tests, the readings from Ubisense are not entirely reliable for repeated experiments. The best results were achieved in the center of the coverage area, and straying from this location would greatly degrade the readings. There were too many variables involved that required perfect alignment for Ubisense to deliver its top performance. At times, results were unacceptable simply because the Ubisense tag was facing a non-ideal direction. This high degree of sensitivity to its environment lowers overall confidence in the ability of the UWB method to handle other flight tests where such errors can compromise safety and mission effectiveness.

B. PROPOSED TECHNOLOGY IMPLEMENTATION

One feasible alternative for the accuracy problem in the vertical channel is to augment the setup by attaching an ultrasonic altitude sensor to the bottom of the quadrotor. This only presents minor complications due to integration in the communications structure, but can greatly benefit overall performance. The sensor currently available in the lab is the XL-MaxSonar-EZ manufactured by MaxBotix. This small device has a rated resolution of 1 cm (versus 15 cm with Ubisense) and has a similar update rate of 10 Hz [13]. The z-position data received could be fed through a

similar Kalman Filter that would be capable of using measurements from both the ultrasonic and the UWB sensors. Ubisense data would also still be the primary tool for measuring the x and y-coordinate positions.

Unfortunately, this device does come with its own limitations. First, it would require use of the quadrotor's gyro readings for rotational acceleration to trigonometrically determine the actual altitude when roll and pitch angles are not zero. Also, the ultrasonic sensor has a maximum range of 7.65 meters, but this would not be an issue in our lab since the quadrotor has to stay below 4.5 meters to maintain Ubisense coverage. However, if the quadrotor experiences a high angle of pitch, then it is possible to exceed this maximum range. If the sensor was implemented successfully for altitude readings, then they could also be utilized for the prediction stage of collision avoidance algorithms. As long as the Hummingbird's maximum payload was not violated, it would be possible to mount several sensors around the quadrotor for detection of other vehicles or stationary objects. These ranges to an unknown object could then be used to calculate a projected path of the obstacle and a resulting collision avoidance maneuver.



Figure 25. MaxBotix Ultrasonic Range Finder (After [13])

Another alternative that could solve the problem of response delay and accuracy is the installation of Vicon motion tracking cameras to replace Ubisense altogether. Vicon uses infrared light reflecting on small spheres that can be mounted on the quadrotor. With an ideal setup where 28 cameras are installed, the system would have a possible latency of 4ms (250 Hz) and a submillimeter accuracy (more than two orders of

magnitude better than Ubisense) [14]. If the necessary funding for this equipment was available, then it would be a tremendous asset to the UAV research abilities of any school or facility.

C. FUTURE WORK

Upon conceptualization of this thesis, there were several optimistic plans for flight testing that were unable to materialize due to time and equipment constraints. However, once the necessary safety nets are in place (literally and figuratively), the next step for research is to attempt running the same altitude hold control without the use of the z-axis isolation frame. This would serve as a more realistic test of the control since the cables that previously guided the quadrotor may have caused natural damping of the autonomous response due to friction against the brackets.

After continued success without the frame, it would then be safe to attempt opening up control of the quadrotor's attitude for forward and lateral movement. This would require implementation of the full dynamics of the vehicle. It seems feasible to execute some short-range, simple maneuvers in this fashion, however, in order to accomplish any aggressive maneuvers, time coordinated flight with multiple UAV's or collision avoidance, it is my opinion that different technology (as discussed in the previous section) should be considered. Assuming that the accuracy of the position measurements could improve by one order of magnitude, and that the time delay in response could be cancelled, then the research possibilities could vastly improve.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- [1] *AscTec Hummingbird with AutoPilot User's Manual*, Ascending Technologies, 2010.
- [2] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne and A. K. Cooke, "Direct method based control system for an autonomous quadrotor," *Journal of Intelligent & Robotic Systems*, vol. 60, no. 2, pp. 285–316, 2010.
- [3] P. Castillo, R. Lozano, A. Dzul, "Stabilization of a mini rotorcraft with four rotors," *IEEE Control Systems Magazine*, vol. 26, pp. 45–55, December 2005.
- [4] V. Cichella, "An innovative approach for the UAV's path following control and application on a X-3D-BL quadrotor," M.S. Thesis, Univ. of Bolgna, Bologna Italy, 2011.
- [5] Z. Sahinoglu, S. Gezici and I. Guvenc, *Ultra-wideband Positioning Systems*, New York: Cambridge University Press, 2008.
- [6] Ubisense Technical Staff, "Section 1: Overview," *Ubisense RTLS Training*, Ubisense, 2008.
- [7] Ubisense Technical Staff, "Section 4: System Tuning," *Ubisense RTLS Training*, Ubisense, 2008.
- [8] Oxford Mobile Robotics Group, "MOOS: Introduction," 2009, <http://www.robots.ox.ac.uk>
- [9] ZigBee Alliance, "ZigBee Specification Overview," 2011, <http://www.zigbee.org/Specifications/ZigBee/Overview.aspx>
- [10] "Introduction to Kalman Filtering," class notes for ME4821, Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, June 2011.
- [11] M. Driels, *Linear Control Systems Engineering*. New York: McGraw-Hill, 1996.
- [12] K. Ogata, *Modern Control Engineering*, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2002
- [13] Maxbotix Inc, "XL-MaxSonar-EZ Sensor Line," 2011, <http://www.maxbotix.com>
- [14] M. Kocourek (private communication), Senior Business Development Manager, Vicon, Los Angeles, 2011.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Professor Isaac Kaminer
Naval Postgraduate School
Monterey, California
4. Professor Vladimir Dobrokhodov
Naval Postgraduate School
Monterey, California