University of Nevada, Reno

Control of a Quadcopter Aerial Robot Using Optic Flow Sensing

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Mechanical Engineering

by

Michael Brandon Hurd

Dr. Kam K. Leang/Thesis Advisor

December 2013

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

MICHAEL BRANDON HURD

entitled

Control of a Quadcopter Aerial Robot Using Optic Flow Sensing

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Kam K. Leang, Ph.D., Advisor

Cahit Evrensel, Ph.D., Committee Member

Yantao Shen, Ph.D., Graduate School Representative

Marsha H. Read, Ph.D., Associate Dean, Graduate School

December 2013

Abstract

This thesis focuses on the motion control of a custom-built quadcopter aerial robot using optic flow sensing. Optic flow sensing is a vision-based approach that can provide a robot the ability to fly in global positioning system (GPS) denied environments, such as indoor environments. In this work, optic flow sensors are used to stabilize the motion of quadcopter robot, where an optic flow algorithm is applied to provide odometry measurements to the quadcopter's central processing unit to monitor the flight heading. The optic-flow sensor and algorithm are capable of gathering and processing the images at 250 frames/sec, and the sensor package weighs 2.5 g and has a footprint of 6 cm^2 in area. The odometry value from the optic flow sensor is then used a feedback information in a simple proportional-integral-derivative (PID) controller on the quadcopter. Experimental results are presented to demonstrate the effectiveness of using optic flow for controlling the motion of the quadcopter aerial robot. The technique presented herein can be applied to different types of aerial robotic systems or unmanned aerial vehicles (UAVs), as well as unmanned ground vehicles (UGV).

Acknowledgments

First and foremost, I would like to thank my advisor, Kam, for his guidance and support throughout this entire process. This work would not have been possible without his help.

I would like to thank my lab mates, Norman Johnson, Robert Riddle, Maxwell Fleming, Brian Kenton, Joel Hubbard, Christopher Dudley, Alex Woods, Marissa Tsugawa, and James Carrico for all the laughs, good times, and helping to point me in the right direction.

I would also like to thank Erica for her amazing support and constant encouragement, which helps make me a better person. Words can't express how grateful I am to have you in my life.

Finally, I would like to thank my Mom, Dad, and sister Leigherin for their unconditional love and support and for always believing in me.

Dedication

To Mom and Dad.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter discusses the goal, objectives, and contribution of the research. The motivation and state-of-the-art work are also presented.

1.1 Goal, Objectives, and Contribution

The goal of this thesis is to control the motion of a quadcopter aerial robot using optic flow sensing. The objectives to achieve the goal are to:

1. Design and build a quadcopter robotic platform that incorporates optic flow sensing;

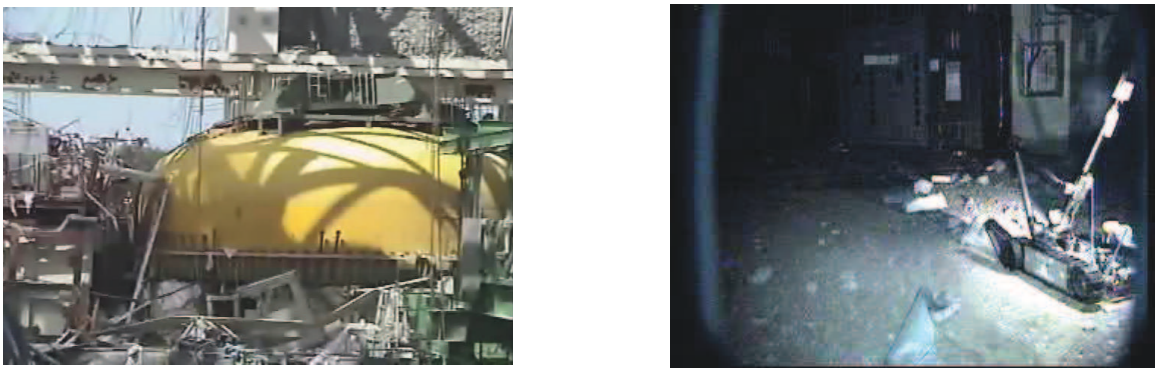
2. Develop control algorithms that incorporate optic flow information to control the motion of the robotic platform; and

3. Measure and quantify the performance of the robotic platform.

The contribution of this research work is investigating the feasibility of using optic flow sensing for controlling the motion of an aerial robot. The results of this work can be applied to the development of novel on-board localization strategies for aerial robotic platforms.

1.2 Motivation

Unmanned aerial vehicles (UAVs) can have unique qualities like small size, high maneuverability, and because of the lower costs they are somewhat disposable, which make them ideal for a wide variety of applications. They have exhibited the ability to maneuver through tight places [3], lift heavy objects [4], build structures [5], and serve as remote surveillance [6]. There are other potential uses of UAVs, which include serving as a wireless network infrastructure [7], inspecting structures [8], and aerial remote mapping [9]. Disasters like the one at the Japanese nuclear power station in



(a) (b)

Figure 1.1: Fukushima Daiichi nuclear power station inspection after the March 11, 2011 nuclear disaster [1]. (a) Photo taken by Honeywell T-Hawk while inspecting reactor. (b) IRobot 510 Packbot inside 1st floor of unit 1 reactor building.

2011 proved to be a great place to utilize unmanned systems. Figure 1.1(b) shows the iRobot 510 Packbot inside the 1st floor of unit 1 reactor building. The 510 Packbot was used to preform various tasks while maneuvering throughout the nuclear power station. It was successful at turning off valves, opening doors, and recording radiation readings throughout various rooms. The drawback of using a ground vehicle like this is they are unable reach high places or maneuver up to different levels of a building quickly and efficiently. The Honeywell T-Hawk as seen in Fig. 1.2 was used to survey the damage on the outside of the power station, which can be seen in Fig. 1.1(a).

The T-Hawk is a well designed MAV and is capable of sustained flights of 50 mins, but it isn't suited for indoor flight because it omits gaseous fumes and needs Global Positioning System (GPS) to localize it's position. GPS is used to localize a vehicles

Figure 1.2: Honeywell T-Hawk MAV [2].

position, it works by triangulating its location from satellites that are visible by the device. The signal from satellites are too weak to travel through the walls of a building, so the use of GPS to localize position is not effective indoors. This is a problem when trying to maneuver through a building. There needs to be a way to localize the quadcopter while navigating indoor environments, that way it won't run into objects or walls.

1.3 Camera Control of a Quadcopter

Quadcopter navigation via cameras has been done in many different ways and is summed up in the following paragraphs. These methods include: a constellation of off board cameras, on-board camera with off board processing, on-board cameras with on-board processing, and on-board vision chips with on-board processing. A constellation of cameras is an effective way of controlling the position of a quadcopter [10–16]. This method of position control is done by monitoring reflective targets that are attached

to the quadcopter and determining the position based on one or more cameras. This system is called a Vicon motion capture system. There are other systems available out there, but the Vicon is the most widely used system by researchers. The drawback of this system is that the quadcopter isn't completely autonomous. A quadcopter can be considered autonomous when it only relies on sensors that are on-board and features in the environment for controlling it's motion.

Instead of using off-board cameras a on-board stereo vision system can be implemented [17, 18]. Stereo vision is computationally heavy because instead of just comparing two images, one from the past and one from the present. Now four separate images need to be compared meaning the past and present images of the right and left cameras need to be compared with each other. Processing multiple large images on-board a quadcopter requires the use of a powerful processor, which can consume lots of power from the battery [17–19]. A way to allow for the processing of multiple images is to have a wireless video feed to a ground station. When the images are sent to the ground station they are processed and then useful data is sent back to the quadcopter to help stabilize it's movement [20–24]. Wirelessly sending a video feed to a ground station can be challenging. Slow transfer rates and a delay from the time the image is taken to the time it is used in the quadcopter poses a problem. Furthermore, the range of the quadcopter would be limited to the range of the signal, which can be greatly reduced from walls when navigating indoors. For these reasons on-board processing of images should be further explored.

The type of camera typically used on-board a quadcopter are Complementary Metal Oxide Silicon (CMOS) or Charge Couple Device (CCD) camera. An open source/open hardware platform was developed using a CMOS camera that is downward facing, which is used in the computation of optical flow and a sonar sensor is used to determine altitude [25] of the UAV. This combination of camera and sonar

can be seen used by a few other research groups [26, 27]. A single camera placed at the bottom of a quadcopter has also been used by various researchers to determine horizontal movement of a UAV [24, 28–30]. This method can be done so all of the image processing is on-board the quadcopter.

Other earlier work that's been done also includes the use of computer mouse sensors like the ADNS-6010 [31–33]. These little cameras are found in older computer mice, which are cheap and also offer post processing that calculates the velocity as it is moved. The post processing of the images frees up a lot of processing power that the quadcopter could be using for main control functions. The downfall of a sensor like this is the lack of light that in an indoor environment, these sensors are better suited for outdoor flight. To combat the downfall that a mouse sensor has a different camera can be used.

Since weight is an issue lower quality images should be used to reduce processing time. A vision chip is a viable option for indoor navigation. A vision chip is similar to a CMOS camera in the way it collects images, but the difference is that some processing of the images can be done before they are sent to a microprocessor. This type of setup is similar to the ADNS-6010, but the vision chip can work well in different lighting conditions, the size and resolution of the image can be selected, and the processing time can be atleast ten times faster.

1.4 Organization

This thesis is organized as follows. Chapter 2 presents a review of various autonomous air vehicles that use optic flow for navigation and hovering. Chapter 3 presents optic flow sensing, the various methods, and how it can be used to control an aerial vehicle. Chapter 4 about the design process for the quadcopter and reviews the various components of the quadcopter system for this thesis. Chapter 5 discusses the

experimental setup, which includes the hardware and the various software used to collect data. The results are presented and discussed in chapter 6 followed by the conclusion and future work in chapter 7. The appendices can be found at the end, which include the pseudo code to operate the quadcopter and the optic flow sensors.

Chapter 2

Background

This chapter provides background information on the current state-of-the-art for autonomous quadcopter systems and how optic flow has been used in UAV flight navigation.

2.1 Quadcopter Aerial Robots

A quadcopter is a four rotor helicopter. It has four arms that are attached to the main body and on each arm there is a motor with a rotor. For the quadcopter in this thesis the motors are actuated by a electronic speed controller (ESC). The ESCs are controlled by the control board, which is the brains and the main component of the aerial vehicle, see Fig. 2.1. Quadcopters posses features that make them ideal candidates for autonomous flight because they are much simpler than traditional helicopters due to the elimination of the swashplate and tail rotor. The swashplate in a traditional helicopter is used to increase the thrust and perform pitch and roll maneuvers; to increase the thrust on a quadcopter the angular velocity of the motors are increased because the pitch of each rotor is fixed. Furthermore the tail rotor is eliminated, which on traditional helicopters is used to stabilize the helicopter about the yaw axis. On a quadcopter the yaw rotation is prevented by using pairs of clockwise and counterclockwise rotating motors, since two rotors are counter rotating they

cancel the torque created by each other. The direction of motor rotations is illustrated in Fig. 2.2. Due to all of these features a quadcopter is easily maintained and is cheaper to replace parts compared to traditional helicopters. The other components

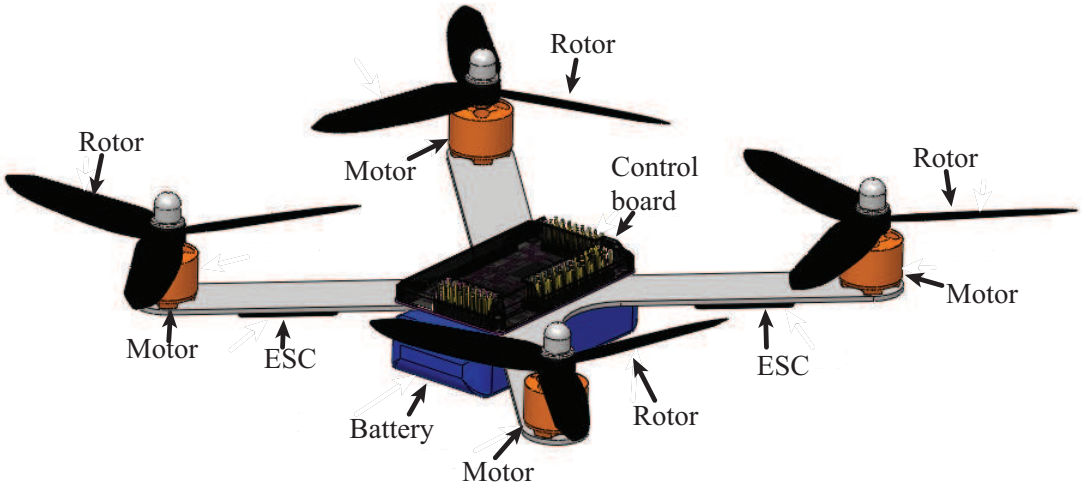


Figure 2.1: The simplest version of a quadcopter with all of the necessary components for flight.

that a quadcopter is composed of are: a control board, battery, and radio receiver. A few additional components were added to the basic quadcopter platform to make it better suited for the work in this thesis. These components include, two cameras that are used for optic flow sensing and a protective ring that adds safety for the operator and the quadcopter. As it is shown in Fig. 2.2 there are four motors denoted by M, where two rotate clockwise and two rotate counterclockwise. This is to counteract the torque produced by each pair of motors, so the quadcopter is stable about the yaw axis. This special configuration also eliminates the need to have a tail rotor, which conventional helicopters need to stabilize about the yaw axis. The quadcopter presented in this thesis has fixed rotors, which means the thrust is controlled by increasing the motor angular velocity. Different combinations of thrust T are used rotate and translate the quadcopter. The different actuations used by the quadcopter for flight are illustrated in Fig. 2.3. The rotation combinations of a quadcopter are

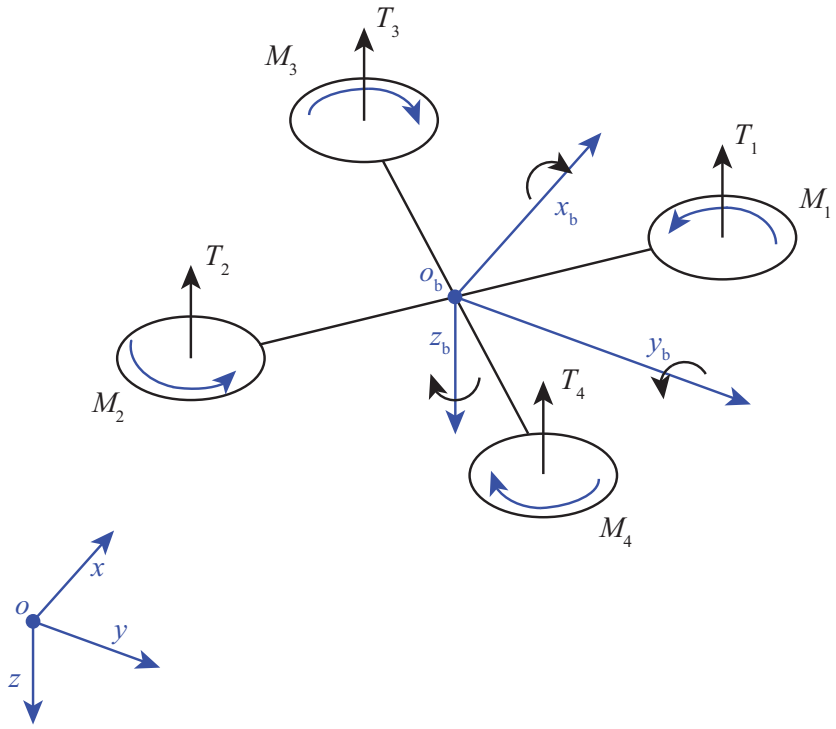


Figure 2.2: Model of the quadcopter in a right-handed coordinate system. The earth coordinate frame is represented by (x,y,z), and the origin is at o. The body reference frame origin o_b is at the quadcopter's center of mass and y_b is pointing forward on the quadcopter. Each motor M has a thrust component, which is T and is related to the angular velocity of the motor.

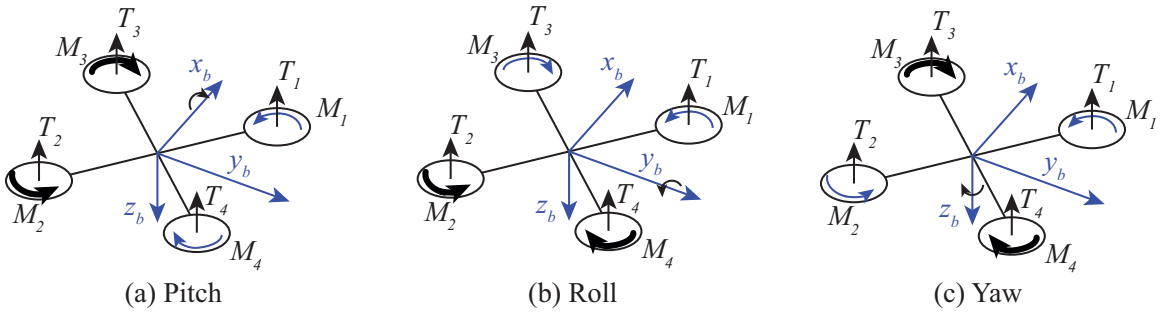


Figure 2.3: Depiction of quadcopter rotational motion. The quadcopter can rotate about all 3-axises, which are called pitch, roll, and yaw. (a) is the pitch motion where M_3 and M_2 are increased to produced the desired motion. (b) Rolling is done by actuating M_1 and M_3 or M_2 and M_4. (c) The quadcopter is actuated about the yaw axis by actuating M_1 and M_2 or M_3 and M_4

pitch, roll, and yaw, which are also used in (x,y,z) translations. A quadcopter can do a counterclockwise roll rotation when motor one and motor four increase their

angular velocity, while motors two, three keep the same or decrease their angular velocity shown in Fig. 2.3(b). The pitch motion is similar to the roll motion, but it is about the y-axis. Motors one and three increase angular velocity, while motors two and four stay the same or decrease their angular velocity seen in Fig. 2.3(a). The yaw motion is about the z-axis, which means for a counterclockwise motion motors one, two need to increase, while three, four decrease or keep a constant angular velocity shown in Fig. 2.3(c). The three combinations of rotations are the essential operations of a quadcopter to stabilize it's self while in flight.

2.2 Different Platform Designs

There are numerous quadcopter platforms available for purchase, which include: Ascending Technologies Pelican [34], Dji Phantom Aerial [35], and the Parallax ELEV-8 [36] to name a few. These quadcopters are fairly expensive and are far more expensive than the quadcopter platform presented in this thesis. The AR Drone [37] is a fairly cheap system, but it isn't suited for the expansion of additional sensors, isn't very strong due to the amount of plastic used, and the gearing used between the rotors and motors can potentially cause problems.

Most of the off-the-shelf platforms don't have anyway to localize themselves while flying through indoor environments. Which means that any unwanted tilting of the quadcopter that isn't picked up by the inertial measurement unit (IMU), which is composed of a three axis accelerometer and a three axis gyro, will result in the quadcopter drifting off course and not stopping until it runs into a object. This may result in the quadcopter crashing. A quadcopter's IMU can not sense the drift if the quadcopter is drifting at a constant velocity due to the lack of transnational position or velocity sensors and error in the sensors within the IMU. Therefore there needs to be a way for the quadcopter to determine if it undergoing a constant velocity

translation. All of the features discussed are the reasons why a custom platform needed to be developed for use in this thesis.

2.3 Quadcopter Indoor Navigation

Indoor navigation can be done by measuring the relative distance to objects in the environment from sensors that are on-board. Sensors like laser range finders (LRF) and RGBD systems like the Xbox Kinect are used. The use of the LRF can be seen in [38], where the technique used is called SLAM (simultanious localization and mapping). The downfall of using a LRF is that the distance data is on a single plane and when a quadcopter tilts there can be a spike in the distance data from the LRF. To reduce the error in the distance, the data can be mixed with gyro data to be corrected. The LRFs require lots of power and are heavy, which can be a problem when implemented on a quadcopter of the size shown in this thesis.

Vicon vision systems have been used to orient and control quadcopters while flying indoors. This method is very accurate to position a quadcopter (50 μm at 375 Hz), which is more than adequate [39]. A system like this requires the user to do a pre-flight setup of a camera system, where the quadcopter is going to be flown. This is a problem if the user wants to use to quadcopter for surveillance or search and rescue.

Using cameras that are on-board a quadcopter proves to be the better option. The Parrot AR.Drone takes advantage of the on-board camera and uses it to determine odometry values in the x and y directions for horizontal movement [40]. Using these types of systems can eliminate the limitations that are inherent to the systems previously described.

The other option for indoor navigation is to use vision chips instead of a typical camera [38, 41–44]. Vision chips consume low power and have the ability to be configured for different lighting and the resolution of the images collected can be varied.

Vision chips also consume vary little power, 0.46 mW for the Ardueye Aphid. This system has the greatest potential to be used on-board a quadcopter for indoor navigation because of the ability to reduce image size, using gray scale instead of RGB, and the low power consumption.

2.4 Optic Flow State-of-the-Art

Optic flow has been around since the late 70's, but it has gained quite a bit of interest for the application to UGVs and UAVs. The earlier work with quadcopters and optic flow has been done with the use of computer mouse sensors [31–33] for optic flow sensing. These little cameras, such as the ADNS-6010 are found in older PC mice. They are cheap, abundant, and process the images from the camera on-board, this means that the odometry values are calculated on the ADNS-6010. The post processing on the ADNS-6010 frees up a lot of processing power on the control board, which allows it to maintain a fast control loop. The downfall of the ADNS-6010 sensor is the lack of light within indoor environments, these sensors are better suited for outdoor flight where the lighting is brighter. Furthermore the ADNS-6010 lacks a usable lens, so a custom optic needs to be fitted and that is easier said than done. Other research groups have employed the use of different types of cameras to reduce the dependency on light and distance.

The type of cameras typically used are Complementary Metal Oxide Silicon (CMOS) and charge coupled device (CCD). Honegger et al developed an open source/open hardware platform using a CMOS camera that is downward facing, which is used to compute the horizontal optical flow and a sonar sensor that is used to monitor the altitude [25]. This combination of camera and sonar can be seen used by [26, 27]. A single camera placed at the bottom of a quadcopter has also been used by various researchers to determine horizontal movement similar to the research groups previously

mentioned [28–30, 45]. This method of optic flow can be done so all of the image processing is on-board the quadcopter.

Stereo vision can be used when horizontal and vertical motion is needed [17, 18]. The downfall of using stereo vision is that it usually requires a large processor and is computationally heavy. This is because stereo vision compares four separate images, these include past and present images of the right and left cameras. Stereo vision turns out to take about four times longer and high resolution is needed. Processing multiple large images on-board a quadcopter requires the use of powerful processors, which can add weight and consume lots of power [19]. Power consumption and weight are of the utmost importance when designing a quadcopter.

To avoid processing the images on-board a quadcopter, would be to use a wireless video feed to a ground station. When the images are sent to the ground station they are then processed and then sent back to the quadcopter for controlling translations [20–24]. Wireless transmission of video back and forth from a ground station to the quadcopter can be challenging because of slow transfer rates and a delay from the time the image is taken to the time it is used in the quadcopter poses a problem. The use of a ground station would limit the range of the quadcopter to the range of the signal, which would also be limited by the walls within a building, which reduce the signal strength. For these reasons on-board processing of images should be further explored.

On-board image processing with a vision chip for imaging has been explored by Burrows [46]. They have experimented with a Blade mCX RTF, which is a coaxial helicopter and various camera configurations on-board. A vision chip is similar to a CCD or CMOS camera in the sense that it grabs images in the same manner. The inherent difference of the vision chip is some post processing of the image is done on-board before the image is sent to the processor. Centeye first used the vision chip

on a UGV to help is navigate and avoid walls. Other uses of the vision chips by Centeye include a ring of eight cameras fixed around a helicopter to stabilize without the use of gyros or accelerometers. This configuration can be seen in action on their website [46].

2.5 Optic Flow Algorithms

There are different ways of computing the optic flow of a scene. The two types of methods for determining optic flow are feature-based and gradient based [47,48]. The feature-based method for computing optic flow is done by selecting some features in one image and then finding the same features in a consecutive image. Once the features are matched in both images the disparity between the features can be calculated. However this process of matching features from one image to the next isn't trivial [49]. This process can be very computationally heavy and can result in error because corresponding the features in a consecutive image is tough. The gradient-based method exploits the relationship of spatial and temporal gradient intensities [48]. These intensity values are the highest at the location of edges, where the pixel intensity value changes the most (ie. from light to dark). The gradient is determined by solving for the derivative in the x and y directions of the image, which is illustrated in chapter 3.

Gradient based methods of determining optic flow are Horn and Schunk [50], Srinvasan [51], Lucas and Kanade [52], and Fennema and Thompson [53]. Out of these optic flow algorithms the Srinvasan and Lucas and Kanade algorithms are explored the thesis. The two algorithms were selected for their ability to quickly and accurately computed the flow on the images with minimal processing. Furthermore the Srinvasan algorithm has the capability of determining the image rotation θ as well as the x and y optic flow in a sequence of images.

2.6 Quadcopter Controllers

There are many different control boards available for use on a quadcopter. The controller that will be needed for this thesis will need to be open source. The open source control boards are ideal for research because they are programmable, the board schematics are available, and there are typically large communities of developers. The controllers listed in Table 2.1 are all open source controllers considered in the design of the quadcopter. Some of the controllers like the AeroQuad [54], APM 2.5 [55], AutoQuad v6.6 [56], KKMulticopter [57], OpenPilot CC3D [58], and UAVP/UAVX [59] have developers that work on the software and provide support when developing on the controller, by far the APM 2.5 has the largest amount of users and numerous supporting documents on the control board. The Cirus MultiWii Lite [60], Cirus MultiWii SE [61], Paris MultiWiiCopter [62], and Quadrino ZoomFlight [63] are all based on the gyro and accelerometer sensors from the Nintendo Wii controllers. The fact that the APM 2.5 has the ability to self level, altitude hold, and position hold makes it more desirable to use. The APM 2.5 is the smallest open source control board available on the market, which makes it a good choice to be integrated into a quadcopter. Lastly the APM 2.5 can be programmed through the Arduino IDE, which makes for quick development.

2.7 Proposed Quadcopter

The quadcopter platform designed in this thesis encompass the key features for appropriate indoor flight. The frame is made of light weight aluminum, which is durable. The entire quadcopter will fit into a small backpack (frame dimensions (15×15) cm). The control board used can be easily programmed, has multiple on-board sensors, and a large community of developers.

Table 2.1: Open source quadcopter control boards

Board	Self Leveling	Altitude Hold	Position Hold
AeroQuad 32	X	X	
APM 2.5	X	X	X
AutoQuad v6.6	X	X	X
Crius All in One PRO	X	X	
Crius MiltiWii Lite	X		
Crius MultiWii SE	X	X	
Hobbyking KK2.0	X		
KKMulticopter			
OpenPilot CC3D	X		
Paris MultiwiiCopter	X	X	
Quadrino ZoomFlight	X	X	
UAVP/UAVX	X	X	X

The sensors used to help stabilize the quadcopter while hovering are vision chips. Each vision chip takes successive images to determine the movement of the quadcopter, weather it be altitude, pitch, roll, or yaw. The movement is detected by the use of optic flow, which will be explained more thoroughly in chapter 3. To be able to utilize optic flow on a quadcopter the imaging sensor needs to be small, light weight, require minimal computation time to process the images. A more detailed description of the optic flow sensors used in the thesis can be found in chapter 4.4. Since the weight of the quadcopter is important lower quality images can be used to utilize a small processor, which also consumes less power and will help to increase payload and flight times. The quality of the image can also be reduced by lowering the resolution and also using a grey scale images instead of RGB. Two cameras were selected to help control the quadcopter, one camera for yaw and altitude and the other for pitch and roll. The yaw and altitude camera is faced forward, while the pitch and roll camera is oriented downwards facing the ground.

2.8 Summary

This chapter covered the fundamentals of quadcopter locomotion, reviewed current state-of-the-art autonomous quadcopter systems, the various control boards that are available. The actual process of optic flow was briefly introduced, the image processing algorithms and optic flow process will be discussed with more depth in the next chapter.

Chapter 3

Optic Flow Sensing

This chapter is a overview of optic flow, how it works, and the different approaches used in this thesis are introduced. Then out the chapter it is discussed how optic flow can be used to control a quadcopter.

3.1 How Optic Flow Works

Optic flow is used to mimic the behavior of how bees orient and navigate themselves through various environments [51]. Applying the optic flow technique to a quadcopter is similar to a how a bee flies and avoids objects. Optic flow sensing is achieved by using a camera as a sensor. The camera takes two images, one at $time = -1$ and one at $time = 0$, then the images are compared with each other to determine if the camera went through a translation and in some cases a rotation [48]. If the camera moved then the corresponding translation or rotation will be reflected in the odometry values given by the optic flow algorithm. Optic flow is a method of determining movement in a scene, whether it be objects in the scene moving (SCMO) or a camera moving throughout a scene (MCSO). There are two other combinations, which include a stationary camera and stationary object (SCSO) and moving camera and moving object (MCMO) these combinations are not considered in this thesis. MCMO isn't considered because we are assuming that the objects in the indoor environment are

stationary, this also further simplifies the optic flow computation. The case of SCSO would mean that the quadcopter isn't moving since the camera is rigidly attached, which would mean that it is at rest and in this case the odometry values would be zero for all directions. When a object moves in a scene with a stationary camera the resulting optic flow can look like Fig. 3.1. The coffee mug displacement is determined

Figure 3.1: Example of optic flow with object moving through scene. (a) No movement. (b) Slight movement to the right with the vectors magnified by a factor of 5.

using optic flow, the movement can be detected even if the displacement isn't visible to the naked eye. The ability to determine this type of movement lies in the algorithms used to calculate the optic flow and the sensitivity of the imaging sensor. After a image is grabbed from the sensor the algorithm looks for the largest gradients in the image. The gradients are the largest where the pixel values in the image change from light to dark, which is also the location of a edge within the image. The coffee mug in Fig. 3.2 depicts the derivative of the image in the x and y direction. When there is a change from light to dark in the image the gradient is largest. Where these large

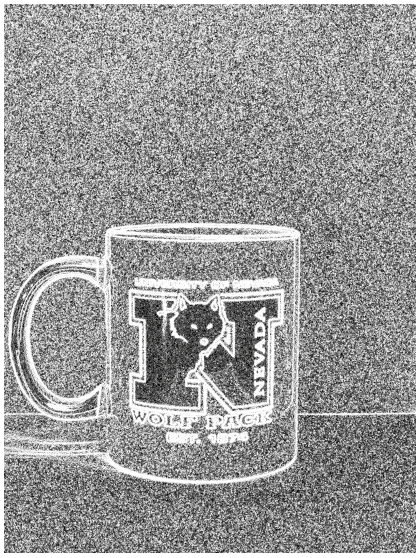


Figure 3.2: Magnitude of the dx and dy image gradient using the central difference method to determine the gradient.

gradients occur is where a line can be seen in Fig. 3.2. The derivative is calculated by the use of a kernel mask and convolving the kernel across the image [64]. The convolution of the kernel mask across a image is done using,

$$f[x,y] * g[x,y] = \sum_{(n_1=-\infty)}^{\infty} \sum_{(n_2=-\infty)}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2], \tag{3.1}$$

where $f[x,y]$ is the image matrix, $g[x,y]$ is the kernel, n_1, n_2 is the current pixel location. There are many different types of kernel masks that can be used. A example of a simple kernel mask, which is the forward deference method is,

$$dx = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

and

$$dy = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{3.3}$$

where dx is derivative in the $x$direction and dy is used to determine the derivative of a image in the y direction. The matrices presented in Eq. (3.3) are used to calculate the forward difference derivative of an image. A second method is a variation of the forward difference, which is called the backward deference method and it is,

$$dx = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix} \tag{3.4}$$

and

$$dy = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \tag{3.5}$$

where dx is used to determine the horizontal derivative and dy is used to determine the vertical derivative of a image. The matrices in Eq. (3.5) are used to calculate the backward difference derivative of an image. The last kernel mask shown is that of the central difference, which is a combination of the forward and backward deference and it is,

$$dx = \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \tag{3.6}$$

and

$$dy = \frac{1}{2} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \tag{3.7}$$

where d_x is used to determine the horizontal derivative and d_y is used to determine the vertical derivative of a image. The matrices in Eq. (3.7) are used to calculate the central difference derivative of an image. Out of Eq. (3.3, 3.5, 3.7) to calculate the

derivative the most accurate method is the central difference method because more neighboring pixels are taken into account to determine the gradient across the image. The central difference kernel is the one used to determine the derivative of the coffee mug Fig. 3.2. Convolving a kernel over a image is fairly easy when the basics of the technique are understood. A simple example is developed in Fig. 3.3 to show how to convolve a central difference kernel about an image. Figure 3.3(a,b) is a simple 3×3

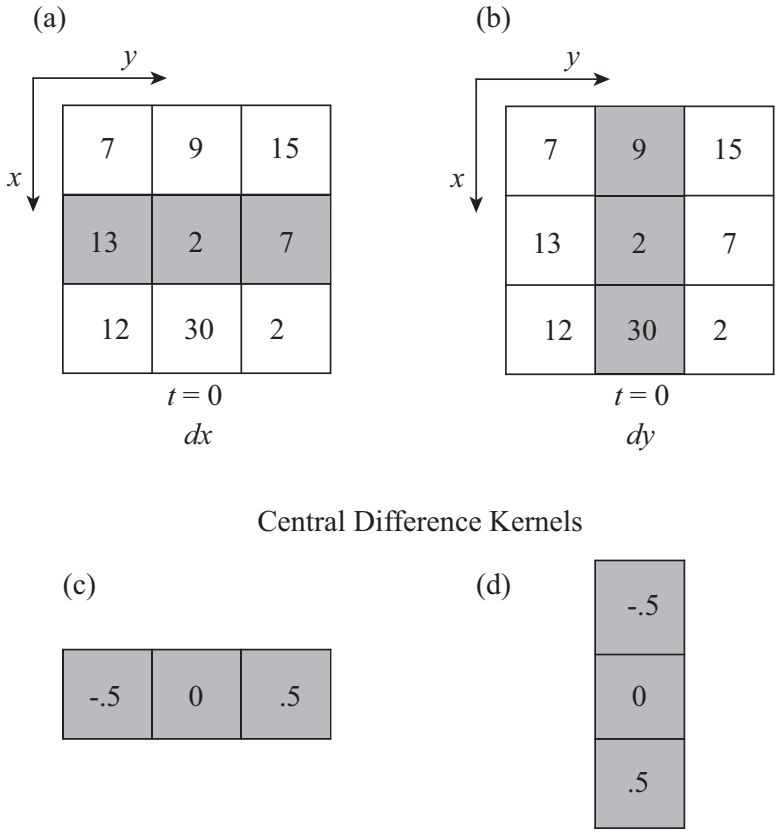


Figure 3.3: Convolving the central difference kernel with respect to the x and y directions. (a) is a 3×3 image and the dx kernel is applied to it. (b) is a 3×3 image and the dy kernel is being applied. (c) is the central difference dx kernel and (d) is the central difference dy kernel.

image and the central difference kernel for dx Fig. 3.3(c) and dy Fig. 3.3(d) are below their respective images. To apply the kernel to the image and determine dx the mask just simply needs to be multiplied by the shaded region and then summed to get the

new value of pixel(2,2). This is depicted in,

$$dx(2,2) = 13(-0.5) + 2(0) + 7(0.5) = -3, \tag{3.8}$$

$$dy(2,2) = 9(-0.5) + 2(0) + 30(0.5) = 10.5, \tag{3.9}$$

the value of $dx(2,2)$ in Eq. (3.8) and $dy(2,2)$ in Eq. (3.9) is the new value for the pixel at $(2,2)$ in the x and y direction respectively. To show these results in a more meaningful way the magnitude is calculated and shown in Fig. 3.2. The magnitude of the derivative in the x and y direction is calculated by using,

$$dxy(2,2) = \sqrt{dx^2 + dy^2}. \tag{3.10}$$

Now that the dx and dy for pixel (2,2) are established the derivative with respect to time needs to be calculated. The same idea is applied to the images, but this time there are three images needed to calculate the derivative with respect to time dt. The images needed are the present image and two past images, which are $t = -2$, $t = -1$, and $t = 0$ respectively.

$$dt(2,2) = 1(-0.5) + 2(0) + 7(0.5) = 3, \tag{3.11}$$

the value of $dt(2,2)$ is the new value for the pixel at $(2,2)$ with respect to time. Now that the various derivatives are established they are then entered into either the Lucas and Kanade [52] or Srinvasan [51] image interpolation algorithm. The Lucas and Kanade algorithm is,

$$u = \frac{-\sum(dy^2)\sum(dxdt) + \sum(dxdy)\sum(dydt)}{\sum(dx^2)\sum(dy^2) - (\sum(dxdy))^2}, \tag{3.12}$$

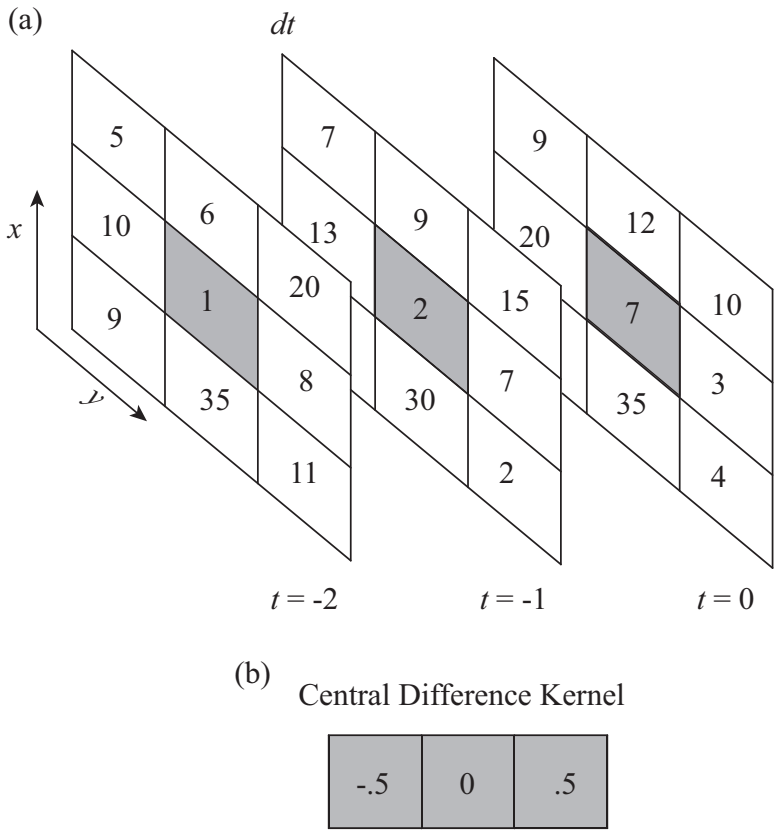


Figure 3.4: Convolving the central difference kernel with respect to time. (a) is the a 3×3 image with the dt kernel applied. (b) is the dt kernel for determining the derivative of consecutive images with respect to time.

$$v = \frac{\sum(dxdy)\sum(dydt) - \sum(dy^2)\sum(dxdt)}{\sum(dx^2)\sum(dy^2) - (\sum(dxdy))^2}, \tag{3.13}$$

where Eq. (3.12, 3.13) are the localized optic flow values in the x and y direction respectively of the image plane. These values can be used to help control a UGV or UAV and it is the method used in Fig. 3.1. There is a simpler method of determining the image derivative for the optic flow calculation, which doesn't involve applying a kernel to any of the images. The other method takes advantage of shifting the entire image left, right, up, and down by one pixel. This method was introduced by Srinivasan [51]. This method of determining the derivative is less computationally heavy, which makes it excellent to implement on a quadcopter. To use the Srinivasan technique first the window function needs to be determined Ψ, which is a 2D gaussian

shown here,

$$\Psi(x,y) = exp(-(\frac{2.772}{p^2}\cdot(x^2+y^2)), \tag{3.14}$$

where p is specified to be the half-width of the gaussian in pixels. The gaussian is a way of smoothing the image, so any noise can be reduced and the edges are more apparent in the image. To shift the images right and left by a reference amount and than to shift the image up and down by the same reference amount the following is used,

$$f_1(x,y) = f_0(x + \Delta x_{ref}), \tag{3.15}$$

$$f_2(x,y) = f_0(x - \Delta x_{ref}), \tag{3.16}$$

$$f_3(x,y) = f_0(y + \Delta y_{ref}), \tag{3.17}$$

and

$$f_4(x,y) = f_0(y - \Delta y_{ref}), \tag{3.18}$$

where the reference amount for Δx_{ref} is 1 pixel, f_0 is the original image and x and y are the pixels across the image. Equations (3.15, 3.16) shift the original image right and left by a reference amount and Eq. (3.17, 3.18), which shifts the original image up and down by a reference amount. These six reference images are developed to help determine the derivative of the image. The next step is to rotate the images by a reference amount, which should be a small angle anywhere from 0.1 deg to 10 deg. The rotations are clockwise and counterclockwise about the center of the image, the equations for doing the image rotations are Eq. (3.21, 3.22) x' and y' are the new pixel-wise location of the pixel's intensity. Once the Δx_{ref} is determined the image can be rotated using,

$$f_5(x,y) = f_0(x', y'), \tag{3.19}$$

$$f_6(x, y) = f_0(x', y'), \tag{3.20}$$

where the x' and y' values are determined by using Eq. (3.21, 3.22),

$$x' = x \cdot \cos(\Delta\theta_{ref}) + y \cdot \sin(\Delta\theta_{ref}), \tag{3.21}$$

$$y' = x \cdot \sin(\Delta\theta_{ref}) + y \cdot \cos(\Delta\theta_{ref}). \tag{3.22}$$

Once f_1, f_2, f_3, f_4, f_5, and f_6 are determined then x, y, and θ odometry values can be calculated using Eq. (3.23, 3.24, 3.25),

$$\begin{aligned} (\frac{\widehat{\Delta x}}{\Delta x_{ref}}) \iint \Psi(f_2 - f_1)^2 dxdy \\ +(\frac{\widehat{\Delta y}}{\Delta y_{ref}}) \iint \Psi(f_4 - f_3)(f_2 - f_1) dxdy \\ +(\frac{\widehat{\Delta\theta}}{\Delta\theta_{ref}}) \iint \Psi(f_6 - f_5)(f_2 - f_1) dxdy \\ = 2 \iint \Psi(f - f_0)(f_2 - f_1) dxdy, \end{aligned} \tag{3.23}$$

$$\begin{aligned} (\frac{\widehat{\Delta x}}{\Delta x_{ref}}) \iint \Psi(f_2 - f_1)(f_4 - f_3) dxdy \\ +(\frac{\widehat{\Delta y}}{\Delta y_{ref}}) \iint \Psi(f_4 - f_3)^2 dxdy \\ +(\frac{\widehat{\Delta\theta}}{\Delta\theta_{ref}}) \iint \Psi(f_6 - f_5)(f_4 - f_3) dxdy \\ = 2 \iint \Psi(f - f_0)(f_4 - f_3) dxdy, \end{aligned} \tag{3.24}$$

$$
\begin{aligned}
(\frac{\widehat{\Delta x}}{\Delta x_{ref}}) \iint \Psi(f_2 - f_1)(f_6 - f_5)dxdy \\
+(\frac{\widehat{\Delta y}}{\Delta y_{ref}}) \iint \Psi(f_4 - f_3)(f_6 - f_5)dxdy \\
+(\frac{\widehat{\Delta \theta}}{\Delta \theta_{ref}}) \iint \Psi(f_6 - f_5)^2 dxdy \\
= 2 \iint \Psi(f - f_0)(f_6 - f_5)dxdy.
\end{aligned}
\tag{3.25}
$$

The Lucas and Kanade and Srinivasan algorithms were just presented. The Srinivasan approach proved to be a simpler technique because the image is shifted when solving for the derivative instead of convolving a kernel with an image. This method also allows for the ability to detect rotations, which can be helpful for the controlling of a quadcopter.

3.2 Summary

The Lucas and Kanade and the Srinvasan algorithms were explained thoroughly and an example of how the Lucas and Kanade optic flow algorithm was used to outline the process. The image derivative was introduced along with how to convolve a kernel across a image to calculate dx, dy, and dt.

Chapter 4

Robot Platform Design

This chapter focuses improving the issues that are inherent to a quadcopter. After the design requirements are set forth the quadcopter is tested to meet the specifications. The quadcopter needs to be small, light weight, strong, and be able to accommodate additional sensors.

Quadcopters have a high power consumption this is the main factor in it's performance, which causes reduced flight times. To address this a light weight design was conceived. The material selected to build the quadcopter frame needs to be durable and light weight, as well as quick and easy to manufacture. The size of the quadcopter needs to be small for indoor flight and needs to be able to negotiate through tight spaces. The size of the quadcopter is ultimately dictated by it's payload capacity, which is determined by the rotor size. The payload capacity of the quadcopter in this thesis is required to be 0.4 Kg, which will allow for the addition of cameras for surveillance or any other sensors to monitor the quadcopter's environment. The following sections discuss the design requirements with more depth and how the final quadcopter testing platform meets the requirements.

4.1 The Quadcopter Design

To start the design process of the quadcopter the minimum components for it to fly were selected. These minimum components include a control board, battery, four motors, four ESCs, a radio receiver, 4 propellers, and the main quadcopter structure. These are all depicted in Fig. 4.1, with the optic flow sensors and the protective ring. The extra components added to the quadcopter each serve an essential function. The optic flow sensors are used to help control the altitude, yaw, pitch, and roll. The front optic flow sensor is used to detect the yaw and altitude, while the bottom optic flow sensor is used to help control the pitch and roll of the quadcopter. Lastly the protective outer ring is used to allow the quadcopter to bump into objects without crashing and it is added protection for the rotors if the quadcopter has an impact with the ground.

The first consideration was how small the quadcopter could be manufactured and still be useful for carrying small payloads and indoor navigation. The determination of this factor lies in the rotor size. The rotors need to provide enough thrust to lift the quadcopter with a small payload. The GWS 5030 props proved to be the best choice. This is due to the fact that the maximum thrust provided by each propeller is about 200 g.

The GWS 5030 far exceeded the total weight of the proposed quadcopter components without the frame, which added up to 291 g. Table 4.1 show the mass breakdown of the quadcopter. The quadcopter in this thesis will allow a for a payload capacity of 0.4 Kg.

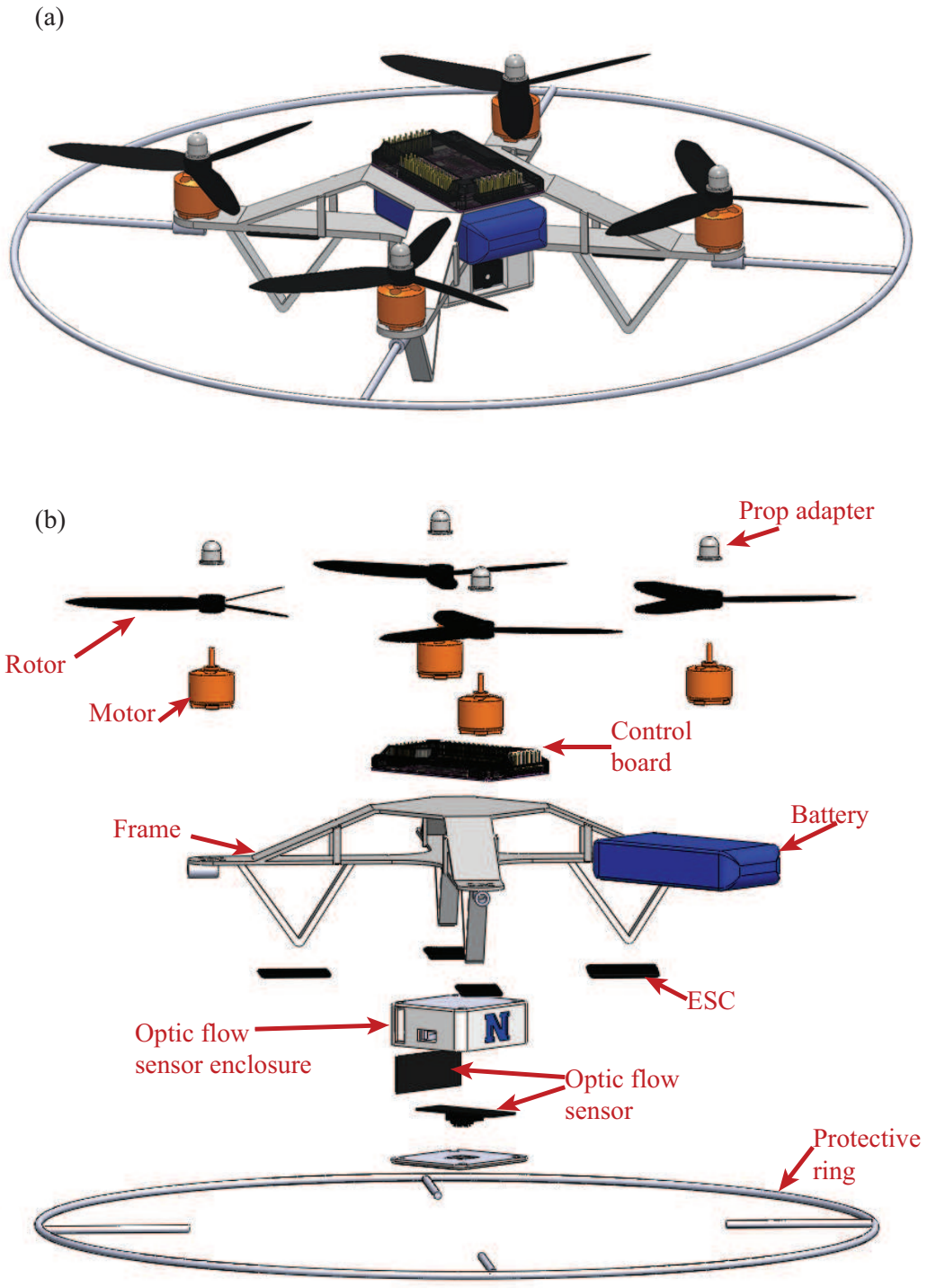


Figure 4.1: Solidworks model of the quadcopter. (a) Isometric view of the quadcopter. (b) Exploded view of the quadcopter depicting all of the components needed.

Table 4.1: Quadcopter component mass breakdown

Items	Weight (g)	Quantity	Subtotal (g)	Total (g)
Motors	17.3	4	69.2	
Prop adapters	1.4	4	5.6	
Rotors	2.3	4	9.2	
ESC	9.9	4	39.6	
Battery	87.5	1	87.5	
Optic flow sensor	4.5	2	9	
Radio receiver	5.2	1	5.2	
Frame	67	1	67	
Controller	21.8	1	21.8	
Protective ring	35.5	1	35.5	
Miscellaneous wiring	8.4	1	8.4	
				358

4.2 Robot Chassis Design

The frame needed to be designed to accommodate all of the required components and be able to survive a fall in the event that there is an accident. The obvious light weight material is carbon fiber, but it is costly and can take longer to manufacture. The next material that could be used to keep the quadcopter light is aluminum, 6061-T1 was selected for it's low cost and availability. If a stronger platform is needed 6061-T6 could be used because it's σ_{yield} is about 3.4 times higher, which allows it to survive larger impacts without undergoing any permanent deformations. Since the rotors were selected the size of the quadcopter was easy to resolve. The tip distance between the rotors is negligible because it is found that the thrust doesn't change from rotor tip interaction [65]. This meant that the rotors could be placed as close as possible without undergoing any significant loss in thrust, so the frame is designed to be (15×15) cm.

The quadcopter frame needed to be as simple as possible, so it could be easily manufactured. It was determined that the simplest design would be a flat plate, but this design quickly proved that it isn't the ideal setup to create a light and strong

frame. The impact requirement for the quadcopter is a 9.5 m/sec crash into the ground. This impact is equivalent to the quadcopter flying upside down and full thrust towards the ground from 1.5 m high. The height was determined to be 1.5 m because most hallways are roughly 3 m high and the quadcopter will be flying at a average of 1.5 m. The quadcopter was modeled to have a impact with the ground at 45 deg and land on a single arm to create to most bending stress. The average force required to stop the quadcopter in the impact described is 130 N. The average force applied to the quadcopter was determined using the impulse momentum equations. When two objects impact with each other the objects go through four phases, which are approach, compression, restitution, and separation [66]. The impulse momentum equations were found to be sufficient to model the impact because the compression phase of the impact is the portion where the ground force resists the impact and any deformation on the quadcopter will happen it will be in the compression stage.

Since the force applied to the quadcopter at impact has been determined to be 130 N a frame needed to be designed to withstand this applied load. There are two different frame configurations studied. The first frame is a flat plate of aluminum Fig. 4.2(b), which is the simplest frame design possible. The second is a frame designed similar to a bridge truss Fig. 4.3(b). The second design is made to take advantage of minimal material and increased strength.

A model to simulate the various frame designs has been developed and it takes advantage of the symmetry that is inherit to the quadcopter's design. The frame is sectioned as seen in Fig. 4.2(a). It is cut in quarters to simplify the simulation, which helps to reduce the simulation time and keeps the same attributes of a full model. The boundary conditions for the model are as follows: the cut section is symmetric, the motor mounting holes are fixed, and the force applied to the frame is the dead weight of the quadcopter. The dead weight was used to determine the areas of high stress in

the frame, which can be seen with a light force. From the simulation of the flat frame

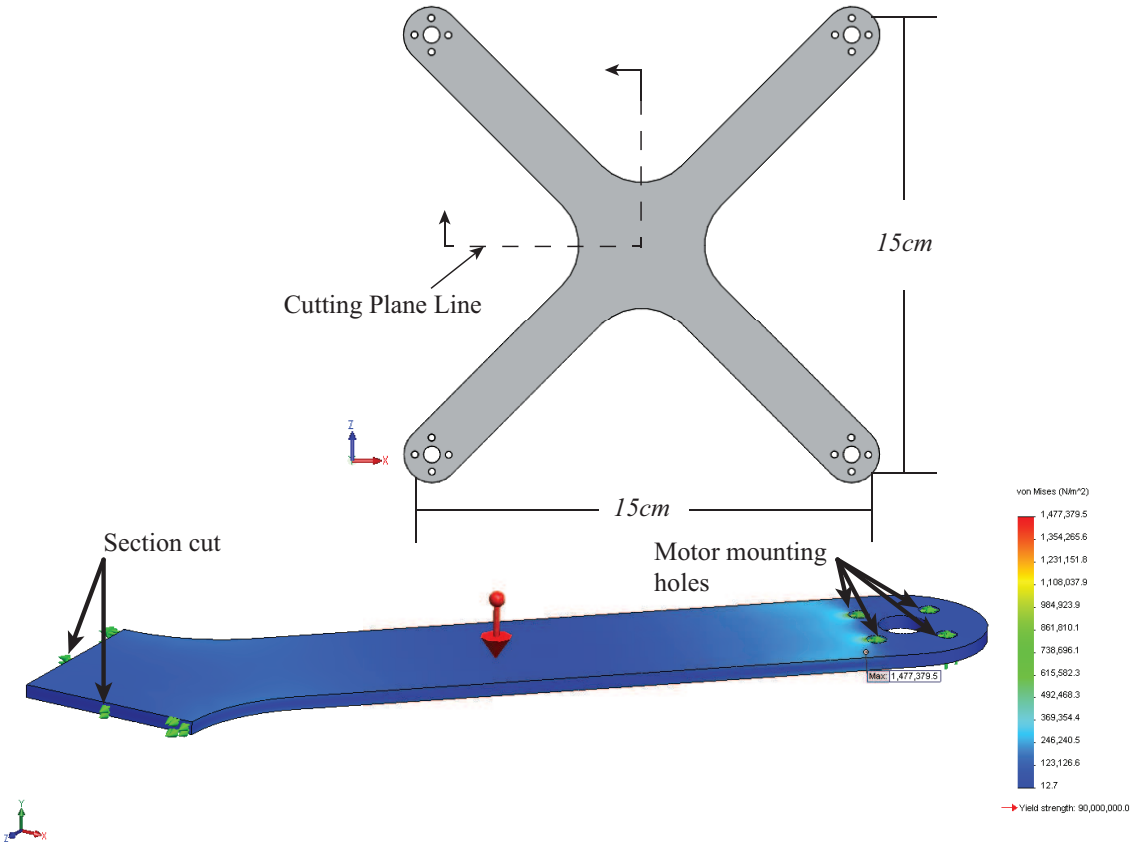


Figure 4.2: Simulation of the quadcopter flat frame design. (a) is the section cut of the frame because a 1/4 model was used in the simulation. (b) Von Mises stress for the 1/4 model of the quadcopter. The left portion of the cut frame has a symmetric boundary condition and the four holes where the motor would mount is fix. Lastly the force applied to the model is just the dead weight of the frame.

it can be seen that the max stress is 1.48 MPa, which isn't what we are concerned about because the yielding of the aluminum in the hole would not keep the quadcopter from flying. The stress we are concerned about is just on the other side of the holes that is light blue and has a value of 0.4 MPa Fig. 4.2(b). It is then determined that the maximum force that can be applied to a single arm in the simple case is 42 N. This value is far below the requirements set forth previously in the section, so a different frame design needs to be approached. The boundary conditions for the model are as follows: the cut section is symmetric, the motor mount holes are fixed, and the force

applied is the dead weight of the quadcopter, similar to the previous model. The

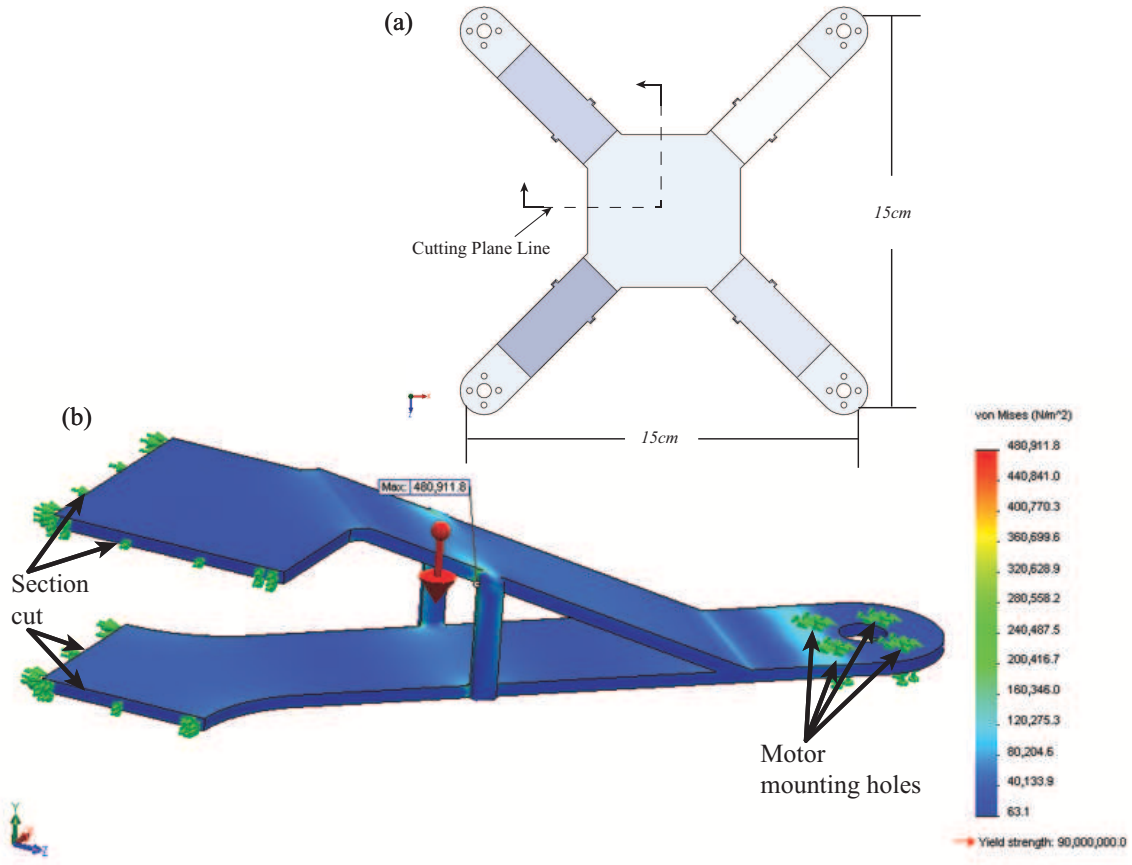


Figure 4.3: Simulation of the quadcopter redesigned frame. (a) is the section cut of the frame because a 1/4 model was used in the simulation. (b) Von Mises stress for the 1/4 model of the quadcopter. The left portion of the cut frame has a symmetric boundary condition and the four holes where the motor would mount is fix. Lastly the force applied to the model is just the dead weight of the frame.

truss design type frame design was choose for the final quadcopter frame design, it takes advantage of triangulation between two layers. The simulation was done on this frame Fig. 4.3(b) using the same boundary conditions that were used in the previous model. The frame was determined to have a stress of 0.12 MPa at the location where failure would occur in a impact with the ground. The maximum force that can be applied to a single arm of this frame is determined to be 140 N. This frame meets the requirement of being able to survive a 9.5 m/sec crash into the ground. The factor of

safety on this design is about 1.1, which is fine because if the quadcopter experiences a crash this great there is no way the rotors will survive, unless it lands upright and in this case the force would be more evenly distributed throughout all of the quadcopter legs and the force would be applied to each arm evenly.

4.3 Control Board

A control board had to be chosen that would accommodate the optic flow sensors and be able to control the motion of the quadcopter. To be able to control the quadcopter there needs to be a certain set of sensors. These sensors include three axis accelerometers and three axis gyroscopes, these are the bare minimum that a quadcopter need to be able to fly. Furthermore the control board needs to be small and light weight because the quadcopter frame that has been designed is only (15×15) cm. For these reasons the 3D Robotics APM 2.5 Fig. 4.4 is used. The APM is a open source platform, which makes it ideal for quick integration into most platforms. The software is based in Arduino and there is a large community that helps to improve the APM. The high level software that came with the APM 2.5 is called arducopter and isn't easily modified, so a separate code to control the quadcopter is developed. The new control loop can utilize the arducopter libraries, which makes it easier to interface with any of the sensors that are built into the APM 2.5. The library most utilized for the MPU-6000, which is the built in inertial measurement unit (IMU) and has a 3-axis accelerometer and 3-axis gyro. This sensor alone isn't able to control the quadcopter because some inherent problems that arise when trying to determine the rotations relative to the earth. To determine the angle of rotation about the pitch, roll, and yaw axis the rate rotation data from the gyro is integrated. The integration ends up having error relative to the desired horizontal position that the quadcopter needs to maintain for stable flight. There are different ways of alleviating this error,

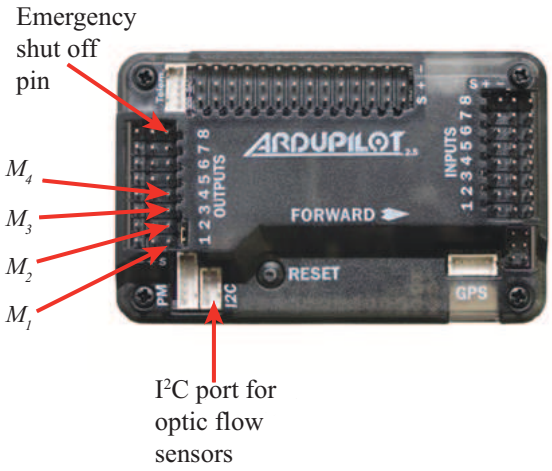


Figure 4.4: APM 2.5 control board with the main pins used to control the quadcopter are illustrated.

which include a Kalman filter and a Complementary filter , which is

$$angle = (\rho)\cdot(g) + (1-\rho)\cdot(a), \tag{4.1}$$

where *angle* is the corrected angle based on the accelerometer and gyro data, g is the integrated data from the gyro rate rotations, a is the angle based on the accelerometers, and ρ is the percentage that will be used of the accelerometer and gyro data. The method that is used in this thesis is the complementary filter. The complementary filter Eq. (4.1) mixes the accelerometer and gyro data by only using a percentage of data from the gyro and accelerometer, which helps to reduce the error. Mixing of the accelerometer and gyro data by it's self isn't enough to help control a quadcopter because the gyro scope still has a lot of drifting error. This error can be alleviated by using a high pass filter, which will only let short duration signals pass through, while filtering out signals that are steady over time. This helps to eliminate

the error that can build up in the gyro data. The high pass filter is,

$$HP = (1 - a{\cdot}dt)y(k) + u(k+1) - u(k). \tag{4.2}$$

where, y is the current angle, u is the filtered value. Before using the high pass filter the constant α is determined, which is

$$a = \frac{\tau}{\tau + dt} \tag{4.3}$$

where α is a constant that is determined from the time frequency that the data is collected dt and the time constant τ that determines how many samples are used in the high pass filter. Once the equations are established for determining the angle of rotations about the pitch, roll, and yaw axis the control loop for the quadcopter can be developed Fig. 4.5 and Fig. 4.6. The control loop used for the quadcopter is a proportional, integral, and derivative PID. A single loop is used to control the pitch, roll, yaw, and altitude. The quadcopter is symmetric about the pitch and roll axis, which means that the PID gains developed for these axis can be the same. The

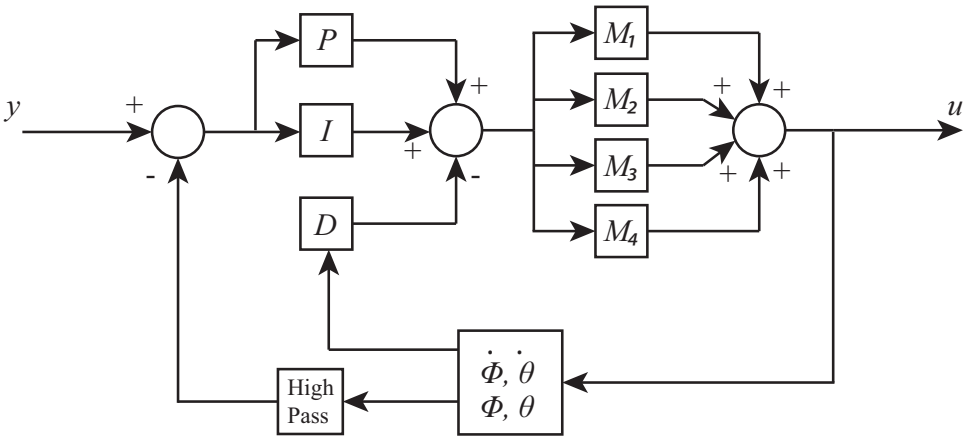


Figure 4.5: Control loop used to stabilize the quadcopter pitch and roll, where M is the motor, y is the input and u is the output of the system.

gains were determined experimentally using the Ziegler Nichols method. The Ziegler

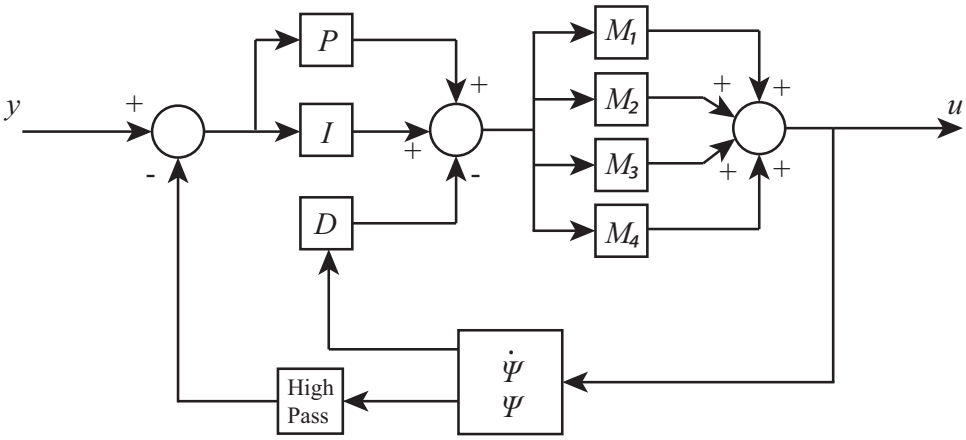


Figure 4.6: The control loop used to stabilize the quadcopter yaw, where M is the motor, y is the input and u is the output of the system.

Nichols method is a good way of determining the initial values for each axis and then the values can be fine tuned to achieve the correct response, which is discussed more in chapter 6.

4.4 Optic Flow Sensors

Optic flow is used on the quadcopter to help control the pitch, roll, yaw, and altitude. The pitch and roll need an additional sensor to help in controlling these axis because the when the quadcopter is trying to stabilize it can undergo a constant velocity horizontal motion that the accelerometers wouldn't be able to monitor. The accelerometers monitor accelerations and not velocity and the gyros measure angular rate. For these reasons the quadcopter can have 0 pitch and roll, while undergoing a constant velocity translation.

The correct optic flow sensors needed to be selected to be fitted on the quadcopter. These sensors needed to collect and process the video feed as fast as possible to minimize any error that can arise from integrating the odometry values from the

optic flow algorithm. The optic flow sensor was first integrated into the APM 2.5 by having the APM 2.5 collect the images and processes them on board. This proved to be too slow because the images were collected and processed at 15 Hz. This slowed down the entire control loop on the quadcopter as well, which caused the quadcopter to have a undesirable response. Another method of sensing the optic flow needed to be determined. The perfect sensor for determining optic flow was determined to be the Ardueye Aphid 4.7 by Centeye. This sensor can be programmed to gather and

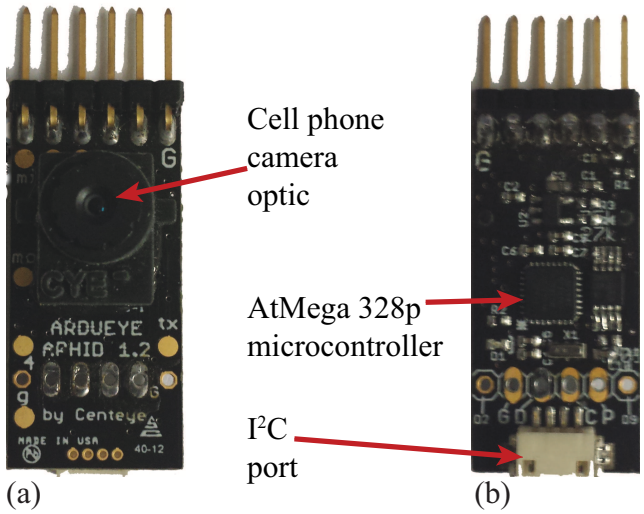


Figure 4.7: The Ardueye Aphid sensor from Centeye. (a) Front of Ardueye Aphid sensor showing the cell phone optic. (b) The back of the optic flow sensor, which shows the external ADC and the ATmega 328p processor.

process the images on board, which will free up the APM 2.5. This sensor is able to determine x and y odometry values at 250 Hz. The data is then sent over I^2C to the APM 2.5. This is the idea setup for the quadcopter because multiple sensors can be connected to the same line. The seamless integration of the optic flow sensors into the control board will make it easy to utilize the sensors to their full potential. The final version on the quadcopter testing platform is in Fig. 4.8. The components that make up the testing platform for the optic flow sensors are illustrated. This testing platform has two optic flow sensors. One sensor is facing forward, so the

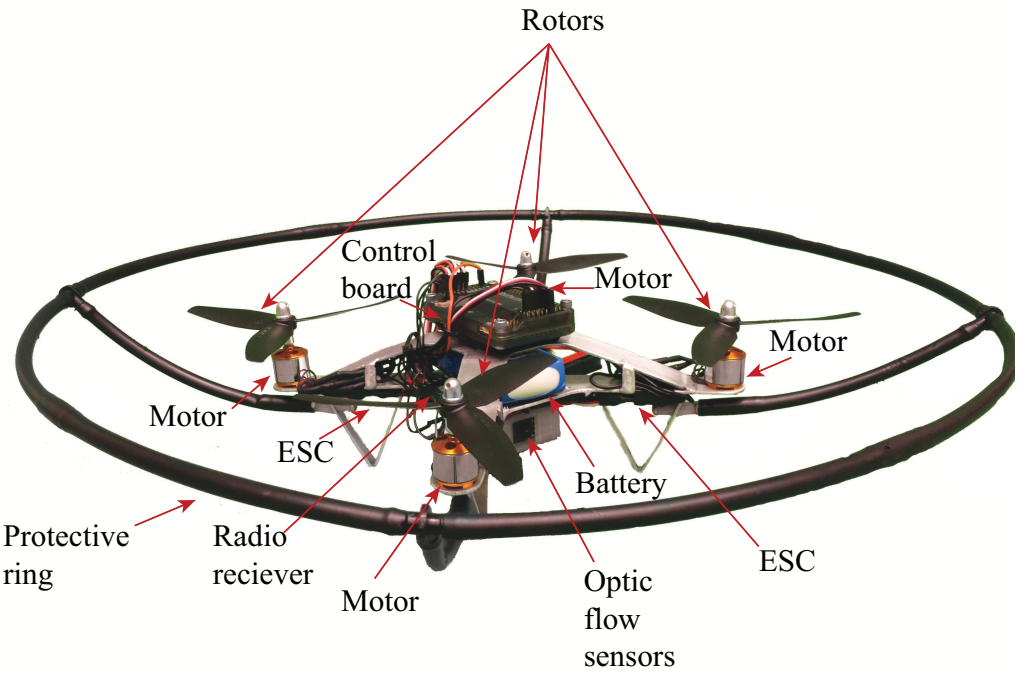


Figure 4.8: A custom built quadcopter platform designed and built for testing in this thesis. The components that make up the quadcopter are shown and illustrated above. There are a few items hidden because of the viewing angle, which include the third and fourth Electronic Speed Controller (ESC) and the second optic flow sensor.

optic flow for vertical and about the yaw axis can be measured. This information will be integrated into the control loop to stabilize the quadcopter while it is being tested. The second optic flow sensor faces downwards. The downward facing sensor can obtain the horizontal motion of the quadcopter, which is related to the pitch and roll. The optic flow information from the downwards facing optic flow sensor is used to control the pitch and roll of the quadcopter aerial vehicle. This will keep the quadcopter from drifting in the horizontal direction. The quadcopter is fitted with a 1000 mAh battery, but it is designed to accommodate a slightly bigger battery if longer flight than 10 mins is desired.

4.5 Summary

The specifications for the design of the quadcopter was explained in this chapter. It has been determined that the quadcopter meets the requirements set forth and can survive a crash from 1.5 m high at full thrust towards the ground. The optic flow sensors chose for this thesis were also described and the characteristics of the optic flow sensors were discussed.

Chapter 5

Experimental Setup

Specific testing apparatuses were created to test the different aspects of a quadcopter. The first setup is for the optic flow algorithm. Then a three degree a freedom test stand was developed for testing the pitch, roll, and yaw. Then the optic flow sensors on the quadcopter needed to be tested, so a test stand to measure the location of the quadcopter with respect to a image was created. Lastly a test was done with the quadcopter to test vertical and yaw rotation optic flow control. All of the different test setups and procedures are described in the following sections.

5.1 Optic Flow Algorithm Test Setup

The quadcopter has been designed and built to utilize optic flow sensing. The next step is to determine if the optic flow algorithm is working correctly, tune the quadcopter's PID, test the tracking error of the optic flow sensors, and test the quadcopter with the combination of vertical takeoff and landing as well as yaw control. The optic flow was tested first by building a gimbal with a optic flow sensor to track a object. The gimbal is depicted in Fig. 5.1. The gimbal has two servos that allow the optic flow sensor to track. One servo is used for vertical motion and the other is to track horizontal motion. The gimbal is designed to track a object that is moving in front of the optic flow sensor. Any object can be tracked in front of the gimbal as long as there

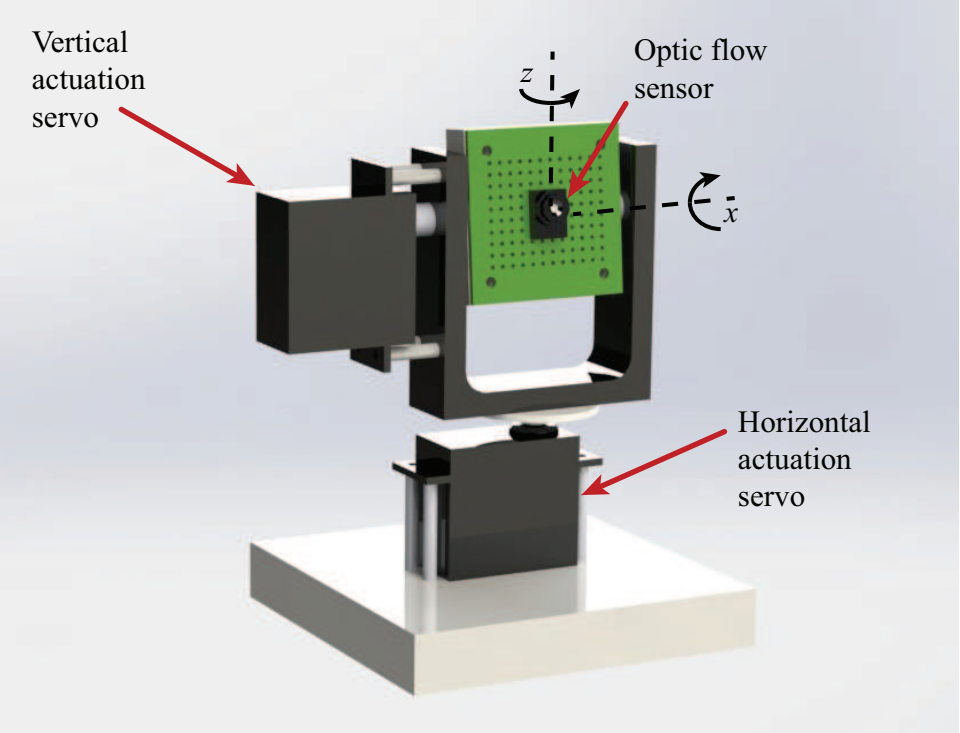


Figure 5.1: Solidworks model of the gimbal used to test optic flow algorithms. The gimbal can rotate about the z-axis to track in the horizontal direction and rotate about the x-axis to vertically track objects.

is a high contrast between the object and the background. This essentially means that if the object is white the background can not be white to be able to track the object. The ideal situation would be if the background was white the object would be black. The gimbal proved to track the object well, which meant that the algorithm used in the gimbal control loop was able to determine the odometry movement of the object.

5.2 Roll, Pitch, and Yaw Test Setup

Since the quadcopter is complete and the control loop is designed The quadcopter needs to be tested to determine the correct gain values for the PID and test to see how the quadcopter responds to a disturbance. These test were performed on the

test stand shown in Fig. 5.2.

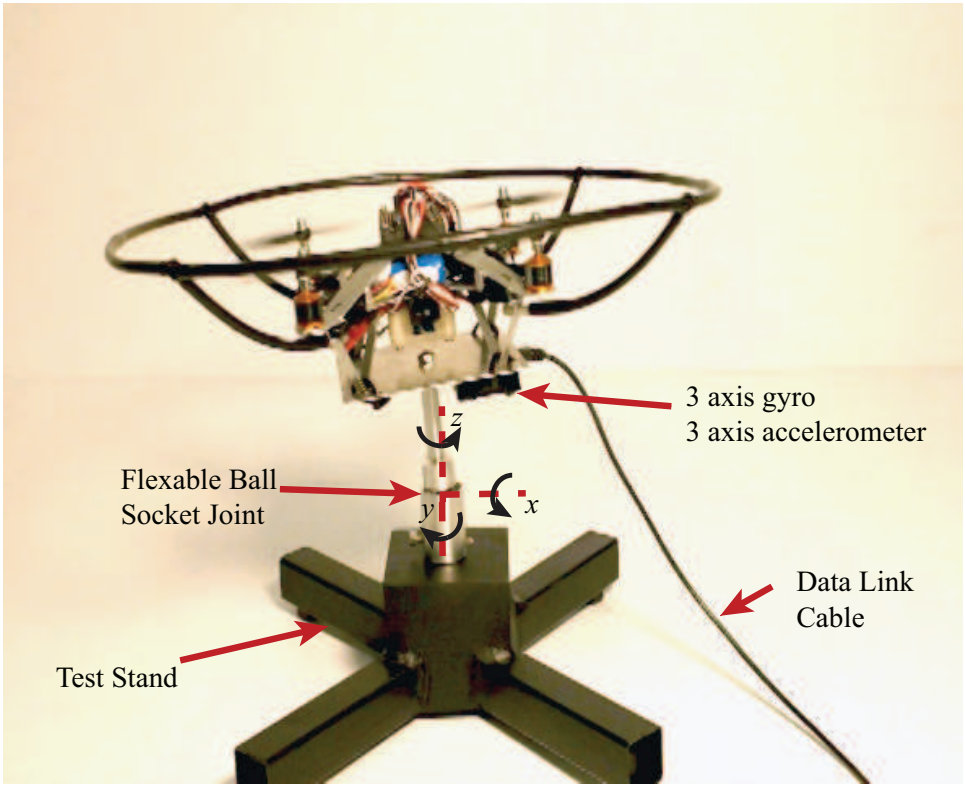


Figure 5.2: Three degree of freedom stand to test and tune the quadcopter.

5.3 Optic Flow Control of Yaw Axis Test Setup

This test stand has a flexible ball socket that allows for three degrees of freedom. The quadcopter is allowed to rotate about the pitch, roll, and yaw axis. The out side of the ball joint has a cylinder that constrains the motion to plus and minus five degrees in all directions except the yaw axis. The sensor used to monitor the movement of the quadcopter is a three axis accelerometer and three axis gyroscope. The data from this sensor is being collected at 27 Hz and that allows for a sufficient amount of data to be collected to monitor. There needed to be a way to determine the tracking error of the optic flow in a controlled manner. To measure the movement

of the quadcopter with respect to a reference image the test stand in Fig. 5.3. The test stand is designed allow the quadcopter to rotate freely about the yaw axis while tracking the image that has an array of University of Nevada logos printed on it.

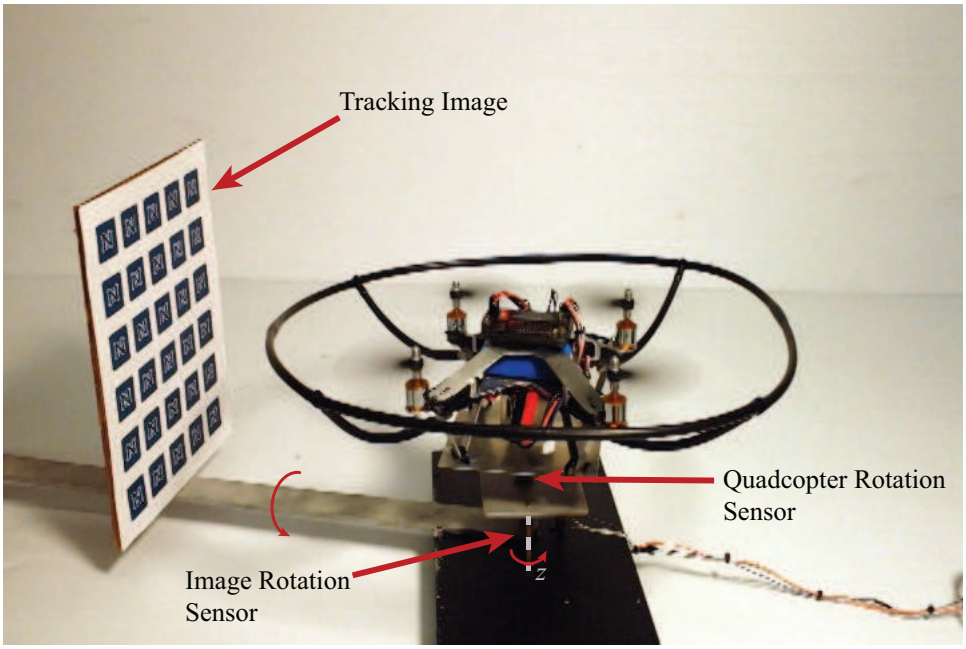


Figure 5.3: The setup that is used characterizes the tracking error of the optic flow sensor about the yaw axis of the quadcopter.

5.4 Optic Flow Control of Yaw Axis and Hover Test Setup

The quadcopter yaw is monitored using a potentiometer and since the image rotates about the same axis the angle is monitored with a separate potentiometer. The voltage in each potentiometer is first zeroed out when the quadcopter and the image are at the starting angle. The quadcopter and image is at 0 deg initially. Once the quadcopter turns on the image is rotated to 45 deg and then back to 0 deg. The last test procedure done on the quadcopter using optic flow is testing vertical take off

and landing combined with yaw rotation. The test setup used to monitor the vertical movement and yaw rotation is shown in Fig. 5.4. The quadcopter starts in the down

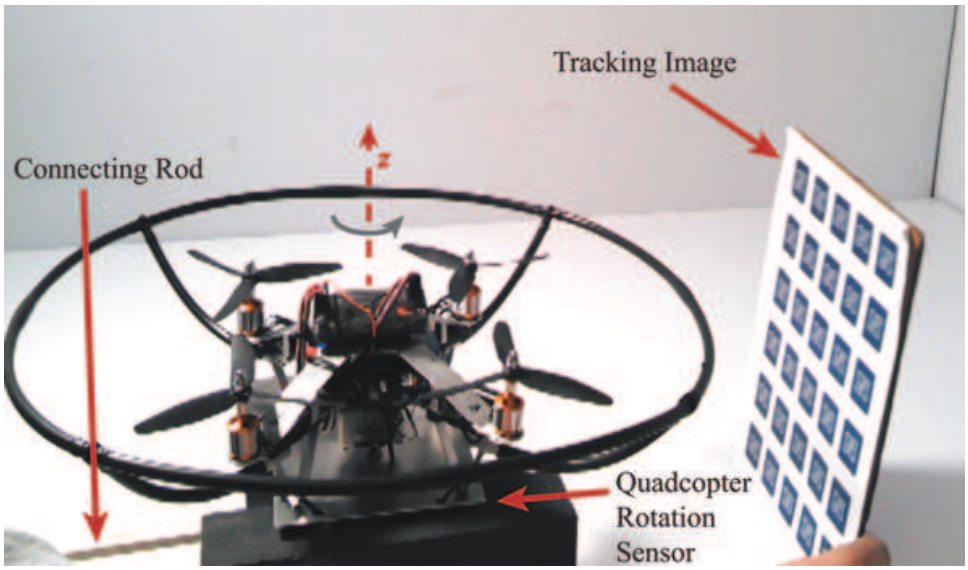


Figure 5.4: Quadcopter attached to a rod to allow vertical motion and rotation about the yaw axis.

position and then when it turns on the image is then moved upwards to a height of about five inches, then the image is rotated to about 30 deg. Once the image reaches the 30 deg angle the process is reversed, so the quadcopter will come to a rest where it initially started. The height of the quadcopter is measured with a potentiometer that is attached to the opposite side of the wooden rod. The potentiometer that is attached to the bottom of the quadcopter is used to monitor the yaw angle of the quadcopter. The results of these tests are presented and explained in the following chapter.

5.5 Summary

The various test setups used for the quadcopter to quantify it's performance with and without optic flow were discussed. The various sensors used for the testing of the

quadcopter platform were also discussed. The testing methods for the quadcopter proved to be sufficient for the results that were desired.

Chapter 6

Experimental Results and Discussion

The quadcopter was tested to determine if it was being controlled by the control board correctly. These tests were designed to allow the maximum amount of movement and to be able to monitor the movement of the quadcopter. The first round of testS were used to determine the correct gains for the quadcopter. The test setup allows for motion to be restricted in the pitch and roll direction, but the yaw axis has 360 degs of rotation. The first tests done on the quadcopter were used to determine what gains are required to have a stable system. The values for the gains were determined

Table 6.1: Ziegler Nichols tuning rule based on critical gain and critical period

Type of Controller	K_p	T_i	T_d
P	$0.5K_u$	∞	0
PI	$0.45K_u$	$\frac{1}{1.2}P_u$	0
PID	$0.6K_u$	$0.5P_u$	$0.125P_u$

using the Ziegler Nichols tuning method as seen in Table 6.1, which is a great tuning procedure when the transfer function of the system is not known [67]. There are two different approaches when using the Ziegler Nichols method. The first approach is done by experimentally applying a unit-step input and measuring the response of the plant. The first method wasn't used because it isn't possible to get a stable response

from just a proportional controller. The second method is done by setting

$$T_i = \infty \tag{6.1}$$

and

$$T_d = 0, \tag{6.2}$$

where T_i is the value for the integrator and T_d is the value for the derivative portion of the controller. Then by only using a proportional control increase the gain until the system exhibits sustained oscillations, this gain is known as the ultimate gain K_u. Once the K_u is determined the correct proportional gain can be calculated, which is

$$P = 0.5 \cdot K_u, \tag{6.3}$$

where K_u is the ultimate gain as described earlier and P is the gain of the controller if only a proportional controller is desired.

6.1 Roll, Pitch, and Yaw Testing

A test was done on the quadcopter to determine if the system is stable with a proportional gain the results of the test are in Fig. 6.1.

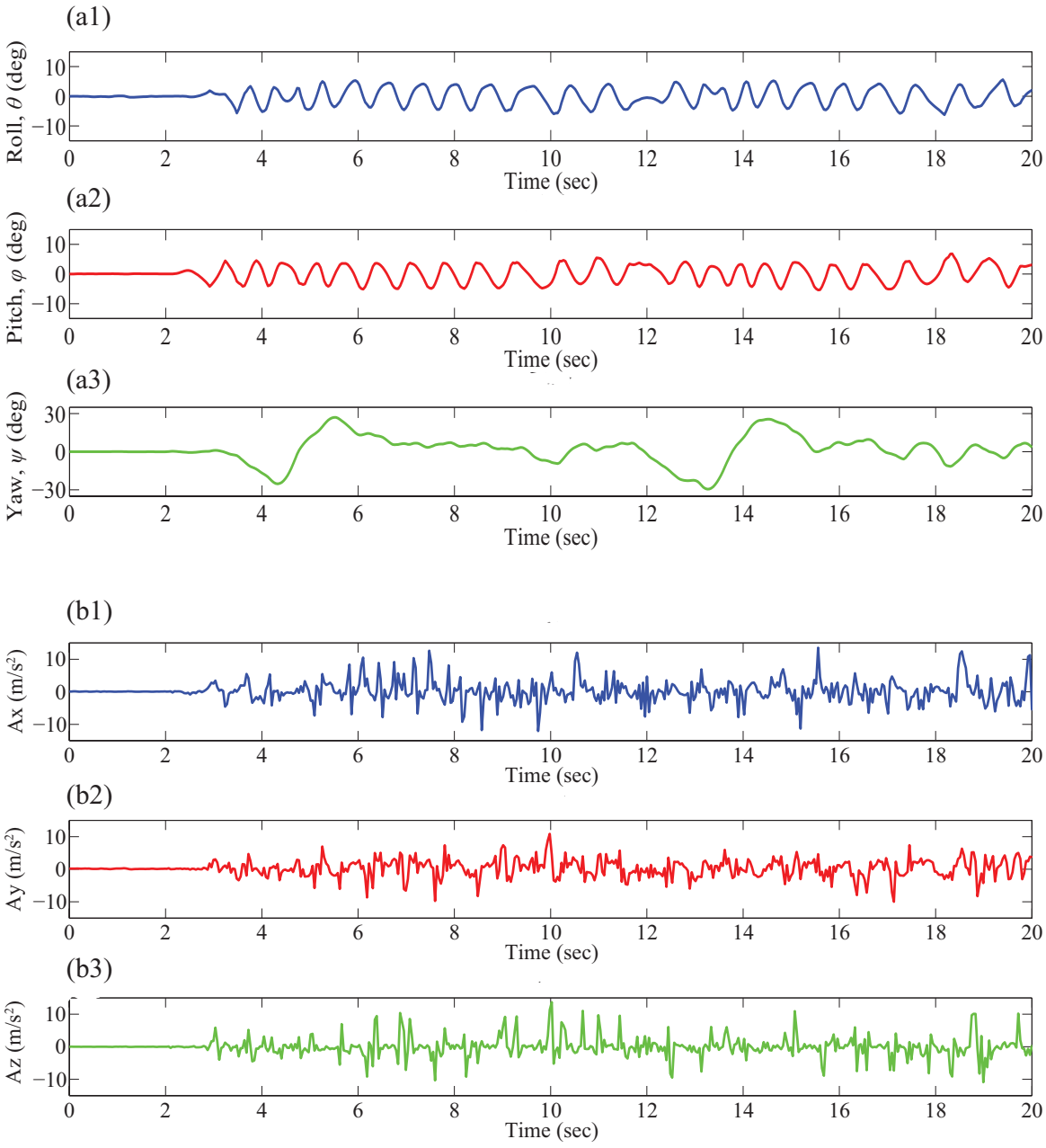


Figure 6.1: Testing stable gains with only a proportional (P) gain. The first set of three graphs shows the angle of the quadcopter about the roll(a1), pitch(a2), and yaw(a3) axis respectively. The second set of three graphs depicts the acceleration in the x(b1), y(b2), and z(b3) direction of the quadcopter.

The gain for the quadcopter was adjusted from the initial value and a stable response still wasn't able to be obtained with just a proportional controller. Next a integral was added to the quadcopter control loop. The PI controller was developed,

$$P = 0.45 \cdot K_u \tag{6.4}$$

and

$$I = 1.2 \cdot K_p / P_u, \tag{6.5}$$

where P_u is the period that the quadcopter's response at K_u. The response of the quadcopter with just a PI controller can be seen in Fig. 6.2.

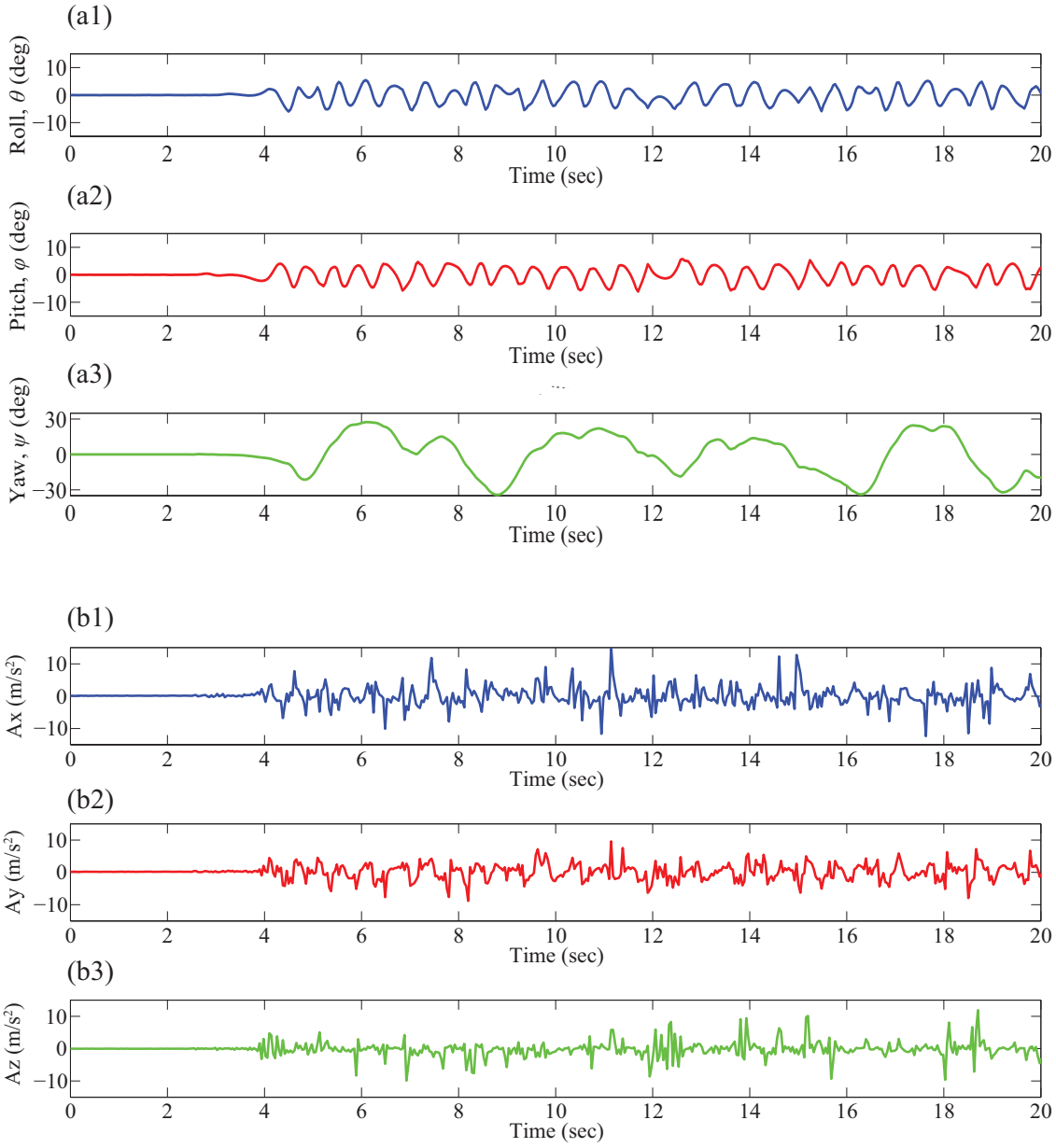


Figure 6.2: Testing stable gains with a proportional and integral (PI) gains. The first set of three graphs shows the angle of the quadcopter about the roll(a1), pitch(a2), and yaw(a3) axis respectively. The second set of three graphs depicts the acceleration in the x(b1), y(b2), and z(b3) direction of the quadcopter.

The PI controller wasn't able to produce a stable response either, as seen in Fig. 6.2. Since the quadcopter can not produce a stable response with a PI controller a PID controller was developed. The derivative in the PID controller will be better suited for the quadcopter because the rate of the rotation will be monitored. To develop the PID controller the following values were calculated:

$$P = 0.6 \cdot K_u, \tag{6.6}$$

$$I = 2 * K_p / P_u, \tag{6.7}$$

and

$$D = P_u / 8 \cdot K_p. \tag{6.8}$$

The response of the quadcopter with the PID controller can be seen in Fig. 6.3.

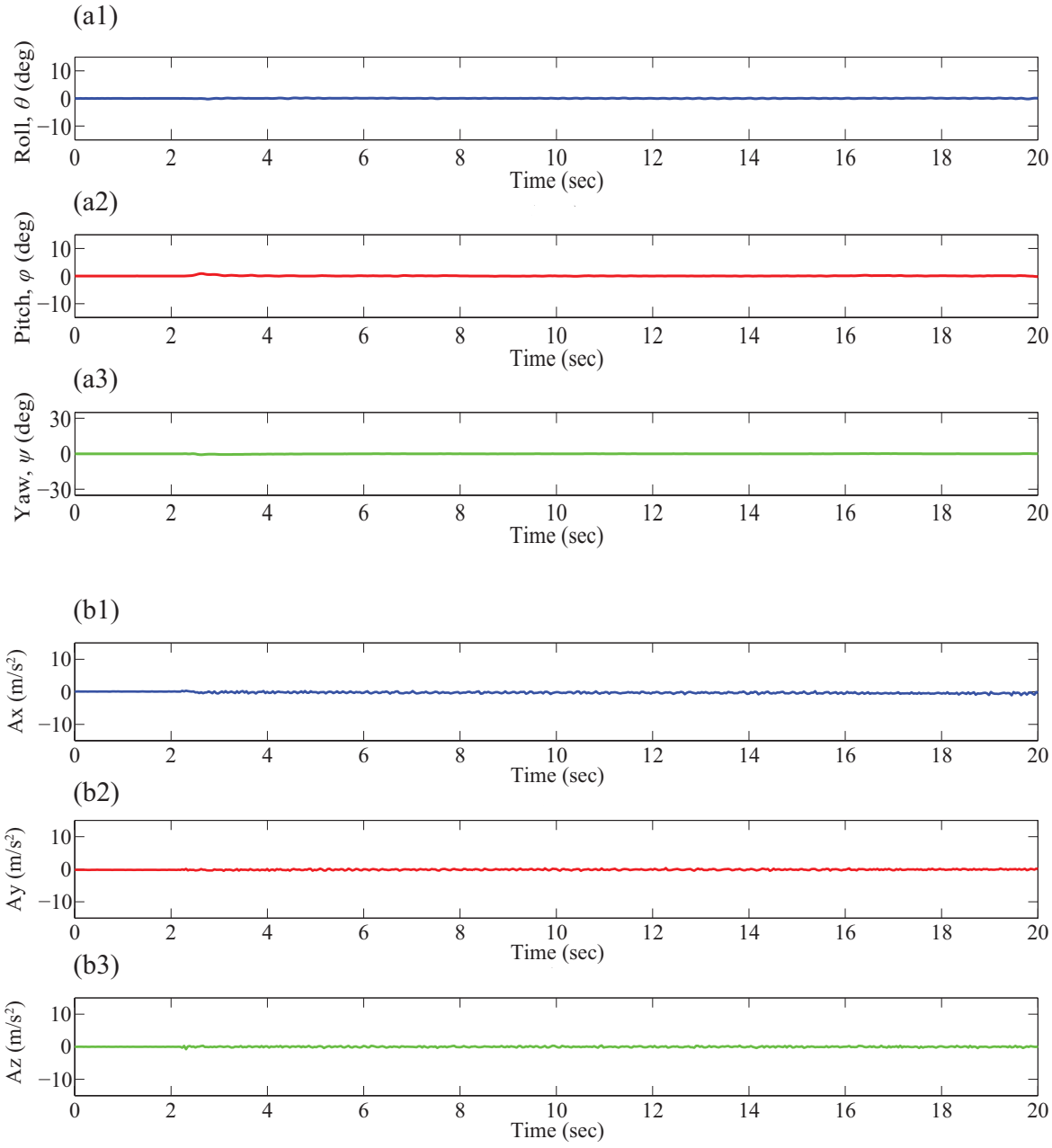


Figure 6.3: Testing stable gains with a proportional, integral, and derivative (PID) gains. The first set of three graphs shows the angle of the quadcopter about the roll(a1), pitch(a2), and yaw(a3) axis respectively. The second set of three graphs depicts the acceleration in the x(b1), y(b2), and z(b3) direction of the quadcopter.

The P, PI, and PID data can be compared by calculating the f_{max} and the f_{RMS}. The f_{max} is calculated from,

$$f_{max} = \underset{\forall t}{max} \left| f(t) \right|, \tag{6.9}$$

where $f(t)$ is the the data set of interest. The RMS value of the data can be calculated with,

$$f_{RMS} = \sqrt{\frac{1}{T} \int_0^T \left| f(t) \right|^2 dt} \tag{6.10}$$

The data collected from the quadcopter as it is hovering in place shows that the roll,

Table 6.2: P, PI, and PID analysis

		P	PI	PID
Roll,θ	Max	6.27	6.00	0.24
	RMS	3.11	3.02	0.07
Pitch,ϕ	Max	6.82	6.12	0.92
	RMS	3.16	3.00	0.16
Yaw,ψ	Max	31.00	35.16	0.82
	RMS	15.06	19.20	0.16
Ax	Max	13.82	15.41	1.69
	RMS	4.00	3.27	0.42
Ay	Max	10.86	10.35	0.51
	RMS	3.05	2.74	0.21
Az	Max	13.70	11.90	0.91
	RMS	3.35	2.70	0.19

pitch, and yaw are all stable in hover. The quadcopter next was tested to determine if the gain values that were selected for the test above give the most idea response.

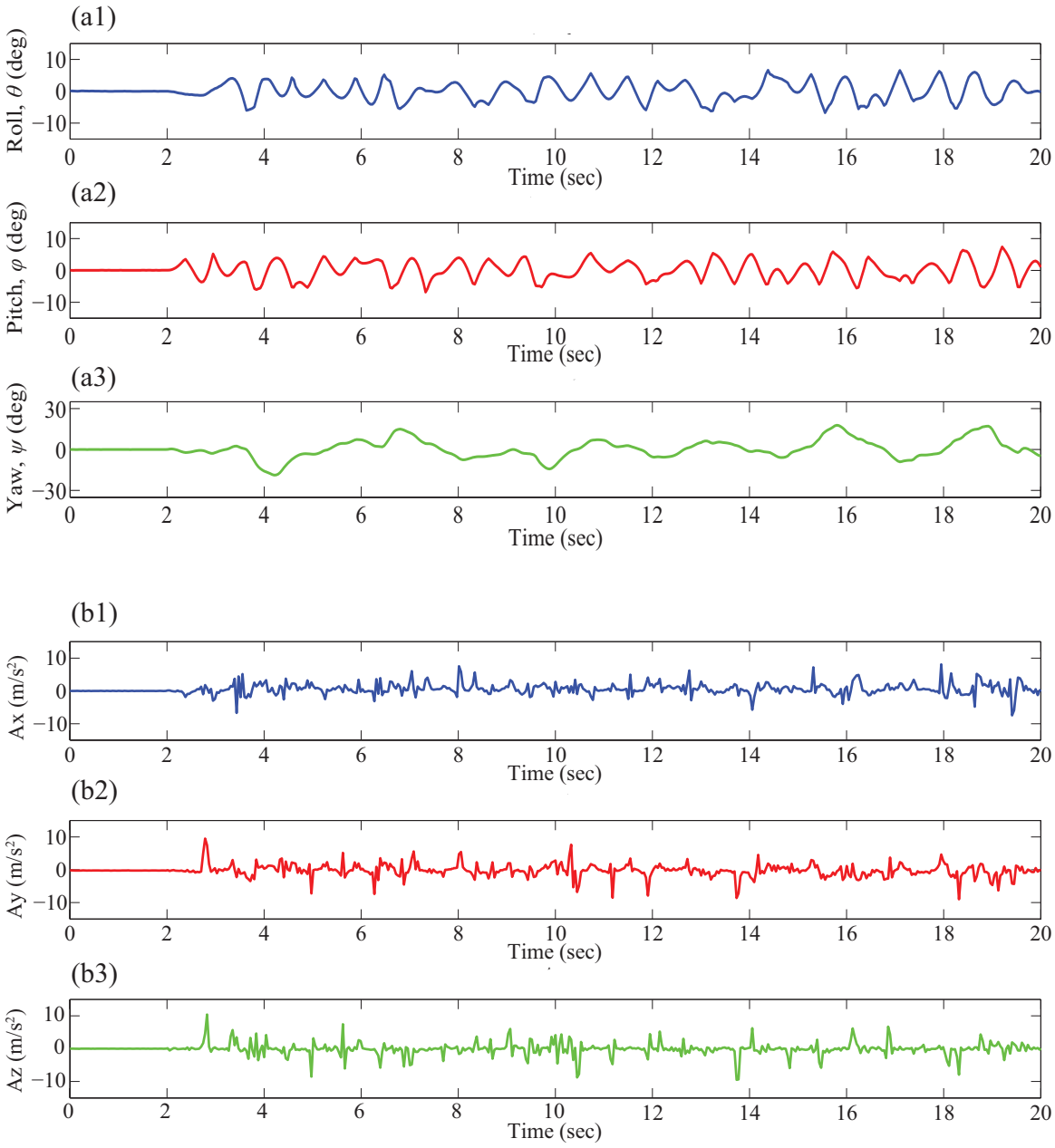


Figure 6.4: The behavior of a quadcopter with PID gain values that result in a unstable system. The first set of three graphs shows the angle of the quadcopter about the roll(a1), pitch(a2), and yaw(a3) axis respectively. The second set of three graphs depicts the acceleration in the x(b1), y(b2), and z(b3) direction of the quadcopter.

Once the quadcopter is turned on it started with zero roll, zero pitch, zero yaw, and it is in the up right position. When the quadcopter is let go it immediately goes unstable and constantly bounces around. The yaw has a maximum overshoot of about 15 deg and it bounces between plus and minus 15 deg. Next the quadcopter's gains were lowered to show a marginally stable system Fig. 6.5.

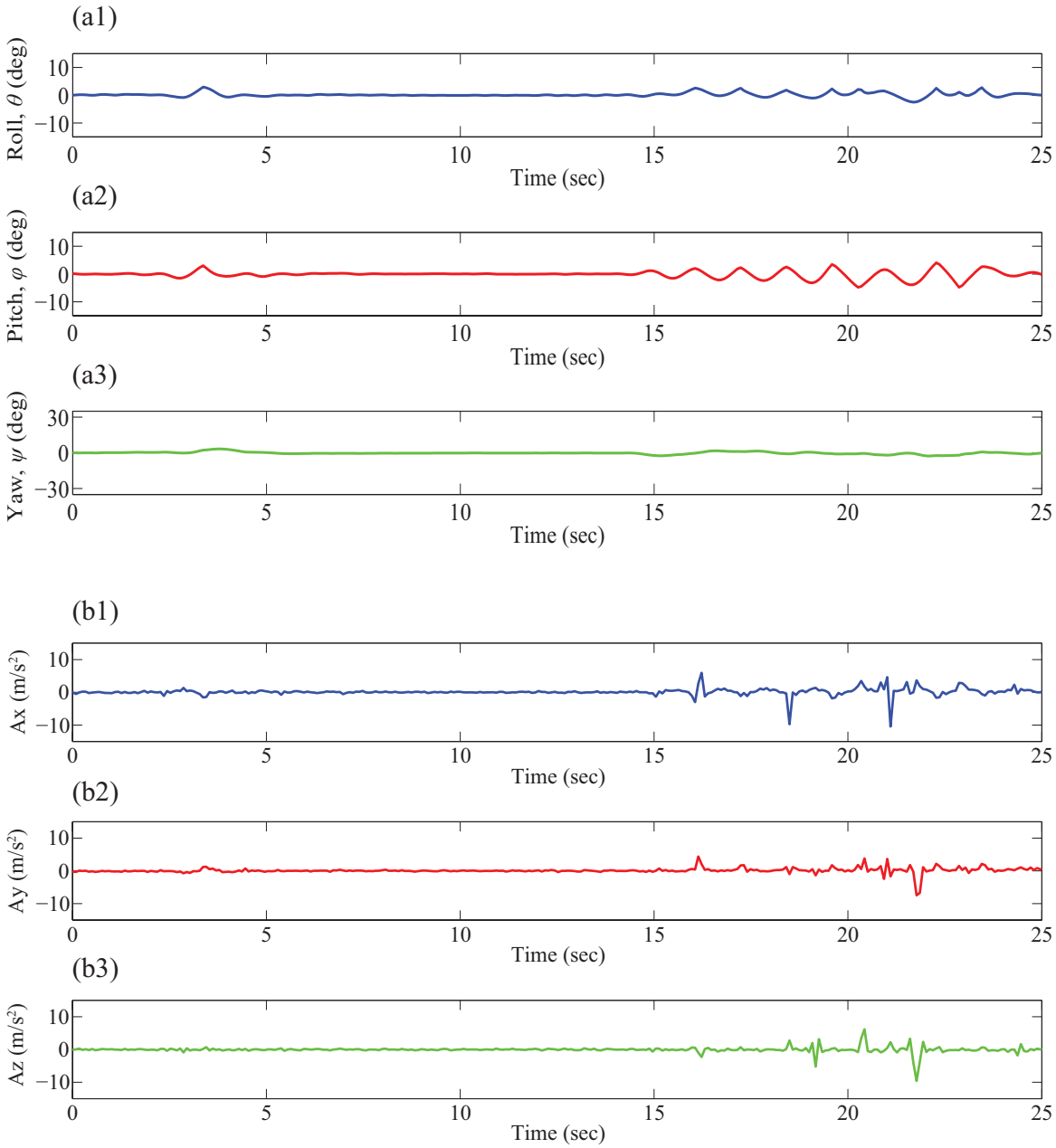


Figure 6.5: The behavior of a quadcopter with PID gain values that result in a marginally stable system. The first set of three graphs shows the angle of the quadcopter about the roll(a1), pitch(a2), and yaw(a3) axis respectively. The second set of three graphs depicts the acceleration in the x(b1), y(b2), and z(b3) direction of the quadcopter.

If a system is marginally stable as shown in Fig. 6.5 the quadcopter bounces around the zero for roll, pitch, and yaw. The angles never seem to get too high and it stabilizes for a second of two and then oscillates again. The last set of test of this type are for stable PID gains. Shown in Fig. 6.6 the when the quadcopter is hovering under it's own power it doesn't have any overshoot. The quadcopter can stabilize about the roll, pitch, and yaw axis.

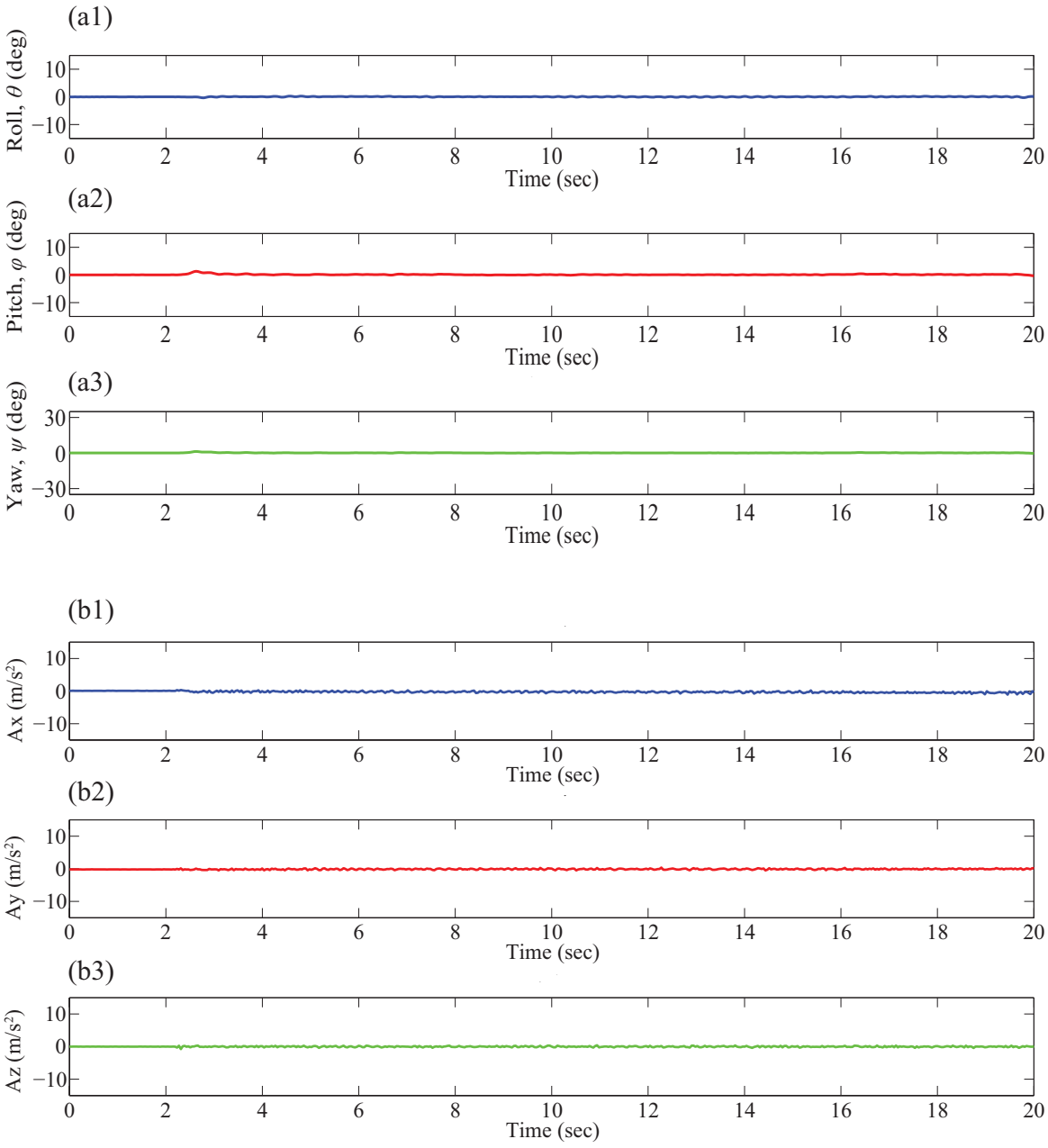


Figure 6.6: The behavior of a quadcopter with PID gain values that result in a stable system. The first set of three graphs shows the angle of the quadcopter about the roll(a1), pitch(a2), and yaw(a3) axis respectively. The second set of three graphs depicts the acceleration in the x(b1), y(b2), and z(b3) direction of the quadcopter.

Table 6.3: Unstable, marginally stable, and stable quadcopter analysis

		Unstable	Marginally stable	Stable
Roll,θ	Max	6.80	3.00	0.34
	RMS	2.70	0.61	0.10
Pitch,ϕ	Max	7.40	4.80	1.30
	RMS	2.70	1.01	0.23
Yaw,ψ	Max	18.80	3.30	1.16
	RMS	6.50	0.80	0.23
Ax	Max	8.10	10.40	1.69
	RMS	1.80	0.98	0.42
Ay	Max	9.50	7.40	0.51
	RMS	1.90	0.72	0.21
Az	Max	10.40	9.60	0.91
	RMS	1.90	0.93	0.19

The values for the gains in Fig. 6.3 and Fig. 6.6 are the same because these values were found to produce the most stable system, which can be seen in the small RMS values. These are the best values for this particular setup. Next the quadcopter is tested while hovering in place and a 5 deg disturbance is applied. The first test has the same conditions that are used in the above tests. The results from the disturbance can be seen in Fig. 6.7.

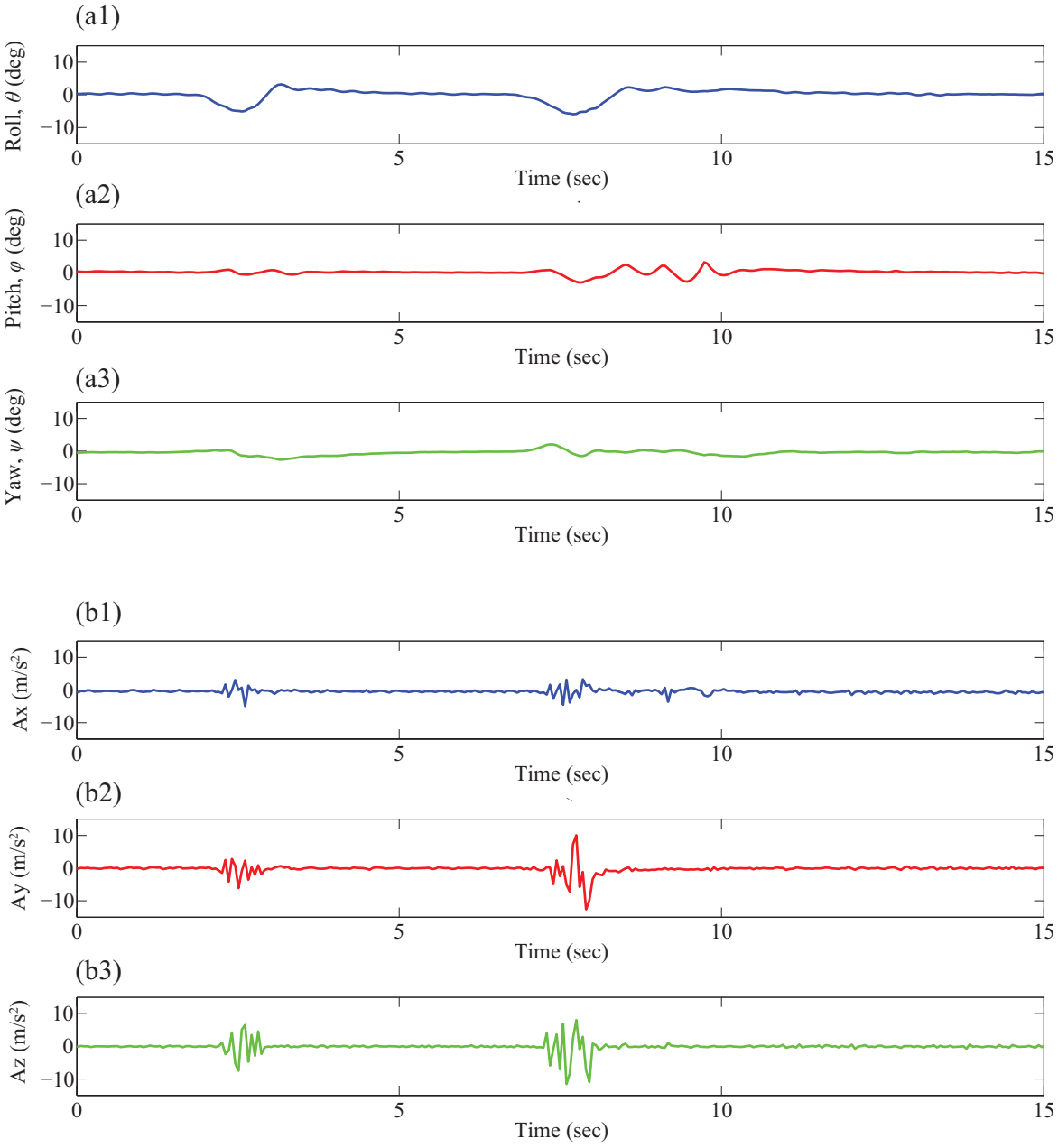


Figure 6.7: Testing the quadcopter with a 5 deg disturbance. The first set of three graphs shows the angle of the quadcopter about the roll(a1), pitch(a2), and yaw(a3) axis respectively. The second set of three graphs depicts the acceleration in the x(b1), y(b2), and z(b3) direction of the quadcopter.

The quadcopter was able to recover from the 5 deg disturbance. The quadcopter responds with 2.5 deg of overshoot and settles to 0 deg within 1.5 sec of the initial release. The height of the quadcopter test stand was then adjusted to be 5 cm longer then the previous test. Since the length of the rod on the test stand is extended the gains needed to be adjusted, so the quadcopter can respond faster and that way it can accommodate for the increased mass of the longer rod. The results of this test can be seen in Fig. 6.8.

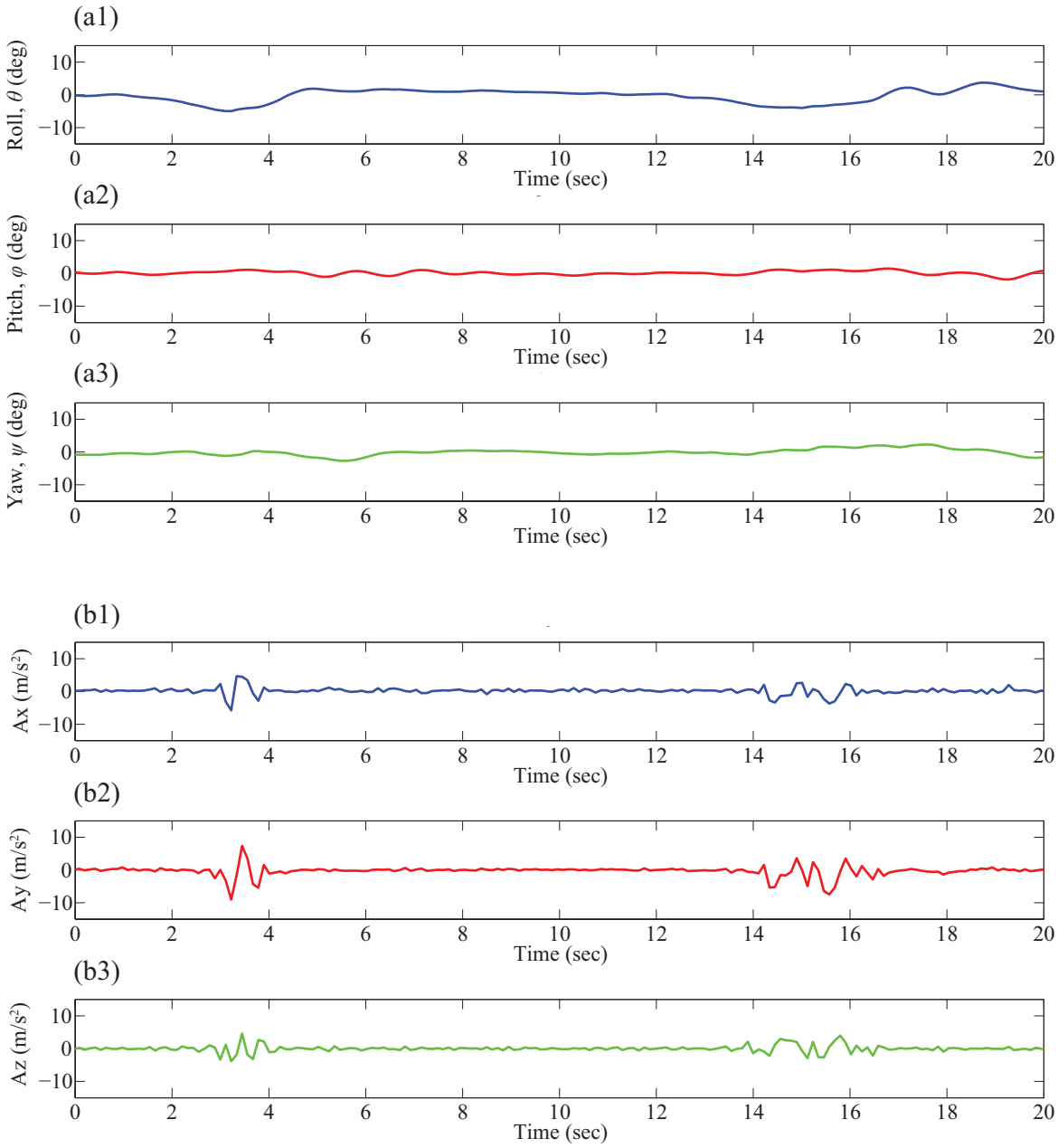


Figure 6.8: Testing the quadcopter with a longer rod that connects the quadcopter to the flexible ball joint. Once the quadcopter is stable it is subjected to a 5 deg disturbance. The first set of three graphs shows the angle of the quadcopter about the roll(a1), pitch(a2), and yaw(a3) axis respectively. The second set of three graphs depicts the acceleration in the x(b1), y(b2), and z(b3) direction of the quadcopter.

Table 6.4: Analysis of quadcopter with varying test stand rod length

		Short Rod	Long Rod
Roll,θ	Max	5.94	5.23
	RMS	1.23	1.63
Pitch,ϕ	Max	3.26	5.40
	RMS	0.60	1.18
Yaw,ψ	Max	2.60	6.07
	RMS	0.62	1.35
Ax	Max	4.92	12.40
	RMS	0.70	1.58
Ay	Max	12.61	9.02
	RMS	1.14	1.27
Az	Max	11.54	4.60
	RMS	1.28	0.74

With the longer rod attached to the quadcopter it responded slower (4 seconds) to the disturbance and a little bit more overshoot (3 deg). The slower response is the result for the longer rod in the test setup and the increased overshoot is due to the higher gains that were needed to actuate the quadcopter. The results of these tests are great for determining if the sensors and the control loop is working on the quadcopter. The downfall is that the test setup doesn't replicate the motion of the quadcopter in free flight. The stand moves the center of gravity down closer to the base and the quadcopter isn't allowed to rotate about the center of mass, which makes it hard to control the hovering. Furthermore the test stand increases inertia and only a portion of the thrust is being allowed to help rotate the quadcopter up right, which can limit the mass that the quadcopter can lift while being tested.

To better quantify the movement of the quadcopter and get a better understanding of the quadcopter in the previous tests. The pitch, roll, and yaw is combined by,

$$f_{mag,max} = \sqrt{(f_{max,\theta})^2 + (f_{max,\phi})^2 + (f_{max,\psi})^2}\ , \tag{6.11}$$

and

$$f_{mag,RMS} = \sqrt{(f_{RMS,\theta})^2 + (f_{RMS,\phi})^2 + (f_{RMS,\psi})^2} \ , \tag{6.12}$$

where $f_{mag,max}$ and $f_{mag,RMS}$ are the magnitudes for the combination of the pitch, roll, and yaw. The magnitude of the accelerations are also calculated using,

$$f_{mag,max} = \sqrt{(f_{max,ax})^2 + (f_{max,ay})^2 + (f_{max,az})^2} \ , \tag{6.13}$$

and

$$f_{mag,RMS} = \sqrt{(f_{RMS,ax})^2 + (f_{RMS,ay})^2 + (f_{RMS,az})^2} \ , \tag{6.14}$$

where $f_{mag,max}$ and $f_{mag,RMS}$ are the magnitudes for the combination of the ax, ay, and az. These values make it easier to compare the movement of the quadcopter. Table 6.5 is a summary of the magnitudes for the previous max and RMS values, which were determined using Eq. (6.11,6.12). Starting with the first set of tests,

Table 6.5: Magnitude of angle for previous tests

	Max (deg)	RMS
Short rod	7.26	1.50
Long rod	9.66	2.42
Unstable	21.32	7.54
Marginally stable	6.55	1.43
Stable	1.78	0.34
P	32.35	15.70
PI	36.19	19.67
PID	1.26	0.24

which were the P, PI, and PID. It can be seen that the quadcopter is unstable with just a P or PI controller because the RMS for each of these is quite high. The ideal value of RMS for a stable quadcopter system is zero. When the derivative was added to the control loop to create a PID the RMS was reduced by about 6,000 percent. The maximum value for the PID is significantly lower, which is 1.26 deg compared

to 32 deg and 36 deg. Once the PID was established the gains were adjusted to determine the correct value to get the best stability. The unstable case is similar to the case where only a proportional controller is used. The marginally stable case has a about 420 percent decrease in the RMS error and only a peak angle of 6.5 deg. When the PID gains are lowered by 30 percent from marginally stable case to the stable case the RMS is decreased by 320 percent and has a peak value of 1.78 deg. Lastly the test stand rod was varied to show that the quadcopter is robust and it can stabilize after a disturbance even with a longer test platform. The same gain values were used for the short and long rod test and during both tests the quadcopter undergoes a disturbance. The RMS value for the long and short rod are fairly similar with the short rod RMS being 1.5 and the long rod RMS is 2.4, which is about 60 percent different from each other. The reason for the response of the quadcopter to be more unstable and have a larger overshoot has to do with the increase in the rod length. When the quadcopter tilts all of the thrust from the quadcopter rotors are not acting in completely against the tilting, which makes it hard for the quadcopter to correct the motion. For this reason the longer rod makes it harder for the quadcopter to stabilize. From table 6.6, which shows the magnitude of the RMS and the max for

Table 6.6: Magnitude of acceleration for previous tests

	Max (m/s^2)	RMS
Short rod	17.79	1.85
Long rod	16.01	2.16
Unstable	12.89	2.96
Marginally stable	13.13	1.55
Stable	2.01	0.64
P	17.96	5.35
PI	18.88	4.57
PID	2.05	0.61

the previous tests. From the table it can be seen that the long rod is slower to respond to the disturbance because the maximum acceleration is 16.01 m/s^2 for the long rod

and is 17.79 m/s^2 for the short rod. Even though the long rod is slower to respond the short rod proves to be more stable because of the RMS data of the accelerometer and angle.

6.2 Optic Flow Control of Yaw Axis Testing

Next the optic flow was tested to determine how well it can track. The stand was configured to only test the yaw rotation and different controller configurations were used, P, PI, and PID. From the previous tests the P and PI can be ruled out for actual flight, but they were still included in the testing to have a better analysis of the optic flow sensors. All of the tests can be seen in Fig. 6.9, Fig. 6.10, Fig. 6.11.

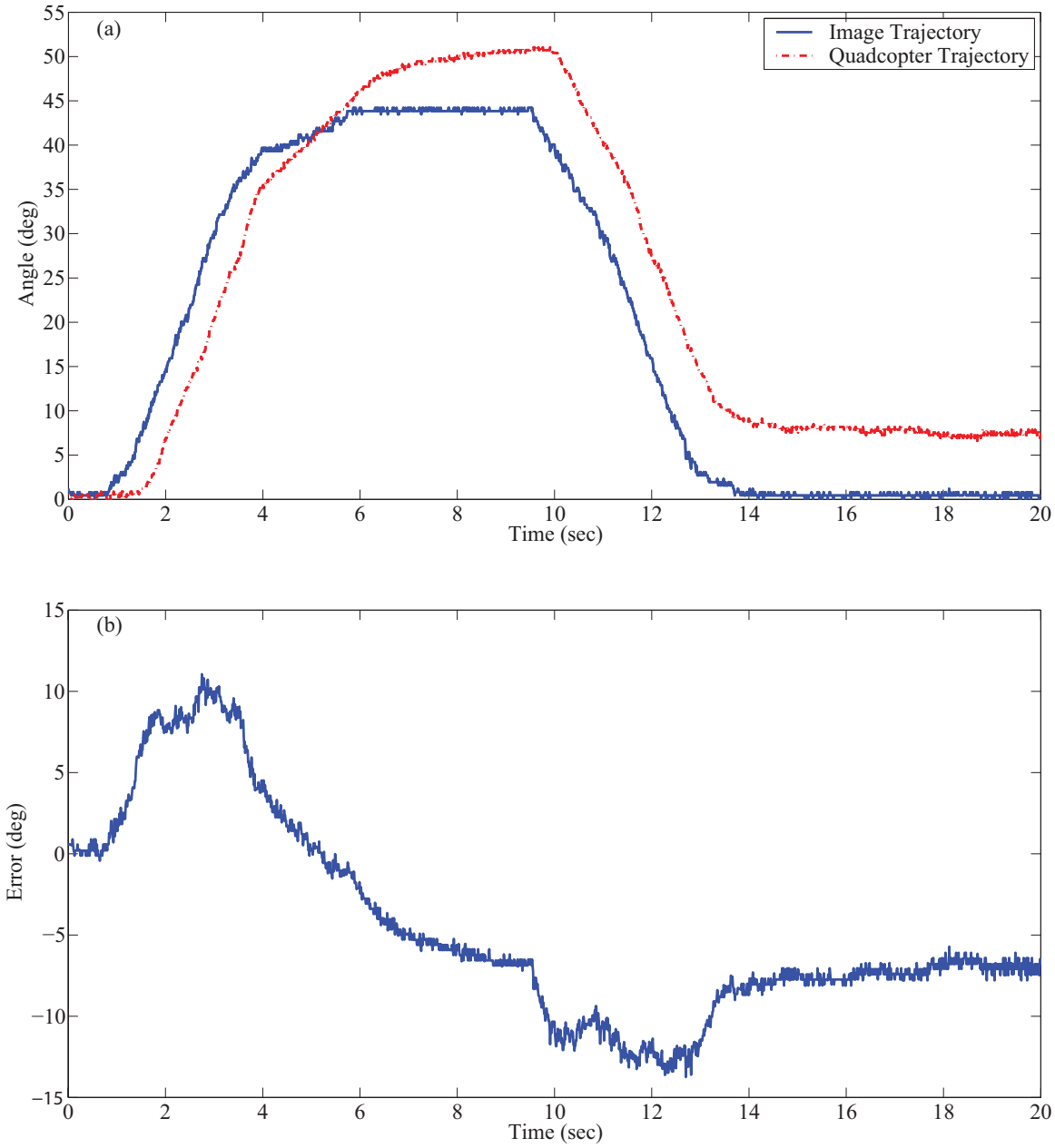


Figure 6.9: The quadcopter with only proportional controller on the yaw axis. The optic flow sensor is used to track a image, which is shown in (a). In (b) the error between the quadcopter and the desired location of the quadcopter it shown.

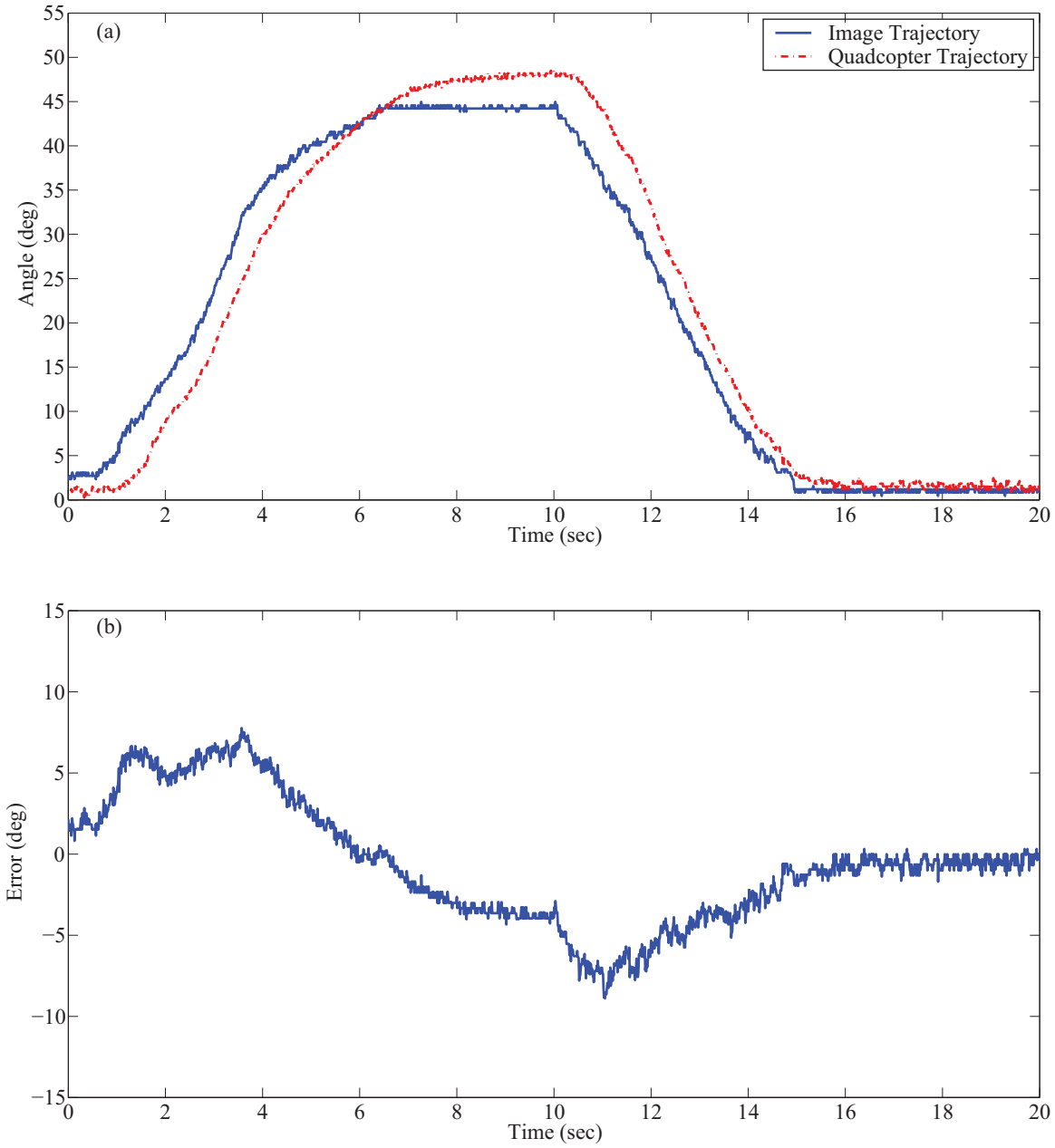


Figure 6.10: The quadcopter with only proportional integral controller on the yaw axis. The optic flow sensor is used to track a image, which is shown in (a). In (b) the error between the quadcopter and the desired location of the quadcopter it shown.

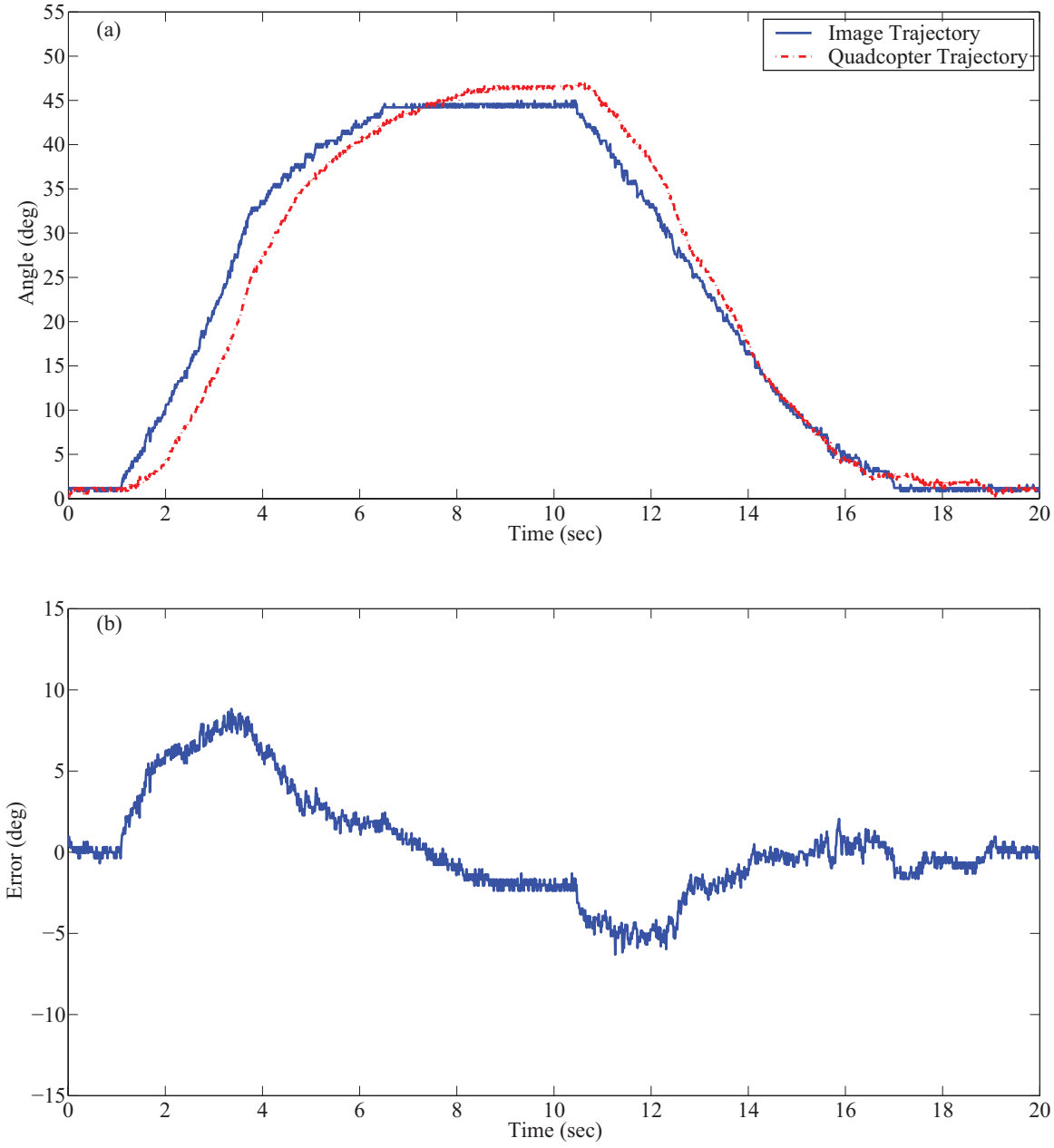


Figure 6.11: The quadcopter with only proportional integral derivative controller on the yaw axis. The optic flow sensor is used to track a image, which is shown in (a). In (b) the error between the quadcopter and the desired location of the quadcopter it shown.

In Table 6.7 the error of the P, PI, and PID control of the quadcopter is summed up. Using the data from the error plot the RMS and max is determined using Eq. (6.9, 6.10). The first thing that is noticeable is that from the P to the PI controller the error was decreased by 98 percent. Than from the PI to the PID controller the amplitude was decreased by 12 percent. The PID controller for the yaw axis proved to be the best choice to control the yaw axis using the optic flow sensors. The Ziegler

Table 6.7: Optic flow yaw data analysis

		P	PI	PID
Error	Max	13.70	8.89	7.90
	RMS	7.61	3.84	3.20

Nichols method was used to tune the gains for the optic flow tests, which is the same method used previously. The first test done was with a proportional gain to control the quadcopter. The quadcopter overshoots the final value by 5 deg and the steady state error is the same as the overshoot, which is 5 deg. When a PI was used in the quadcopter the overshoot and steady state error was reduced by half. Since in testing a PI wasn't able to reduce the overshoot enough a PID was implemented. The overshoot and the steady state error was about 2 deg, which is far better than the P and PI controller. The PID proved to be the best for a quick and accurate response given a 10 deg/sec input of the image for the quadcopter to track.

6.3 Optic Flow Control of Yaw Axis and Hover Testing

The final test done on the quadcopter was to demonstrate the ability to track a image for vertical takeoff then yaw to about 30 deg and then return to it's starting yaw angle and height. The results of this test are depicted in Fig. 6.12.

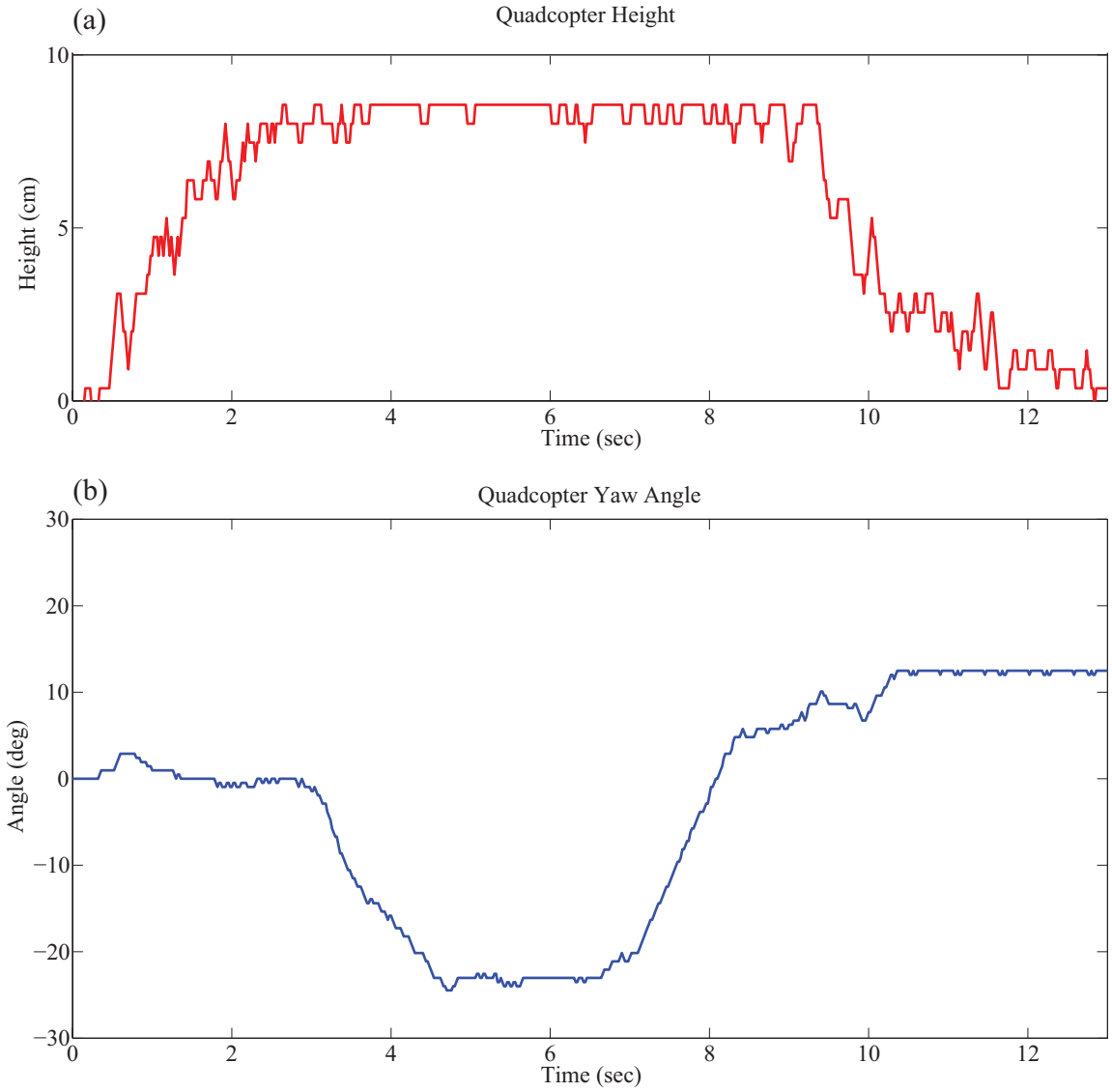


Figure 6.12: The vertical flight and yaw testing. (a) Shows the height of the quadcopter during the vertical takeoff and landing. (b) Is the quadcopter yaw angle as it is being cycled through the vertical takeoff and landing test.

The image that the quadcopter is tracking was moved to a height of 10 cm and the error between the quadcopter and the image is 2 cm. Next the image is rotated to 30 deg and the error between the image is 5 deg. Then the image is rotated back to 0 deg and lowered to a height of 0 cm. The quadcopter was able to follow the image initially, but if the image is moved around too much the quadcopter can lose the edge it was initially tracking. If the edge is lost and the optic flow sensor can't find another edge, it will either stay at the same height of keep rotating until it finds a edge. This is why the quadcopter's yaw angle increases as it is lowering back to the ground.

6.4 Summary

The quadcopter was tested to determine it's ability to reject a disturbance and then the quadcopter was tested with the optic flow sensors enabled. The sensors proved that they are capable of feeding back information to the quadcopter that will help to control and stabilize the motion.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis the quadcopter control algorithm was tested along with the use of optic flow sensing. The quadcopter platform designed and built for this thesis proved to be the perfect fit for the tests that were performed. Furthermore the size and weight of the quadcopter are perfect for the application of indoor flight. The quadcopter's small size, durability, and ease of construction make it an ideal candidate for any search and rescue or surveillance mission. There were various testes done to determine how the quadcopter performed by it's self and how well it can handle disturbances. The quadcopter was able to demonstrate stable flight because the RMS for the PID is 0.34, which is a 320 percent improvement from marginally stable case and a 4,500 percent improvement from the proportional controller while on the test stand. Also the quadcopter demonstrated that it can recover after a disturbance that takes it away from equilibrium for this disturbance it was found that the quadcopter performed better with a shorter rod connecting it to the test stand, which gives a maximum overshoot of 2.5 deg. The overshoot of the short rod is 1 deg higher than the long rod and the long rod is slower to respond, but the response of the quadcopter is slower. The slower response of the quadcopter with the long rod is due to the increased weight of the rod and the longer arc length that the quadcopter needs to travel to

reach equilibrium. The quadcopter's system proved it's self to be robust when tested with the different rod lengths while on the test stand.

When testing the optic flow sensors and their ability to track while on the the quadcopter they were fairly accurate and only having a maximum error of 7.9 deg with a PID controller. The use of the PID controller was a 12.5 percent decrease in overshoot from the PI controller and a 73 percent decrease from the P controller. The only problem with this test is that it didn't demonstrate how the quadcopter would perform under free flight. The test done where the quadcopter is allowed to move vertically is a more realistic scenario. Since the quadcopter is allowed to move freely in more directions it makes it harder for the quadcopter to follow the image that it was tracking.

From the tests done on the quadcopter there were limitations that are apparent while testing the optic flow sensors. The optic flow sensors can not track in a scene if there are not any edges present. For example if the quadcopter lost the image it was initially tracking it would see the white background and since the background doesn't have a gradient it would not be able to track the difference from one image to the next. This is only a problem for a very specific scenario because most rooms have objects in them and those objects create edges, which have the ability to be tracked. A similar problem arises when the lighting in the scene is low. When lighting is low in a scene the optic flow sensor is not able to determine the difference between the objects and the background in the scene, which is similar to edges not be present in images.

7.2 Future Work

The tracking ability of the optic flow sensor can be enhanced. To improve the optic flow sensing in areas with various lighting, a light sensor can be used that works with

the vision chip to adjust the gain within the chip, which will allow the optic flow sensor to work more consistently in different lighting conditions. Also if the quadcopter was equipped with a light, the area around the quadcopter can be illuminated to allow flight in low light areas.

The next improvement can be with the SRAM, which is the memory on the optic flow sensor where the images are being stored. Currently the maximum image size that can be collected is (16×16) pixels. If the memory is expanded the image size can be greatly increased, which means that the sensor can be more accurate in different environments because it will be able to track more edges in a given scene. Lastly the addition of a infrared camera and light to use for tracking would give the quadcopter the ability to see edges in dark scenes as well as having the ability to move throughout a well light scene. The infrared camera and light would also add a stealth aspect to the quadcopter that can be advantageous in certain scenarios.

Quadcopters have many applications and can have to potential to serve as a invaluable tool in search and rescue and can help to save lives. The use of optic flow on a quadcopter or any autonomous platform is valuable especially for indoor navigation where GPS isn't available.

Bibliography

[1] (2013, Nov) Tokyo electric power company. [Online]. Available: http://www.tepco.co.jp/en/

[2] (2013, Nov) Honeywell international inc. [Online]. Available: http://honeywell.com/

[3] G. Hoffmann, D. Rajnarayan, S. Waslander, D. Dostal, J. S. Jang, and C. Tomlin, "The stanford testbed of autonomous rotorcraft for multi agent control (starmac)," in *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd*, vol. 2, 2004, pp. 12.E.4–121–10 Vol.2.

[4] A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M. A. Hsieh, L. E. Parker, and K. Sty, Eds., *Cooperative Grasping and Transport Using Multiple Quadrotors*. Springer Berlin Heidelberg, 2013.

[5] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction with quadrotor teams," *Autonomous Robots*, vol. 33, no. 3, pp. 323–336, 2012.

[6] K. Daniel and C. Whietfeld, "Using public network infrastructures for uav remote sensing in civilian security operations," *Homeland Security Affairs*, vol. 7, 2011.

[7] M. Alvissalim, B. Zaman, Z. Hafizh, M. Ma'sum, G. Jati, W. Jatmiko, and P. Mursanto, "Swarm quadrotor robots for telecommunication network coverage area expansion in disaster area," in *SICE Annual Conference (SICE), 2012 Proceedings of*, 2012, pp. 2256–2261.

[8] I. Sa and P. Corke, "Verticle infrastructure inspection using a quadcopter and shared autonomy control," in *The 8th International Conference on Field and Service Robotics*, K. Yoshida and S. Tadokoro, Eds. Matsushima, Miyagi, Japan: Springer, 2012.

[9] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.

[10] E. Altug, J. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 1, 2002, pp. 72–77 vol.1.

[11] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus, "Energy-efficient autonomous four-rotor flying robot controlled at 1 khz," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 361–366.

[12] J. Kim, M.-S. Kang, and S. Park, "Accurate modeling and robust hovering control for a quadrotor vtol aircraft," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 9–26, 2010.

[13] G. R. Laszlo. Kis, Zoltan. Prohaszka, "Calibration and testing issues of the vision, inertial measurement and control system of an autonomous indoor quadrotor helicopter," in *RAAD*, 2008.

[14] C. Liu, W.-H. Chen, and J. Andrews, "Piecewise constant model predictive control for autonomous helicopters," *Robotics and Autonomous Systems*, vol. 59, no. 78, pp. 571 – 579, 2011.

[15] S. Lupashin, A. Schollig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 1642–1648.

[16] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed: experimental evaluation of multirobot aerial control algorithms," *IEEE Robotic Automatin Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[17] D. Honegger, P. Greisen, L. Meier, P. Tanskanen, and M. Pollefeys, "Real-time velocity estimation based on optical ow and disparity matching," in *Intelligent Robots and Systems*, 2012.

[18] L. Meier, P. Tanskanen, L. Heng, G. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *AUTONOMOUS ROBOTS*, vol. 33, no. 1-2, pp. 21–39, 2012.

[19] D. Watman and H. Murayama, "Design of a miniature, multi-directional optical ow sensor for micro aerial vehicles," in *International Conference on Robotics and Automation*, 2011.

[20] J. Zufferey, A. Beyeler, and D. Floreano, "Autonomous ight at low altitude using light sensors and little computational power," *International Journal of Micro Air Vehicles*, vol. 2, pp. 107–117, 2010.

[21] M. Bonak, D. Matko, and S. Blai, "Quadrocopter control using an on-board video system with off-board processing," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 657–667, 2012.

[22] C. Ivancsits and M. Lee, "Visual navigation system for small unmanned aerial vehicles," *Sensor Review*, vol. 33, no. 3, pp. 267–291, 2013.

[23] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart, "Mav navigation through indoor corridors using optical ow," in *International Conference on Robotics and Automation*, 2010.

[24] F. Kendoul, I. Fantoni, and K. Nonami, "Optic ow-based vision system for autonomous 3d localization and control of small aerial vehicles," *Robotics and Autonomous Systems*, vol. 57, no. 67, pp. 591–602, 2009.

[25] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, "An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications," in *International Conference on Robotics and Automation*, 2013.

[26] J. Bristeau, F. Callou, D. Vissiere, and N. Petit, "The navigation and control technology inside the ar.drone micro uav," in *18th IFAC World Congress*, 2011.

[27] (2013, Jun) Ar.droneparrot. [Online]. Available: http://ardrone.parrot.com

[28] L. Garca Carrillo, E. Rondon, A. Sanchez, A. Dzul, and R. Lozano, "Stabilization and trajectory tracking of a quad-rotor using vision," *Journal Intelligent Robotic Systems*, vol. 61, no. 1-4, pp. 103–118, 2010.

[29] S. Griffiths, J. Saunders, A. Curtis, B. Barber, T. McLain, and R. Beard, "Maximizing miniature aerial vehicles," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 34–43, 2006.

[30] J. Conroy, G. Gremillion, B. Ranganathan, and J. Humbert, "Implementation of wide-field integration of optic flow for autonomous quadrotor navigation," *AUTONOMOUS ROBOTS*, vol. 2009, no. 3, pp. 189–198, 2009.

[31] C. Pradalier, S. Bouabdallah, P. Gohl, M. Egli, G. Caprari, and R. Siegwart, "The coax micro-helicopter: A ying platform for education and research," *Advances in Autonomous Mini Robots*, pp. 89–99, 2012.

[32] A. Beyeler, J.-C. Zufferey, and D. Floreano, "Vision-based control of near obstacle flight," *Autonomous Robots*, vol. 27, pp. 201–219, 2009.

[33] F. Fraundorfer, L. Heng, D. Honegger, G. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," in *Intelligent Robots and Systems*, 2012.

[34] (2013, Sept) Ascending technologies. [Online]. Available: http://www.asctec.de/

[35] (2013, Oct) Dji innovations. [Online]. Available: http://www.dji.com/

[36] (2013, Nov) Parallax inc. [Online]. Available: http://www.parallax.com/

[37] (2013, Jun) Parrot. [Online]. Available: http://ardrone2.parrot.com/

[38] A. Bachrach, S. Prentice, R. He, and N. Roy, "Rangerobust autonomous navigation in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 5, pp. 644–666, 2011.

[39] M. Turpin, N. Michael, and V. Kumar, "Trajectory design and control for aggressive formation flight with quadrotors," *Autonomous Robots*, vol. 33, no. 1-2, pp. 143–156, 2012.

[40] K. Boudjit and C. Larbes, "Control and stabilization applied to micro quadrotor ar.drone," in *Proceedings of the 3rd International Conference on Application and Theory of Automation in Command and Control Systems*, ser. ATACCS '13. ACM, 2013, pp. 122–127.

[41] O. Amidi, T. Kanade, and R. Miller, "Vision-based autonomous helicopter research at carnegie mellon robotics institute," in *Heli Japan*, 1998.

[42] P. Corke, "An inertial and visual sensing system for a small autonomous helicopter," *Journal of Robotic Systems*, vol. 21, no. 2, pp. 43–51, 2004.

[43] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A system for autonomous flight using onboard computer vision," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 2992–2997.

[44] C. Kemp, "Visual control of a miniature quad-rotor helicopter," 2006.

[45] F. Kendoul, I. Fantoni, and K. Nonami, "Optic flow-based vision system for autonomous 3d localization and control of small aerial vehicles," *Robotics Autonomous Systems*, vol. 57, no. 6-7, pp. 591–602, 2006.

[46] (2013, Jun) Centeye, inc. [Online]. Available: http://centeye.com

[47] S. Barnard and W. Thompson, "Disparity analysis of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 4, pp. 333–340, 1980.

[48] R. Jain, R. Kasturi, and B. Schunck, Eds., *Machine Vision*. McGraw Hill, 1995.

[49] S. Ullman, "Interpretation of visual motion," *MIT Press, Cambridge*, 1979.

[50] B. Horn and B. Schunck, "Determining optical ow," *Articial Intelligence*, vol. 17, pp. 185–203, 1981.

[51] M. V. Srinivasan and R. L. Gregory, "How bees exploit optic flow: Behavioural experiments and neural models [and discussion]," *Philosophical Transactions: Biological Sciences*, vol. 337, no. 1281, pp. pp. 253–259.

[52] B. Lucas, "Generalized image matching by the method of differences," *PhD thesis*, vol. School of Computer Science, Carnegie Mellon University, 1984.

[53] C. Fennema and W. Thompson, "Velocity determination in scenes containing several moving objects," *Comp. Graph. and Image Proc.*, vol. 9, pp. 301–315, 1979.

[54] (2013, Feb) Aeroquad 32. [Online]. Available: http://http://aeroquad.com/

[55] (2013, Feb) Apm 2.5. [Online]. Available: http://3drobotics.com

[56] (2013, Feb) Autoquad v6.6. [Online]. Available: http://autoquad.org/

[57] (2013, Feb) Kkmulticopter. [Online]. Available: http://multicopter.org/wiki/Kkmulticopter/

[58] (2013, Feb) Openpilot cc3d. [Online]. Available: http://wiki.openpilot.org/

[59] (2013, Feb) Uavp/uavx. [Online]. Available: http://www.quadroufo.com/

[60] (2013, Feb) Cirus multiwii lite. [Online]. Available: http://www.multiwii.com/software

[61] (2013, Feb) Cirus multiwii se. [Online]. Available: http://www.multiwii.com/software

[62] (2013, Feb) Paris multiwiicopter. [Online]. Available: http://www.multiwii.com/software

[63] (2013, Feb) Quadrino zoomflight. [Online]. Available: http://www.multiwii.com/software

[64] G. Ritter and J. Wilson, Eds., *Handbook of Computer Vision Algorithms in Image Algebra*. CRC Press, 2001.

[65] A. M. Harrington, "Optimal propulsion system design for a micro quad rotor," *ProQuest Dissertations and Theses*, p. 139, 2011.

[66] M. Huang, Ed., *Vechicle Crash Mechanics*. CRC Press, 2002.

[67] K. Ogata, Ed., *Modern Control Engineering*. Pearson Higher Ed, 2010.

Appendix A

Code

Pseudo code for quadcopter setup and control

Quadcopter is turned on

Initial Setup

Initialize the gyro, accelerometer and motor outputs

Arm the motor ESCs by pulsing them high and then low

Run the leveling sequence for the gyro and accelerometer

Once the quadcopter

While the power is on

Read the gyro and accelerometer sensor

Calculate the angle of the quadcopter based on the accelerometer

Integrate the gyro data

Calculate the angle of the quadcopter based on the integrated gyro data

Filter the gyro data with a high pass filter

Combine the gyro and accelerometer quadcopter angles with a complementary filter

Read the odometry values from the front and bottom optic flow sensors

Actuate the motors (u1,u2,u3,u4) using the following:

u1 = (Throttle) - (Pitch Angle * kp + Pitch $\int$Angle * ki + Pitch Angle dt * kd) + (Roll Angle * kp + Roll $\int$Angle * ki + Roll Angle dt * kd) - (currentAngleZOptflow * kpYaw + Yaw $\int$Angle * kiYaw + Yaw Angle dt * kdYaw) + ((currentHeightOptflow) * kpHeight + ($\int$ Height) * kiHeight + (Height dt) * kdHeight);

u2 = (Throttle) + (Pitch Angle * kp + Pitch $\int$Angle * ki + Pitch Angle dt *kd) - (Roll Angle * kp + Roll $\int$Angle * ki + Roll Angle dt *kd) - (currentAngleZOptflow * kpYaw + Yaw $\int$Angle * kiYaw + Yaw Angle dt * kdYaw) + ((currentHeightOptflow) * kpHeight + ($\int$Height) * kiHeight + (Height dt) * kdHeight);

u3 = (Throttle) - (Pitch Angle * kp + Pitch $\int$Angle * ki + Pitch Angle dt *kd) - (Roll Angle * kp + Roll $\int$Angle * ki + Roll Angle dt * kd) + (currentAngleZOptflow * kpYaw + Yaw $\int$Angle * kiYaw + Yaw Angle dt * kdYaw) + ((currentHeightOptflow) * kpHeight

```
    + (∫Height) * kiHeight + (Height dt) * kdHeight );
  u4 = ( Throttle ) + ( Pitch Angle * kp + Pitch ∫Angle * ki + Pitch
    Angle dt *kd ) + ( Roll Angle * kp + Roll ∫Angle * ki + Roll
    Angle dt *kd ) + ( currentAngleZOptflow * kpYaw + Yaw ∫Angle *
    kiYaw + Yaw Angle dt * kdYaw ) + ( (currentHeightOptflow) * kpHeight
    + (∫Height) * kiHeight + (Height dt) * kdHeight );
  end
end
Pseudo code for optic flow sensors
Optic flow sensor is turned on
Initial Setup
  Set all ports used low
While sensor is on
  Copy previous image in a separate array
  Read pixel value for every pixel in the desired image size
  Store the new image in an array
  Shift current and past images up, down, left, and right and save them
    to different arrays
  Rotate current and past images clockwise and counter clockwise by a
    specific amount and store
  Use the shifted and rotated images in the optic flow algorithm
  Store odometry values
  Send odometry values over I²C when called
  end
end
```