NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

EXTENDING QUAD-ROTOR UAV AUTONOMY WITH ONBOARD IMAGE PROCESSING

by

Bradley R. Turnbaugh

March 2015

Thesis Advisor: Oleg A. Yakimenko
Co-Advisor: Feng Lin

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2015	Master's Thesis

4. TITLE AND SUBTITLE
EXTENDING QUAD-ROTOR UAV AUTONOMY WITH ONBOARD IMAGE PROCESSING

5. FUNDING NUMBERS

6. AUTHOR(S) Bradley R. Turnbaugh

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

One of the most dynamic technological advances of the last decade is the development of unmanned and autonomous vehicles. For the military, these vehicles represent a safer and more efficient way of fighting wars in aerial, ground, maritime, and underwater domains. Public and private companies have also vigorously researched these vehicles and used them for a wide range of tasks, from search-and-rescue operations to building inspections. Navigating these vehicles typically involves the use of GPS or other external cues to follow a path, detecting for and correcting errors along the way.

The purpose of this research is to investigate the feasibility of tracking a ground target using a quadrotor that navigates solely based on relative position to the target. To achieve this goal, the quadrotor, a Quanser Qball-X4, is fitted with a small camera. By processing the camera's image and utilizing pitch, roll, and altitude data from other onboard sensors, a targeting solution can be derived. To track the target, the tracking vehicle defines error as any deviation from the desired angular offset from that target, continuously correcting that error to maintain its desired offset. By using relative position, the tracking vehicle can continue to follow the target using its onboard camera.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Tracking, relative, unmanned vehicles, autonomous vehicles, error, control, offset.	123
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540–01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

EXTENDING QUAD-ROTOR UAV AUTONOMY WITH ONBOARD IMAGE PROCESSING

Bradley R. Turnbaugh
Lieutenant, United States Navy
B.S., University of Illinois at Urbana-Champaign, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2015

Author: Bradley R. Turnbaugh

Approved by: Oleg A. Yakimenko
Thesis Advisor

Feng Lin
Co-Advisor

Garth V. Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

One of the most dynamic technological advances of the last decade is the development of unmanned and autonomous vehicles. For the military, these vehicles represent a safer and more efficient way of fighting wars in aerial, ground, maritime, and underwater domains. Public and private companies have also vigorously researched these vehicles and used them for a wide range of tasks, from search-and-rescue operations to building inspections. Navigating these vehicles typically involves the use of GPS or other external cues to follow a path, detecting for and correcting errors along the way.

The purpose of this research is to investigate the feasibility of tracking a ground target using a quadrotor that navigates solely based on relative position to the target. To achieve this goal, the quadrotor, a Quanser Qball-X4, is fitted with a small camera. By processing the camera's image and utilizing pitch, roll, and altitude data from other onboard sensors, a targeting solution can be derived. To track the target, the tracking vehicle defines error as any deviation from the desired angular offset from that target, continuously correcting that error to maintain its desired offset. By using relative position, the tracking vehicle can continue to follow the target using its onboard camera.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

GPS	Global Positioning System
LED	Light-Emitting Diode
LOS	Line-of-Sight
LQR	Linear-Quadratic Regulator
LTP	Local Tangent Plane
OCR	Optical Character Recognition
PD	Proportional-Derivative
PWM	Pulse-Width Modulation
RGB	Red-Green-Blue
TCAS	Traffic Collision Avoidance System
TCP/IP	Transmission Control Protocol/Internet Protocol
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
USB	Universal Serial Bus
USV	Unmanned Surface Vehicle
UUV	Unmanned Underwater Vehicle

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank Dr. Oleg Yakimenko for his advice and support throughout this research; Dr. Feng Lin for his assistance from the other side of the world; Juan Gonzalez for technical support and problem solving in the laboratory; Matt O'Brian for the backup and video during the experiments; Amirpasha Javid and the rest of the folks at Quanser for providing great customer support on the Qball-X4 and Quarc software; Levi Jones, Michael Martin, and all of Dr. Yakimenko's former thesis students for leaving the quadrotors and other laboratory equipment in better shape than they found it; Dr. Timothy Chung and Dr. Douglas Horner for helping to plant the seeds that led to this research; the Mechanical and Aerospace Engineering Department and Undersea Warfare Academic Group for all their support leading up to and during this project; and my wonderful wife Katy for always being there for me.

THIS PAGE INTENTIONALLY LEFT BLANK

I. BACKGROUND AND PROBLEM FORMULATION

Though unmanned vehicles have been utilized for over a century, many major advances have occurred in the past two decades that have caused military and civilian agencies to incorporate the use of unmanned vehicles in various missions and tasks. Today, these vehicles perform a variety of tasks, from strike to surveillance to communications. Some vehicles, like the MQ-1 Predator, are operated remotely by a human operator; others, like the BGM-109 Tomahawk cruise missile, can maneuver autonomously by following pre-programmed control law.

Unmanned vehicles are often grouped by the domain in which they operate. Examples include unmanned aerial vehicles (UAVs), unmanned ground vehicles (UGVs), unmanned (water) surface vehicles (USVs), and unmanned underwater vehicles (UUVs).[1] Though these vehicles have many differences, the principles that dictate their control are very similar. The successful control of any autonomous vehicle is contingent on its ability to detect and correct for error, allowing it to make the necessary adjustments to complete its mission even when the unexpected occurs.

One major function of unmanned vehicles is tracking. The ability to track another moving object without human input is extremely useful in many applications, from vehicle pursuit to formation flight to maritime tracking. This thesis focused on the optical tracking of a ground target from an unmanned rotorcraft using a single fixed camera mounted on the front of the aircraft. Control of the aircraft was based exclusively on the relative position of the target; any deviation of the target from the center of the camera's view was treated as an error, and the aircraft rolled and pitched as needed to minimize that error. Through this method of control, an autonomous vehicle should be able to track any target, maintaining strong contact of that target throughout the pursuit.

A. RELATED WORK

Unmanned vehicles are used by researchers throughout the world to study control theory, aerodynamics, guidance, and dozens of other subjects. Many are equipped with

onboard cameras for experiments involving tracking, image recognition and search theory among other topics. A small sample of this work is summarized in this section.

(1) National University of Singapore

The National University of Singapore maintains a dynamic unmanned vehicles program, where students conduct research involving UAVs, some of which have onboard cameras. One recent project proposed to build a real-time tracking algorithm for following a ground target.[2] The research team attached a small color video camera on their SheLion quadrotor, installing a small servo mechanism to allow the camera to pan and tilt.[2] For image recognition, they used pattern recognition and thresholding to separate the target from the background, using the pixel indices of the target and the angle of the camera to resolve the target's location.[2] They designed a tracking algorithm to follow the target by moving the UAV and swiveling the camera. One of the goals of the research is to eventually develop a method of automatically landing a UAV on a moving platform without the use of GPS.[2] The image in Figure 1 shows a picture of the SheLion from another experiment.

Figure 1. The SheLion quadrotor, part of the family of quadrotors used for research by the National University of Singapore (from "Lion UAV Systems", National University of Singapore, URL: http:// uav.ece.nus.edu.sg/uavfamilies.html [cited 1 March 2015]).

(2) Arizona State University

The Autonomous Systems Technologies Research and Integration Laboratory (ASTRIL) at Arizona State University is also working on a project involving GPS-denied tracking and following for a UAV.[3] Their research uses the Parrot AR Drone 2.0 quadrotor, which shown in the left photo of Figure 2. The algorithm is tested in an outdoor suburban environment, where the Parrot successfully tracks a variety of objects including people, cars, and windows.[3] ASTRIL's research is open source; their code is posted on their web page along with videos showing their algorithm in action.[3]

Figure 2. The Parrot AR Drone 2.0, which is used in ASTRIL's research to identify, track, and follow various targets. The left photograph shows the UAV, while the right photograph shows a view from its onboard camera with a box around its target.[3]

(3) University of Central Florida

Researchers at the Computer Vision Lab at the University of Central Florida created an algorithm for tracking a ground vehicle from a fixed wing aircraft. The ground vehicle is optically tracked using a camera hanging underneath the aircraft.[4] The camera, shown in Figure 3, contains a gimbal lock and is capable of 360° horizontal and 90° vertical motion. The Computer Vision Lab simulated this tracking algorithm by having the aircraft perform circle maneuvers while maintaining a lock on the target.[4] Examples of this trajectory can be seen in the right half of Figure 3; the aircraft trajectory is red and the ground vehicle trajectory is blue.

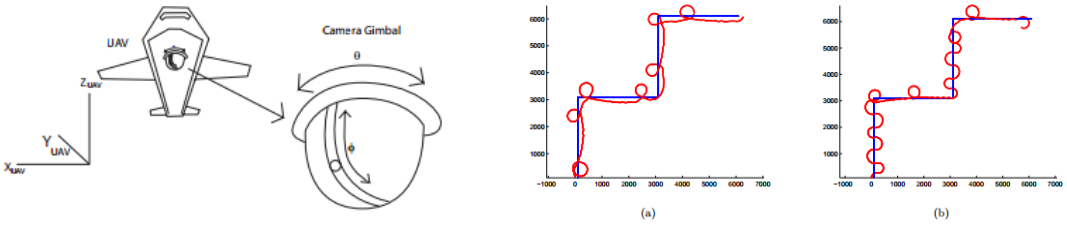

Figure 3. A tracking algorithm developed at the University of Central Florida. The left diagram shows the gimbal-lock camera, and the right plots show the results of the laboratory's simulation.[4]

(4) Naval Postgraduate School

The Naval Postgraduate School in Monterey, California has been researching unmanned vehicles for more than a decade. One of the more extensive undertakings involves developing an algorithm to allow a parafoil system, called the Snowflake, to land on the deck of a ship while underway.[5] This algorithm used a fixed sensor to estimate the motion of its target to develop a landing solution upon that target. Additionally, the algorithm computes the winds affecting the Snowflake's flight path, allowing the system to build a more optimal landing trajectory.[5]

There have also been numerous research projects involving trajectory generation at the Naval Postgraduate School using the Quanser Qball-X4 and the Parrot AR Drone 2.0. In one project, a Qball-X4 flew a pre-programmed pattern above the ground, imaging the surface and detecting obstacles.[6] The location of these obstacles was sent to an unmanned ground vehicle, which then developed an optimal path to travel through the same space while avoiding the obstacles.[6] Another research project involved the implementation of multiple Qball-X4, which were programmed to cooperatively fly along a pre-programmed trajectory in a very small space.[7]

B. APPLICATIONS OF RELATIVE POSITION TRACKING

The use of relative position in developing a tracking solution extends across multiple domains. In every case, a sensor must provide an accurate bearing to the target. Also, range to target must be measured, either using the same sensor or other measurements.

1. Formation Flight

Fighter and stunt pilots regularly use relative position tracking when flying in formation. In a typical two-plane formation, the lead aircraft sets a consistent airspeed and altitude while limiting any abrupt movements. The other aircraft, or wingman, maintains a constant offset from the lead, typically behind and to the left of the lead aircraft. The wingman must continuously work to maintain that position, making small corrections in power and attitude to solidify that position. Unlike the lead, the wingman does not rely on instruments; instead, he or she relies on a sight picture to maintain a tight formation. The image in Figure 4 shows an example of the proper sight picture, using the Mooney M20 as an example.

Figure 4. Example of proper sight picture in formation flight with the left wing crossing the front tip of the aircraft (from Mooney Caravan, *Mooney Caravan Formation Guide*, Rev. 3, Feb. 2013, p. 23).

A UAV offers many advantages over a manned aircraft for formation flying. With good processing power, the UAV's reaction time is much quicker, allowing the aircraft to make necessary corrections swiftly and continuously. The following aircraft can be programmed to quickly recognize trends, such as a decrease in speed, and adjust its power and attitude accordingly. In addition, close formation flying can be strenuous for a wingman over time; an unmanned aircraft will not have this issue. As a result, UAVs

offer great potential in the realm of formation flying, as formations of 50 or 100 aircraft can operate in close proximity safely and effectively.

To fly in formation using relative position tracking, the wingman UAV must have a good sensor that provides three-dimensional bearing as well as range. Greater accuracy would allow the wingman to fly a tighter formation. Also, safeguards must be in place, such as an emergency "pull from formation" maneuver in case of engine failure, since emergencies have the potential to damage multiple aircraft.

2. Open-Ocean Search and Rescue

A search and rescue operation presents many unique challenges. One of the greatest challenges is the search itself, which often involves searching a very large area of water for the vessel that is in distress. The capability of UAVs to fly in formation, as described previously, could allow unmanned surveillance aircrafts to search the area with a massive sweep width, especially if 100 or more UAVs are used in the search. Additionally, once the vehicles find the vessel under distress, relative motion tracking could be used to allow an unmanned vehicle to maintain position near the vessel while feeding accurate and updated position information back to Navy or Coast Guard units. Use of UAVs in these scenarios could save lives in future operations, adding to the capabilities of the U.S. Coast Guard while making the open ocean safer for everybody.

3. Vehicle Pursuit

Relative position tracking could be a very useful approach when tracking a ground vehicle using aerial assets. The tracking vehicle must be capable of matching the speed and maneuvering capabilities of the target, and must account for any obstacles, especially in an urban or mountainous environment. Additionally, the tracking vehicle must be able to correctly identify the target and differentiate it from other objects in the area. This can be done using image recognition, threat assessment, or a unique infrared signature.

Vehicle pursuit would be very useful for military and civil agencies. A UAV tracking a vehicle would likely report the target's location data, giving ground assets an

opportunity to capture or kill the target. Covert tracking may be possible as well, allowing an agency to gather valuable intelligence concerning enemy bases and other associations. These methods could be applied to multiple, cooperative tracking UAVs, adding a redundancy to the pursuit that decreases the chances of losing contact.

4. Submarine Tracking

While the previous examples have focused on using UAVs as tracking vehicles, it is important to note that many of the same control laws apply to UUVs. Therefore, the same relative motion tracking techniques could allow a UUV to successfully track a submarine. Submarine tracking has been a challenging problem for decades, especially since submarines have gotten much quieter and harder to detect. Sonar's bearing accuracy is much lower than that of radar or optical sensors. Improvements in sonar technology combined with other sensor advancements, such as magnetic anomaly detection, could finally allow the tracking platforms to gain an advantage. Also, sending a group of UUVs to search for a submarine, using similar techniques as formation-flying UAVs, could vastly improve the chances of detecting a submarine, even in a large body of water.

C. BUILDING A TRACKING SOLUTION

Isaac Newton discussed the concepts of absolute and relative motion in the late seventeenth century, stating that a body standing on a ship has zero relative motion, but possesses the same absolute motion as the ship.[8] In modern control theory, this concept is vital to the discussion of the frame of reference. To an observer on the ground, two aircraft flying in the same direction and at the same velocity appear to travel in the same direction and at the same velocity. To an observer on one of the aircraft, however, the other aircraft appears to be at rest. The ground observer is using the local tangent plane (LTP) frame of reference, while the aircraft observer is using a body-fixed frame of reference.

When building a vehicle tracking solution, relative position is paramount. The tracking vehicle should strive to be in a set position relative to the target, with zero relative velocity. To achieve this ideal position, the vehicle must sense the target's

relative position and velocity, using this data to continuously refine its position. Any maneuvering or acceleration of the target can be sensed in the same way, allowing the vehicle to successfully track the target using these relative dynamics.

There are many important considerations that must be taken when developing a tracking algorithm, including object recognition, obstacle avoidance, and the tracking vehicle's operating limits. This section discusses these considerations as well as how these risks are mitigated in this research.

1. Object Recognition

Target recognition consists of loading an image or group of images into an algorithm that identifies various objects within that image. This task, which is inherently easy to the human eye, often involves complex algorithms and can require a high amount of processing power, especially for high-resolution images.[9] There are many methods that are used to recognize these objects. One common method is to detect features, such as lines, blobs, or intersections, and compare the features against object templates that exist in the computer's memory.[9] This method can allow a computer to identify objects, often in real-time, with a very high degree of accuracy.

One example of object recognition is optical character recognition (OCR), which allows a scanned image to be converted into text, producing a document that can be edited and republished by a computer user. OCR uses algorithms to analyze the edges and gaps in each character, making an estimate as to which character it is.[10] Handwriting can also be converted into computer text using these object recognition techniques, though the accuracy of these algorithms is often not as good. Additionally, the content of photographs can be analyzed. Researchers at Google and Stanford University have created software that can describe a scene, such as people playing Frisbee or elephants marching on a field, with fairly high accuracy.[11]

For this problem, the processing speed was severely limited. Because of this, the target was "tagged" with an easily-recognizable mark, allowing the tracking vehicle to distinguish the target from other objects in the laboratory environment. It is important to

note that for a real world application object recognition would definitely play an important role, making processing power an important consideration.

2. Obstacle Avoidance

The ability for an unmanned vehicle to avoid obstacles in a dynamic environment is as challenging as it is essential. Obstacle avoidance can be divided into three types: optimal, thorough search, and reactive.[12] An optimal system often involves building a route to minimize a factor or set of factors that could include time, fuel consumption, risk, and operational cost.[12] A thorough search system typically involves a comparison of different possible paths, using some of these factors to choose the best path for the unmanned vehicle to take. Both optimal and thorough search usually take prior planning and knowledge of the environment, and often require a large amount of processing power for complex environments.[12]

The third type of obstacle avoidance is a reactive system, which usually requires sensors that are either attached to the vehicle or are monitoring the environment from another platform. When an obstacle is detected, the vehicle must maneuver to avoid the obstacle then adjust its route to continue to its destination. One example of a reactive system is the Traffic Collision Avoidance System or TCAS, which is equipped on manned aircraft and is designed to give immediate maneuvering instructions to pilots when the possibility of an air-to-air collision exists.[13] Another reactive system, specifically for autonomous vehicles, uses artificial potential fields to model the environment as a set of attraction and repulsion points, changing the vehicle's trajectory to maintain a safe distance from obstacles.[12]

Most obstacle avoidance systems are developed for navigation problems. In the case of a tracking problem, the movement of the tracking vehicle is entirely dependent on what the target does, meaning that all obstacle avoidance must be reactive. Image recognition, while being used to identify the target, can also be used to locate obstacles. A radar system is also useful, as it can derive the bearing and range to an obstacle. Another method of obstacle avoidance is to incorporate the use of a second vehicle to

provide a real-time overlay of the environment and help direct the tracking vehicle around obstacles.

Obstacle avoidance, while extremely important, typically requires increased payload, greater resources, and a large amount of processing power. For this experiment, the assumption was that obstacles are not a factor. The test runs were completed in a controlled laboratory environment that was guaranteed to be free of obstructions. In a real-world tracking problem, however, obstacle avoidance is extremely important and will always require thorough planning and monitoring.

3. Operating Limits

Every manned aircraft has a unique set of operating limitations, which are calculated through extensive research and tested using simulations and wind tunnel assessments. These limits are further confirmed in flight by experienced test pilots. As a result, every aircraft has limitations, both maximum and minimum, for speed, angle of attack, payload, and a variety of other performance characteristics.[14] Failure to abide by these limitations could put the aircraft and its occupants at risk.

Unmanned aircraft have similar limitations that the operator must take into account during operation. Operating beyond these limitations is definitely risky, as this behavior could result in loss of aircraft, risk to people and property on the ground, and mission failure. Unlike manned aircraft, however, most operating limits are not known for UAVs, especially very simple and inexpensive ones. The best way to prevent an unmanned aircraft from exceeding its limits is to build performance restraints into the control algorithm that prevent the aircraft from getting close to these limits. This is especially important when using relative position to track a vehicle, since the target could potentially force the aircraft to exceed its limitations.

For this experiment, pitch and roll were limited to 8°, which kept the vehicle from losing control and limits its horizontal speed. Additionally, the aircraft was carefully tested when new components were attached to ensure that its balance was not adversely affected. The implementation of these controls allowed the UAV to remain in control throughout each experiment.

D. PROBLEM FORMULATION

The objective of this research was to program an Unmanned Aerial Vehicle to track a ground target using relative position, as measured from onboard sensors. The Quanser Qball-X4 quadrotor was selected as the tracking vehicle and an independently-controlled toy car was selected as the target. The primary tracking sensor was a small camera attached to the front of the Qball-X4. This camera was stationary, meaning it always pointed in the same direction and fed video in real-time to the computer that controlled the vehicle.

Figure 5 outlines the algorithm that was built to successfully track the target. First, an image was collected with the onboard camera and the target was identified on that image. The target's position on that image was then combined with other sensor data to calculate the target's location relative to the Qball-X4. Once the location of the target was known, it could be compared to a pre-programmed "desired position" to calculate a two-dimensional offset, which was treated as error. Finally, the controller adjusted the UAV to minimize the error in conjunction with the limitations imposed on the aircraft. This algorithm was repeated continuously throughout each test.

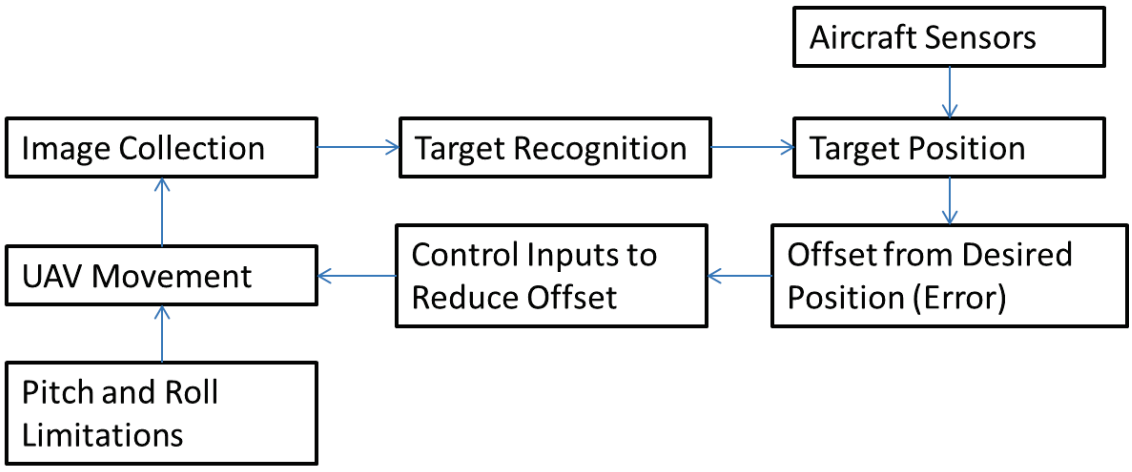

Figure 5. Tracking algorithm used in this research. This method allowed the Qball-X4 to continuously build a tracking solution and maintain a constant position relative to the target.

Due to payload and processing power limits, controls were implemented to better equip the UAV for the tracking problem. First of all, the image processing was simplified by "tagging" the vehicle with a contrasting shape that allowed the target to be easily identifiable. Second, the experiments were conducted in a well-lit laboratory that was free of obstacles and uneven terrain. Third, the orientation of the tracking was constant, meaning that the tracking vehicle did not change heading in the tests. Finally, the tracking vehicle was not required to search for the target, always beginning each test with the target in the camera's field of view.

II. QUANSER QBALL-X4 AND ITS CONTROLLERS

Quanser is a Canadian company that builds a variety of real-time control systems for use in education and research. Their products are open-architecture, which allows researchers to test a variety of new hardware and software concepts.[15] One of their products, the Quanser Qball-X4, is a research UAV consisting of four motors that drive 10-inch propellers. The quadrotor is encased by a protective carbon fiber cage, which keeps the Qball-X4 safe from collisions between it and obstacles or other vehicles.[16] The Qball-X4, which is shown in Figure 6, is 65 cm high, 72 cm wide, 74 cm long (including onboard camera), and weighs 1.5 kg.

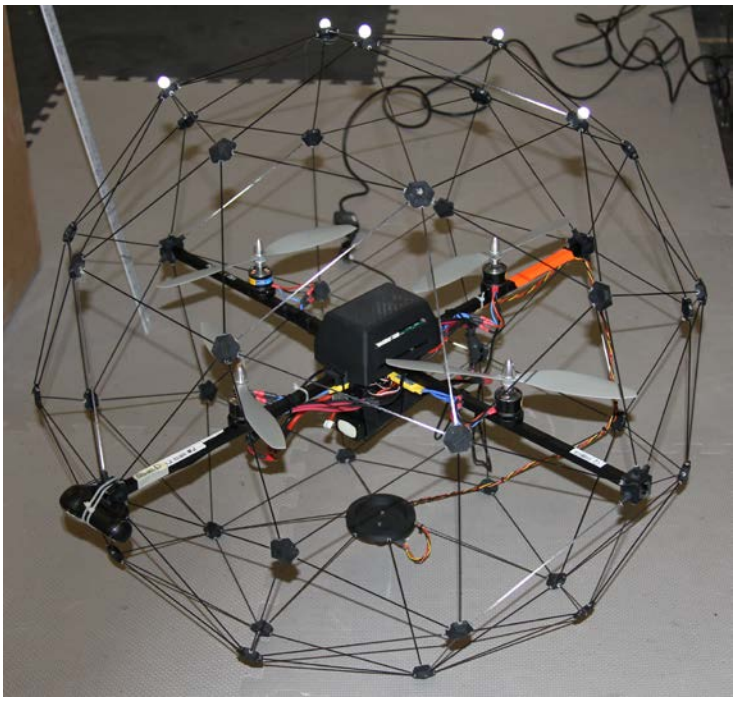

Figure 6. Quanser Qball-X4 quadrotor.

A. COMPONENTS

(1) Motors and Propellers

The Qball-X4 has four E-Flite Park 400 (740 Kv) motors which are mounted to the crossbeam frame of the UAV.[16] The motors are equally spaced from the center of the crossbeam and drive four counter-rotating APC 10x4.7 propellers. The front and back motors spin clockwise, while the left and right motors spin counter-clockwise. The motors are connected to the HiQ through speed controllers, which regulate the rotational speed of the motors.[16] The image in Figure 7 shows a motor-propeller assembly.

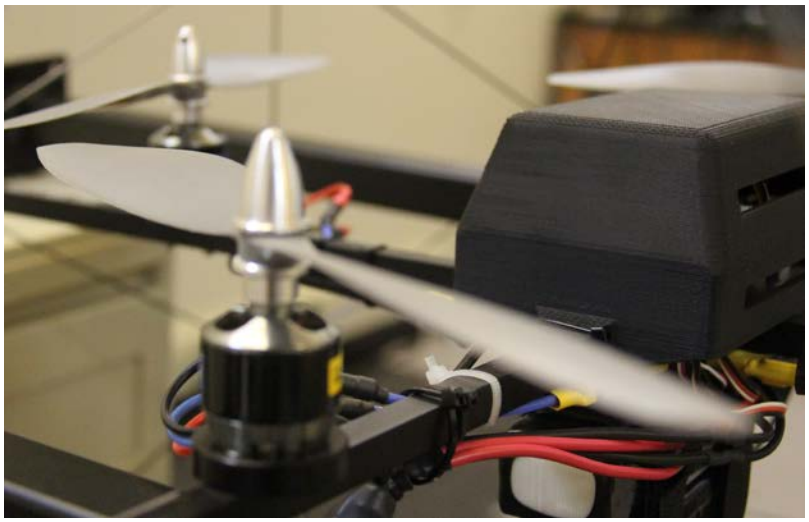

Figure 7. Motor-propeller assembly for the front motor of the Qball-X4.

(2) HiQ and Gumstix

The HiQ Data Acquisition Board controls the UAV by reading the various sensors and sending speed controls to the four motors. The HiQ is connected to the ground station through a wireless modem, allowing it to send and receive instructions. The Gumstix embedded computer, which is attached to the card, processes this information according to the Linux-based control instructions that are uploaded before each run.[16] The HiQ, pictured in Figure 8, contains a variety of sensors, as well as input/output slots for additional sensors[16]:

- 10 PWM outputs (servo motor outputs)

- 3-axis gyroscope, range configurable for ±75˚/s, ±150˚/s, or ±300˚/s, resolution
- 0.0125˚/s/LSB at a range setting of ±75˚/s
- 3-axis accelerometer, resolution 3.33 mg/LSB
- 6 analog inputs, 12-bit, +3.3V
- 3-axis magnetometer, 0.5 mGa/LSB
- 8 channel RF receiver input (optional)
- 4 Maxbotix sonar inputs
- 2 pressure sensors, absolute and relative pressure
- Input power 10–20V

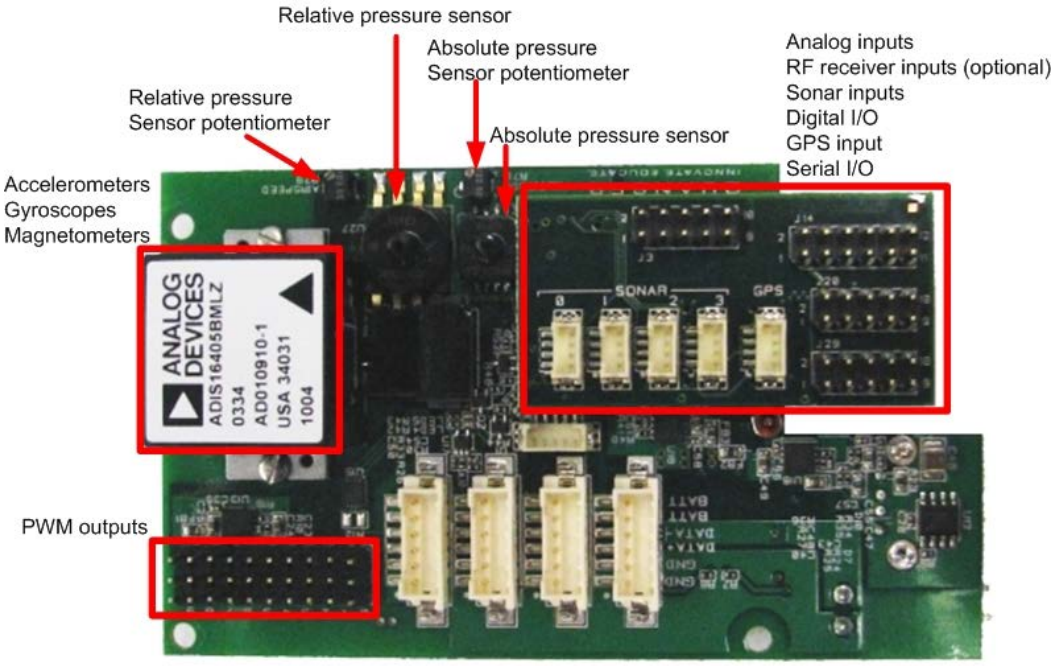

Figure 8. HiQ/Gumstix with embedded sensors and input/output slots.[16]

(3) Batteries

The Qball-X4 is powered by two 3-cell 2500mAh LiPo batteries. The batteries power the motors and HiQ card.[16] The batteries used in this experiment, shown in the image in Figure 9, are fully interchangeable and rechargeable. They are stacked vertically under the HiQ card, carefully placed to maintain the UAV's balance, and secured to the frame with two Velcro straps. Fully-recharged batteries can provide the Qball-X4 with up to 10 minutes of continuous flight time.[17]

Figure 9. 3-cell 2500mAh LiPo battery, manufactured by Rotor RC.

(4) Joystick

The Qball joystick allows the operator to fly the Qball-X4 manually. The joystick contains two sticks and four trim knobs, allowing control of the throttle, yaw, roll, and pitch of the UAV. Though the control model allows the operator to set autonomous modes for one or more of these controls, the throttle control acts as an activation and kill switch, providing an important safety mechanism that is required for all Qball-X4 operations.[16] The joystick, which is shown in the image in Figure 10, provides inputs to the master control model through a TCP/IP interface.

Figure 10. Quanser Qball joystick used to control the Qball-X4 in manual mode.

(5) Software

One major advantage of the Gumstix processor is that it can assume the control load during a test run. This allows the UAV to gather data from the sensors and send power outputs to the motors internally, allowing for more stable flight. To set this up, the control model must be uploaded to the HiQ before the initial test and any time the model is changed. This requires a Microsoft Windows 7 host machine, where the control model can be edited and adjusted as necessary before uploading it to the UAV.

The base control model is provided from Quanser and uses for Matlab 2011a and Simulink, which are produced by The Mathworks. Additionally, Quanser provides the Quanser Real-Time Control (QuaRC) toolbox, which adds specialized features and an additional Simulink block set, allowing the user to rapidly test different controllers and efficiently upload them to the HiQ card.[16] Once the model is loaded onto the UAV, the user can still control the UAV with the joystick as well as tune parameters as necessary, thanks to the wireless connection between the HiQ and the host computer.

B. MODELING

Many complex UAVs are controlled by varying engine speed, changing propeller blade pitch, and moving flight control surfaces. The Quanser Qball-X4 contains fixed-pitch propellers and no moveable control surfaces, meaning that all control is accomplished by varying the speed of the motors. To allow for this, the motors are controlled independently using the speed controllers on the HiQ. Additionally, the propellers are counter-rotating; the front and back propellers rotate clockwise, while the left and right propellers rotate counter-clockwise. This allows the operator full control of the UAV's yaw axis. The illustration in Figure 11 shows the basic top-down view of the thrust axes.

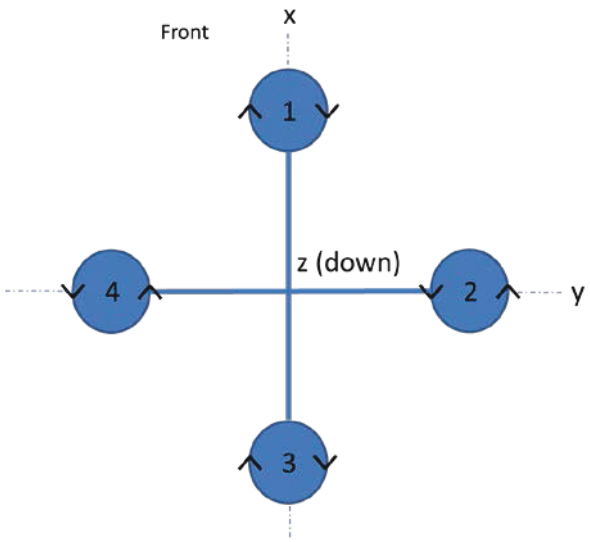

Figure 11. Model of the Qball-X4, with defined x, y, and z axes.

(1) Thrust

The following first-order system is used to model the thrust of each propeller.

$$F_i \quad K\frac{\omega}{s+\omega}u_i \tag{1}$$

where i is the propeller number, u is the PWM input to the motor, ω is the actuator bandwidth, and K is the preset gain. To represent the actuator dynamics without the gain, the state variable v is used.[16]

$$v_i \quad \frac{\omega}{s+\omega}u_i \tag{2}$$

(2) Roll and Pitch

Due to the symmetry of the Qball-X4, the behaviors of the roll and pitch axes are identical. For either axis, rotation is produced by creating a difference in thrust between two opposite propellers: a thrust difference between propellers 1 and 3 creates pitch rotation around the y-axis, while a thrust difference between propellers 2 and 4 creates roll rotation around the x-axis. To model this rotation, it is assumed that the roll and pitch rotations occur independent of each other. For the following example, pitch will be used, but roll can be calculated using the same equations. The diagram in Figure 12 shows a

view of the UAV looking down the y-axis, where L is the distance between either motor and the center of the frame.

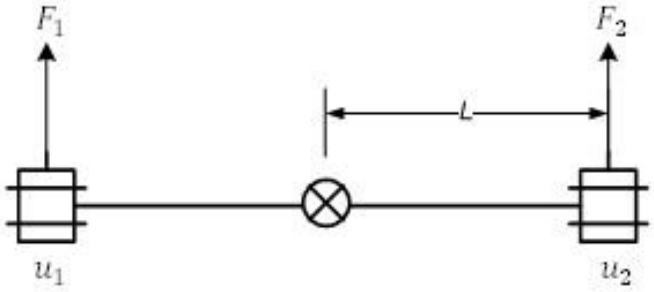

Figure 12. Model of the forces involved in the roll or pitch of the Qball-X4.[16]

Using the difference in thrust F, calculated in Eq. (1), the pitch angle θ can be derived.

$$J\ddot{\theta} = \Delta FL \tag{3}$$

where J is the rotational inertia along the pitch axis, L is the distance between the center of gravity and the center of thrust, and ΔF is the difference in thrust between motors 1 and 3. Table 1 contains parameters of the Qball-X4, including the values of J and L. Furthermore, combining the pitch axis dynamics with the actuator dynamics for the propellers allows the following state space equation set to be derived.[16]

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \dfrac{KL}{J} \\ 0 & 0 & -\omega \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \Delta u \tag{4}$$

Parameter	Value
K	120 N
ω	15 rad/sec
J_{roll}	0.03 kg·m^2
J_{pitch}	0.03 kg·m^2
M	1.4 kg
K_y	4 N·m
J_{yaw}	0.04 kg·m^2
L	0.2 m

Table 1. Parameters used to model the Qball-X4.[16]

(3) Yaw

To create rotation along the z-axis, or yaw, the UAV must increase the thrust on two opposing motors while decreasing the thrust on the other two motors. This creates aerodynamic torque, which causes the Qball to turn in a given direction without a pitch or roll motion. To model this motion, the following equation is used.

$$J_y \ddot{\theta}_y = \Delta\tau \tag{5}$$

where J_y is the rotational inertia around the z-axis, θ_y is the yaw angle, and $\Delta\tau$ is the difference between the sum of the thrust of motors 1 and 3 and the sum of the thrust of motors 2 and 4.[16] This equation is then rewritten in state space form as follows.

$$\begin{bmatrix} \dot{\theta}_y \\ \ddot{\theta}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_y \\ \dot{\theta}_y \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{K_y}{J_y} \end{bmatrix} \Delta\tau \tag{6}$$

(4) Position

The position of the Qball-X4 along the x- and y-axes can be modeled using total thrust and roll/pitch angles.

$$\begin{aligned} M\ddot{X} &= 4F\sin p \\ M\ddot{Y} &= -4F\sin r \end{aligned} \tag{7}$$

where M is the mass of the UAV as defined in Table 1, F is the thrust generated by each propeller, and p and r are the pitch and roll angles, respectively.[16] Using the small angle assumption for the pitch and roll angles, the following state space equations can be derived.

$$\begin{aligned} \begin{bmatrix} \dot{X} \\ \ddot{X} \\ \dot{v} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \dfrac{4K}{M}p \\ 0 & 0 & -\omega \end{bmatrix} \begin{bmatrix} X \\ \dot{X} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} u \\ \begin{bmatrix} \dot{Y} \\ \ddot{Y} \\ \dot{v} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \dfrac{4K}{M}r \\ 0 & 0 & -\omega \end{bmatrix} \begin{bmatrix} Y \\ \dot{Y} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} u \end{aligned} \tag{8}$$

(5) Altitude

The altitude of the UAV, Z, is affected by the sum of the thrust of all four propellers, and can be written as follows.[16]

$$M\ddot{Z} = 4F\cos r\cos p - Mg \tag{9}$$

Using the small angle approximation, the cosine values can be dropped from the equation. The resulting state space equation is as follows.

$$\begin{bmatrix} \dot{Z} \\ \ddot{Z} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \dfrac{4K}{M} \\ 0 & 0 & -\omega \end{bmatrix} \begin{bmatrix} Z \\ \dot{Z} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} u + \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} \tag{10}$$

C. CONTROLLER

The Qball-X4 controllers, like the models, are designed by Quanser. For this experiment, the controllers are largely unchanged; the source of error will change from OptiTrack position error to relative position error, but the methods of control will remain the same. This section will discuss the controllers for roll, pitch, yaw, position, and altitude.

(1) Roll and Pitch Controllers

The roll and pitch controllers are linear quadratic regulators (LQRs). To illustrate these symmetrical controllers further, this section will focus on the pitch controller, though it can be noted that both controllers use the same gains. First, Eq. 4 is modified by adding a fourth state $\dot{s} = \theta$ to allow for integrator feedback, creating the following equation.[16]

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{v} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{KL}{J} & 0 \\ 0 & 0 & -\omega & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ v \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \\ 0 \end{bmatrix} \Delta u \tag{11}$$

The above equation is in state-space form where A is a 4x4 matrix and B is a 4x1 matrix. To derive the gains, a cost function (J) must be built using pre-defined weighting matrices Q and R as follows.

$$J = \int_0^\infty (x^T Q x + u^T R u) dt$$

$$Q = \begin{bmatrix} 70 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 22000 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}; R = 30000 \tag{12}$$

The four matrices (A, B, Q, R) are then inputted into Matlab's LQR function to calculate the gain matrix, which produces poles at -19.96, -0.38, and -5.19±5.59*i*.

(2) Yaw Controller

The yaw controller is an LQR controller that contains two inputs. The state-space equation used for the yaw controller is identical to Eq. 6, and the Q and R matrices are as follows.

$$J = \int_0^\infty (x^T Q x + u^T R u) dt$$
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}; R = 1000 \tag{13}$$

When these four matrices are inputted into Matlab's LQR function to calculate the gain matrix, the poles are -3.76±1.29*i*.

(3) Position Controller

The position controller is similar to the pitch and roll controllers, as the state matrix is amended to include a fourth state for integrator feedback.

$$\begin{bmatrix} \dot{X} \\ \ddot{X} \\ \dot{v} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{4K}{M} p & 0 \\ 0 & 0 & -\omega & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ \dot{X} \\ v \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \\ 0 \end{bmatrix} u$$
$$\begin{bmatrix} \dot{Y} \\ \ddot{Y} \\ \dot{v} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{4K}{M} r & 0 \\ 0 & 0 & -\omega & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Y \\ \dot{Y} \\ v \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \\ 0 \end{bmatrix} u \tag{14}$$

For the position controller, the following cost function is used.

$$J = \int_0^\infty (x^T Q x + u^T R u) dt$$
$$Q = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; R = 50 \tag{15}$$

Again, these four matrices are inputted into the Matlab LQR function to calculate the gain matrix. The poles for the position controller are -6.71,-0.14, and -1.61±0.79*i*.

(4) Altitude Controller

The altitude of the Qball-X4 is maintained through an LQR controller that uses a modified version of Eq. 10 with a state $\dot{s} = Z$ added to provide integrator feedback.

$$\begin{bmatrix} \dot{Z} \\ \ddot{Z} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} Z \\ \dot{Z} \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{4K}{M} \\ 0 \end{bmatrix} u \tag{16}$$

The cost function is defined as follows.

$$J = \int_0^\infty (x^T Q x + u^T R u) dt$$
$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 50 \end{bmatrix}; R = 5 \times 10^6 \tag{17}$$

These four matrices are inputted into the Matlab LQR function to calculate the gain matrix, producing poles at -1.01 and -0.51±0.79*i*.

III. ONBOARD SENSORS AND CONTROL MODIFICATION

The Qball-X4 is equipped with multiple sensors to measure various flight characteristics. Also, the OptiTrack motion capture system was installed in the laboratory to provide additional flight data to the controllers. Finally, a camera was attached to the UAV to build the targeting solution that was necessary for this research

A. SENSORS

The Quanser Qball-X4 incorporates several sensors. Many sensors are included in the original design and are integrated into the HiQ/Gumstix interface. Others, like the OptiTrack Motion Capture System, are completely external, designed to contribute to the control model by providing data collection, feedback of movement, and navigation support. For this experiment, a Logitech camera was attached to the Qball-X4 to provide the source for the image processing model.

1. Installed Sensors

The Qball-X4 contains sensors that are installed by Quanser, including:[16]

- 3-axis gyroscope with range settings of ±75°/s, ±150°/s, and ±300°/s, and a resolution of 0.0125°/s/LSB at the ±75°/s range setting
- 3-axis accelerometer with a resolution of 3.33 mg/LSB
- 3-axis magnetometer, 0.5 mGa/LSB
- 4 Maxbotix XL-Maxsonar EZ3 sonar inputs

The gyroscope and accelerometer readings are used continuously by the pitch, roll, and yaw controllers, providing the main feedback for these controllers to calculate the necessary throttle changes. Additionally, data from the gyroscope and accelerometer feedback is used to calculate the instantaneous states of these rotations, which is used by the controllers and collected as experimental data. The magnetometer provides an accurate heading for the Qball-X4 and is primarily used to give the yaw controller feedback. Additionally, when heading control is in automatic mode, the magnetometer allows the Qball-X4 to maintain a constant heading throughout the experiment. All three of these sensors are located on the HiQ card in the center of the Qball-X4 assembly.

The Maxbotix sonar is mounted underneath the frame of the UAV and is connected to the HiQ by wire. Figure 13 shows the sonar, which provides accurate height above ground for altitudes between 20cm and 765cm with a resolution of 1cm. For altitudes below 20cm, the sonar measures the altitude as 20cm.[16] When height mode is set to automatic, the sonar reading is used to adjust the throttles, allowing the UAV to maintain a preset altitude. For this experiment, the altitude is also used, in combination with the pitch reading, to calculate the altitude of the onboard camera. Also, the sonar is only accurate over a flat surface, and can become inaccurate if it flies over an object. For this reason, the targeting solution requires that the UAV remains behind the target instead of directly over the target.

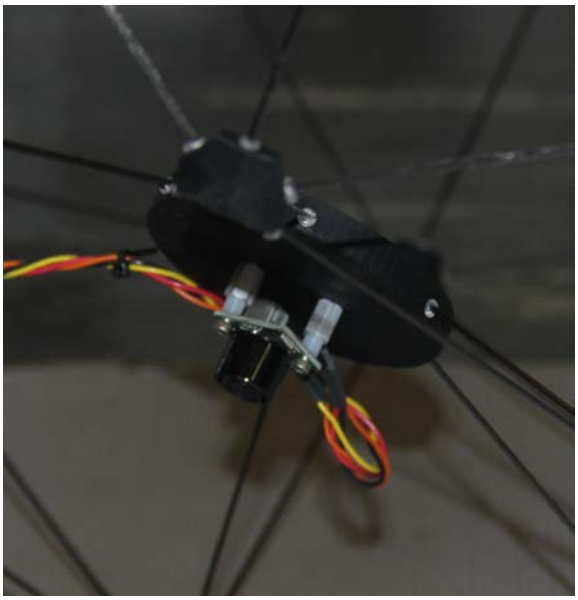

Figure 13. Maxbotix sonar.

2. OptiTrack Motion Capture System

The OptiTrack Motion Capture System, developed by Natural Point, Incorporated, is a system of high-speed tracking cameras that are used for film, sports training, biomechanics, and many other research and entertainment applications.[18] The system consists of a series of cameras that capture the motion of designated reflective

markers, using the motion for data collection as well as real-time calculations. In this experiment, the following Optitrack accessories are used:

- 10 Flex 3 motion capture cameras, each with 0.3 MP resolution, 100 frames per second frame rate, USB 2.0 interface, and 26 LEDs, as shown in the image in Figure 14[19]
- NaturalPoint Tracking Tools software
- Facial 3mm reflective markers
- OptiWand calibration tool
- Calibration Square
- USB hubs and cables

Figure 14. Flex 3 motion capture camera, one of 10 used for this experiment.

The 10 motion capture cameras encircle the Qball-X4 flight area and are mounted about 3 meters above the floor. They are interconnected using the USB hubs and cables and are linked to the ground station computer. Before the experiment, the motion capture system was calibrated per the manufacturer's instructions using the software, OptiWand and Calibration Square. Five reflective markers are connected to the top of the Qball-X4 protective cage, creating a unique signature that allows the cameras to sense the UAV's exact position and dynamics throughout the flight test.

Data from the motion capture system is provided to the Qball-X4 Simulink model, where it can be used in real-time for any designated task. The system provides

accurate three-dimensional position data, with x- and z-axes based on the calibration, and the y-axis based on the height above ground. The cameras can also measure the velocity and acceleration of the Qball-X4 within those axes. The Optitrack motion camera system provides the pitch, roll, and yaw angles as well as the heading of the Qball-X4.

The goal of this experiment is for the Qball-X4 to track the target solely based on the target's position and movement. For this reason, the real-time capability of the motion capture system is not used when programming the controllers. The OptiTrack system is used during lift off, allowing the UAV to stabilize at its prescribed altitude before tracking mode is activated. The OptiTrack also provides important data collection capability, allowing for the analysis of several flight parameters.

3. Onboard Camera

The ideal onboard camera for this research is one that is lightweight, wireless, and capable of seamlessly streaming quality images to the ground station at a high frame rate. Several attempts were made using the Trek Ai-Ball camera, which is extremely light as well as wireless. The Ai-Ball camera operates by transmitting a series of images to an IP address. Attempts to establish a direct TCP/IP link between the camera and ground station were unsuccessful. There was success in streaming the images to an IP address then uploading this image into a Matlab script, but this method slowed the model to between one and two frames per second, much slower than what was acceptable for this experiment.

The experiment used a wired camera, which was capable of attaining the frame rates necessary for the research. The Logitech Webcam Pro 9000 was attached to the front of the Qball-X4, as shown in the image in Figure 15. The Webcam Pro 9000 is a 2 MP camera that has a focal length of 2 mm and a 75° diagonal field of view. It is capable of capturing a 1600x1200 still or video image. At the reduced resolution of 800x600, the Webcam Pro 9000 can capture video at a rate of 30 frames per second.[20] The model used in this research processed a 320x240 image that was streamed to the ground station at 10 frames per second. The camera was attached to the front of the z-axis crossbar with two zip ties.

Figure 15. Logitech Webcam Pro 9000, attached to the front of the Qball-X4.

The base of the camera was removed prior to attaching it to the Qball-X4, reducing its mass to 50 grams. A USB 2.0 cable sends its data to the ground station. The cable is wired underneath the z-crossbar to the rear motor, then drops below the frame. As a result, special care must be given to ensure that the cable does not come into contact with the propellers during flight. A USB extender cable is used to ensure that an excess of slack is present during flight operations.

B. MODIFICATIONS TO CONTROL SCHEME

The Qball-X4 control model is a real-time Simulink model that controls all aspects of the UAV flight. The control model, shown in the illustration in Figure 16, consists of 11 subsystems that will be discussed in detail in this section. The Targeting Solution subsystem was created specifically for this research. The other 10 subsystems were designed by Quanser, but some modifications were made to integrate the image processing and targeting tasks.

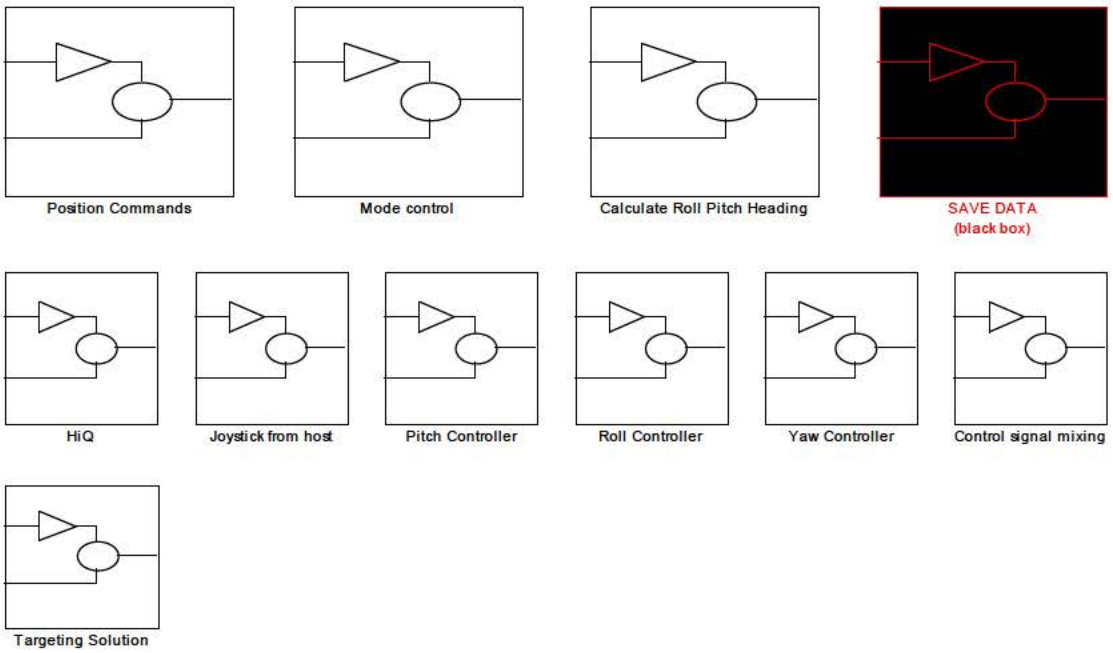

Figure 16. Qball-X4 Control Model main page, which contains all major subsystems of the control model.

1. Joystick Controller

To run the control model, the separate Joystick Controller model must run in the background. The Joystick Controller, shown in the image in Figure 17, receives manual mode inputs from the joystick that correspond to the roll, pitch, yaw, and throttle controls in that mode. Additionally, the controller includes a safety switch for the throttle that can force the UAV to land even in the automatic modes. The Joystick Controller also receives data from the OptiTrack motion control system. All data is gathered and sent to the master control model via TCP/IP connection.

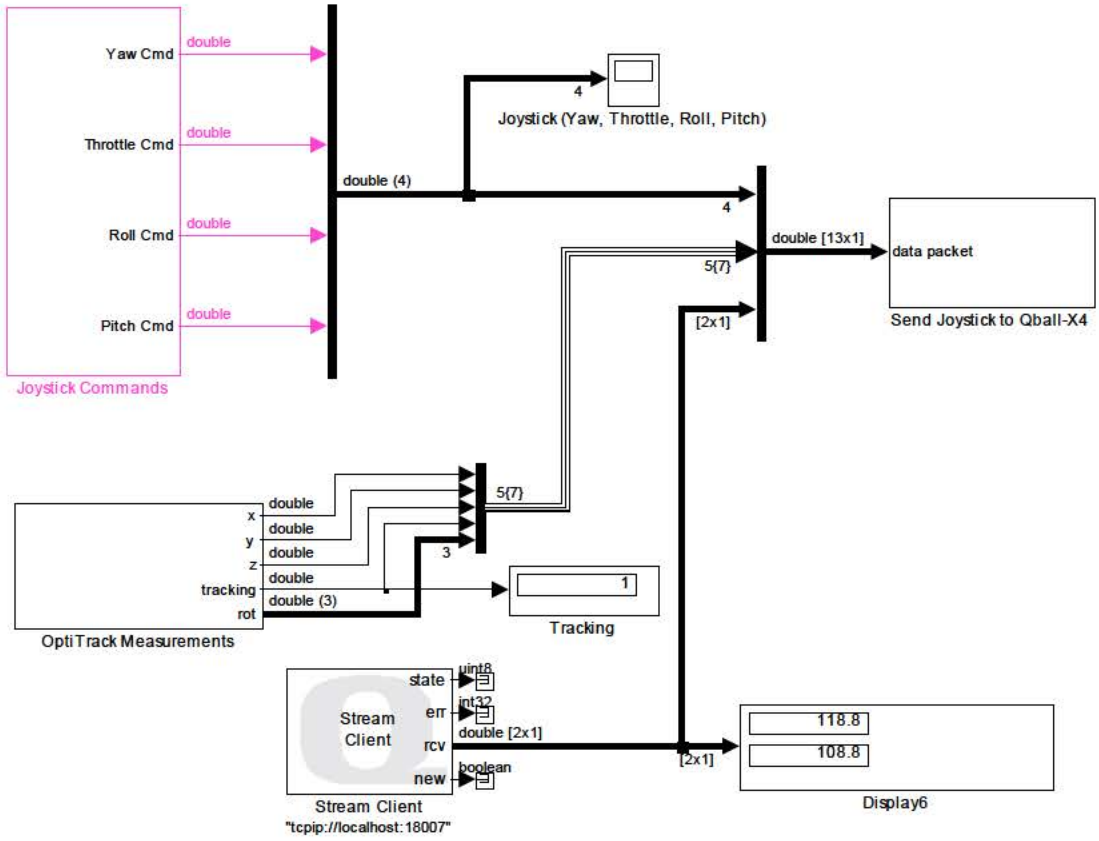

Figure 17. Joystick Controller Simulink model, which is separate from the main control model.

This model was modified to include two camera inputs, which are received into this model from the Image Recognition model using a TCP/IP connection. The two signals are the two pixel numbers, row and column, that represent the center of the target as seen from the onboard camera of the UAV. These values are joined with the joystick and OptiTrack signals and sent to the Qball-X4 Control Model for further processing.

2. Joystick from Host Subsystem

The Joystick From Host subsystem, which is shown in the image in Figure 18, receives the 13-port signal directly from the Joystick Controller Model and splits it, relaying the individual signals to a series of tags that are used by the other subsystems. The first four signals are the roll, pitch, yaw, and throttle commands from the joystick. The next seven signals contain the three position coordinates and three angular positions

of the Qball-X4, along with a signal that verifies the presence of the UAV. Finally, two pixel indices are sent that correspond to the row and column of the target. The pixel indices are a modification of the original subsystem, as they are used by the Target Recognition subsystem to calculate the relative position of the target.

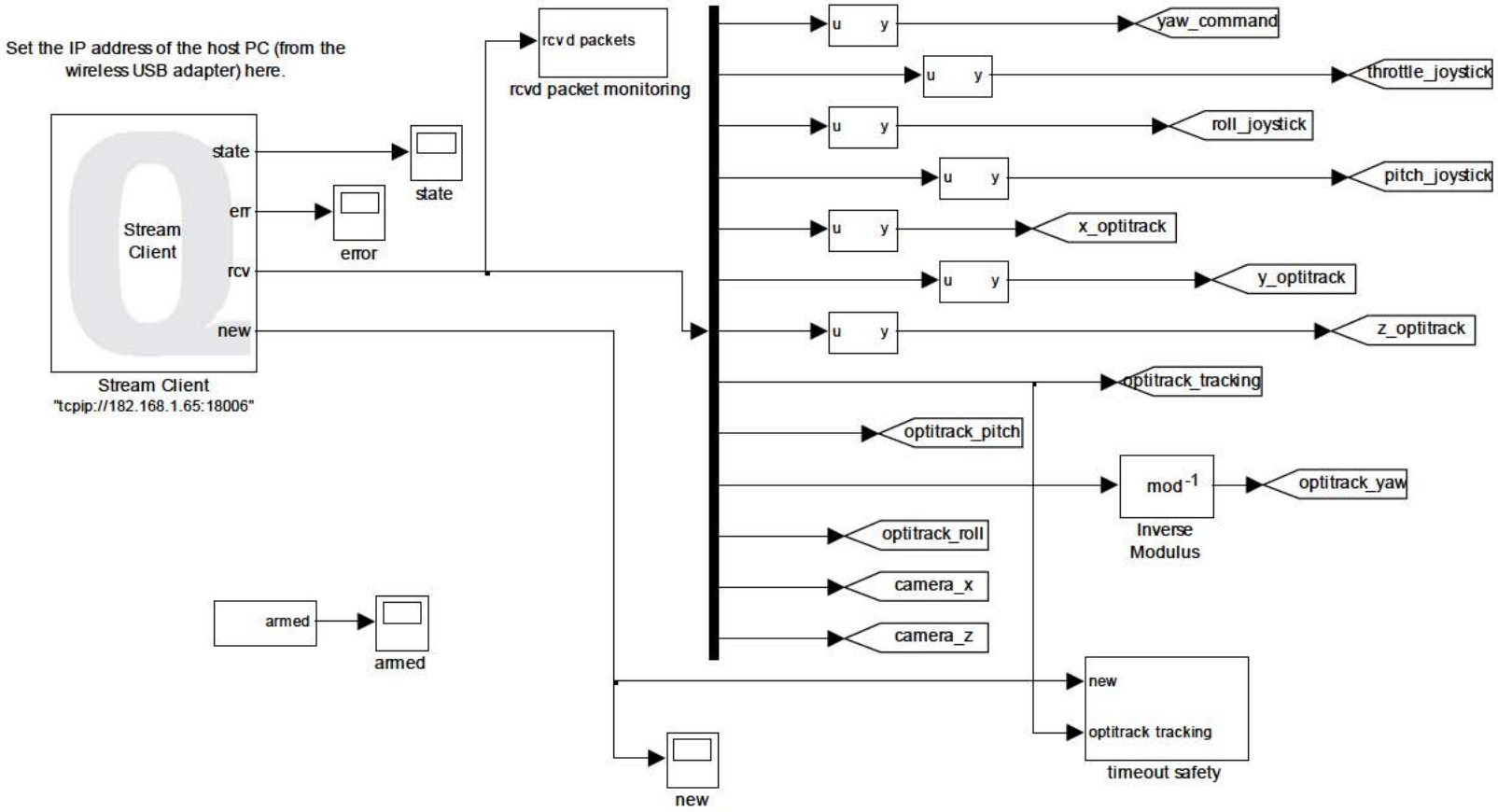

Figure 18. Joystick From Host subsystem, which receives inputs from the Joystick Controller model.

The verification of presence of the UAV is used by the timeout safety subsystem, which alerts the user if no vehicles are being tracked by the OptiTrack system. This safety subsystem also displays a warning message if a signal spike is received or if the signal stops streaming from the Joystick Controller model.

3. Mode Control Subsystem

The Mode Control subsystem, which is shown in the diagram in Figure 19, allows the user to switch between manual and automatic control for four different modes of flight. Height mode allows the model to control the throttle speeds as necessary to maintain a preset altitude, unless the throttle input on the joystick is set to zero, which will force the Qball-X4 to land. Position mode allows the model to control the pitch and roll of the UAV to maintain a preset position. Heading mode causes the model to control the yaw of the Qball-X4 to maintain the programmed heading.

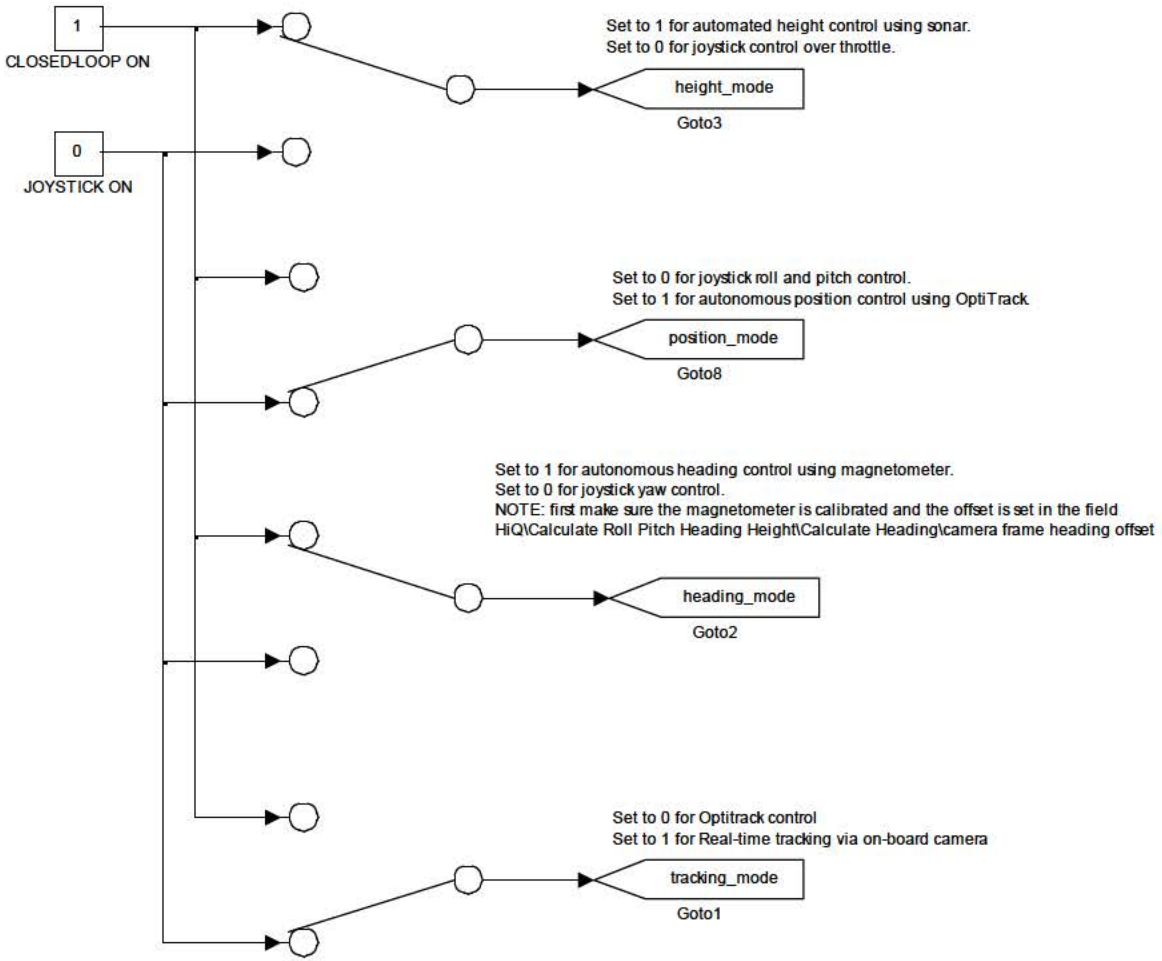

Figure 19. Mode Control subsystem, which allows the user to adjust the level of autonomy during flight.

The tracking mode was added to the model to support this research. This mode, when activated, allows the model to control the pitch and roll of the vehicle as necessary to maintain the desired offset angle from the target. When this mode is off, the model will control the UAV to maintain the designated position using the OptiTrack motion capture system. For the experimental portion of this thesis, all modes are activated to allow the model to completely control the Qball-X4 for the tracking problem.

4. Position Commands Subsystem

The purpose of this subsystem, which is displayed in Figure 20, is to provide operating parameters for the automatic modes of the model, as designated by the Mode Control subsystem. The desired heading is entered in the heading command block and sent to the Yaw Controller subsystem. The commanded altitude is entered using a slider gain and sent to the Throttle Switching subsystem where it is compared with the actual altitude and used to adjust the throttles accordingly, allowing the UAV to maintain the commanded altitude.

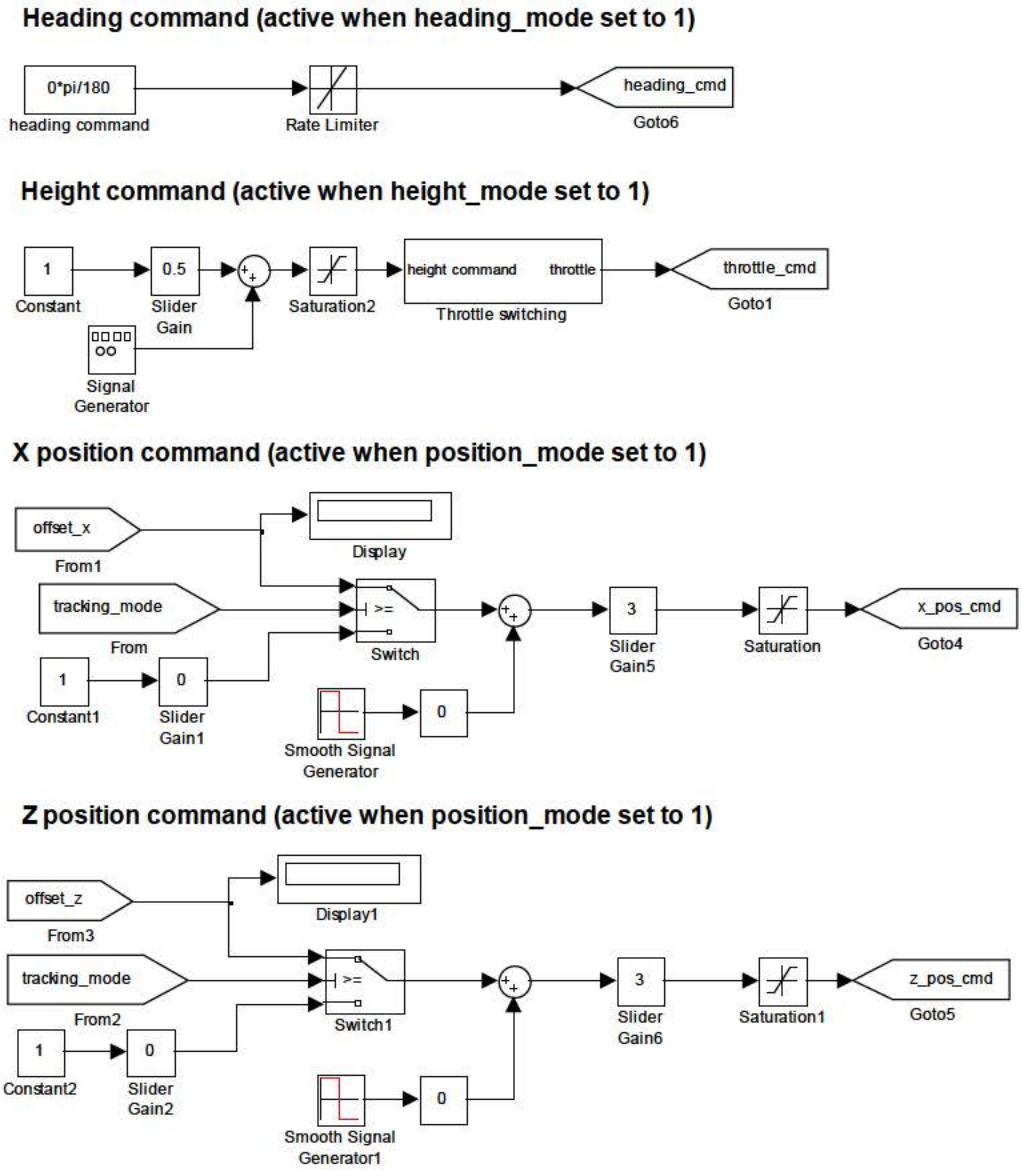

Figure 20. Position Commands subsystem, used for automatic modes.

The X and Z position commands are based on which tracking mode is selected, and have been modified from the original model. When OptiTrack is selected, the UAV will fly to and maintain the lateral position selected by the sliders. When tracking solution is selected, the UAV will fly to and maintain the desired offset from the target. This offset is calculated in the Target Recognition subsystem, based on the inputs from the onboard camera. In either case, a position command is sent for each axis to the roll

and pitch controllers, allowing the model to calculate the necessary roll and pitch control inputs.

5. Calculate Roll Pitch Heading Subsystem

The Calculate Roll Pitch Heading subsystem, which is shown in the illustration in Figure 21, is an unmodified subsystem that computes all three angles using the onboard gyroscopes and accelerometers. Two of its subsystems are Calculate Roll and Calculate Pitch, which calculate the respective angle using accelerometers. These angles are used in the controllers as well as the Targeting Solution subsystem. Calculate Heading computes the magnetic heading using the magnetometer readings, allowing the Heading Mode to maintain the desired heading. These three subsystems are displayed in Figure 22, Figure 23 and Figure 24.

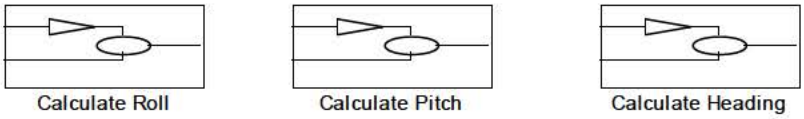

Figure 21. Calculate Roll Pitch Heading subsystem, which contains each measurement in a separate subsystem.

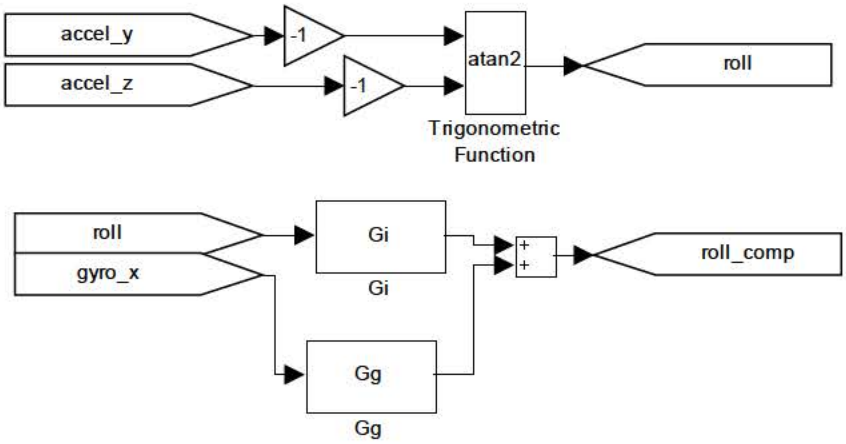

Figure 22. Calculate Roll subsystem.

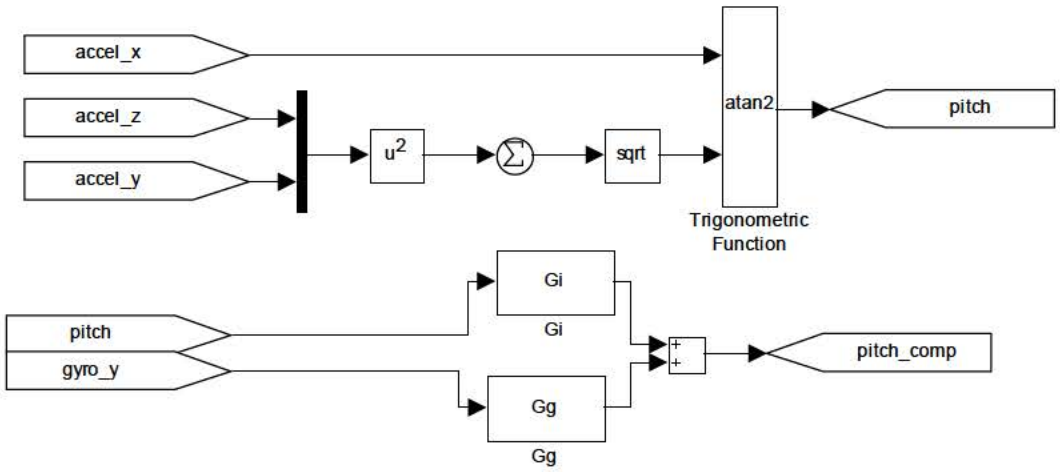

Figure 23. Calculate Pitch subsystem.

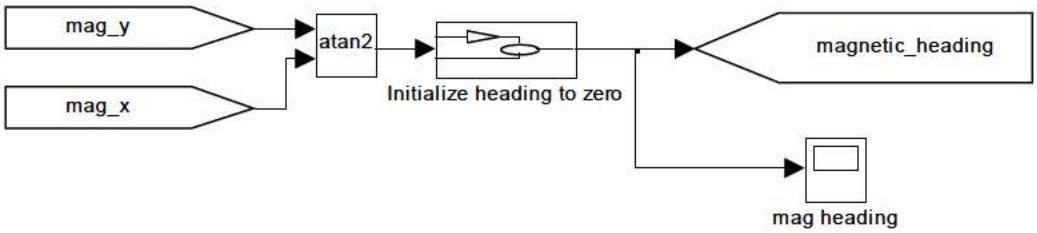

Figure 24. Calculate Heading subsystem.

6. Pitch/Roll Controller Subsystems

The Pitch and Roll Controller subsystems control the pitch and roll of the Qball-X4. Since both controllers behave in exactly the same manner, this section will focus on the Pitch Controller. The illustration in Figure 25 shows the main Pitch Controller subsystem, which consists of two parts: pitch command and pitch control. The pitch command system determines the command from the joystick, OptiTrack system, or targeting solution system. This command is given to the controller, where it is compared to the gyroscope and current throttle command to calculate an input for the Control Signal Mixing subsystem.

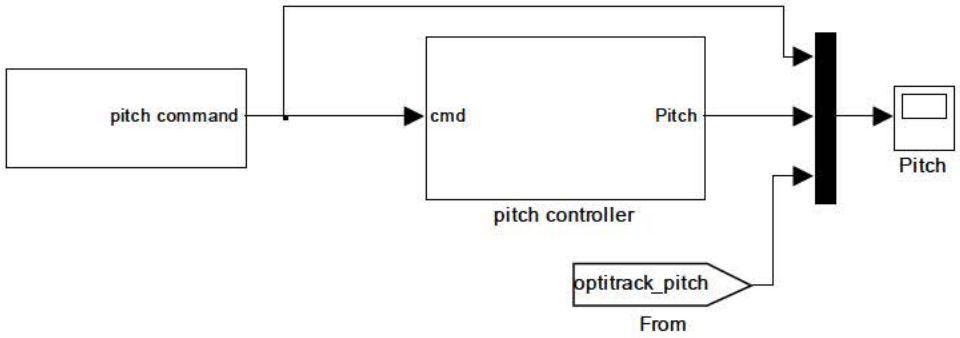

Figure 25. Pitch Controller subsystem, which contains the Pitch Command and Pitch Control subsystems.

(1) Pitch Command

The Pitch Command subsystem, which is shown in the illustration in Figure 26, determines which type of command will be relayed to the pitch axis based on the settings in the Mode Control subsystem. When position mode is set to "0," the joystick directly controls the pitch axis. When position mode is activated, either OptiTrack or Targeting Solution will determine the pitch input. The input is then routed through a safety switch that checks the throttle position, then to the pitch controller to carry out the command.

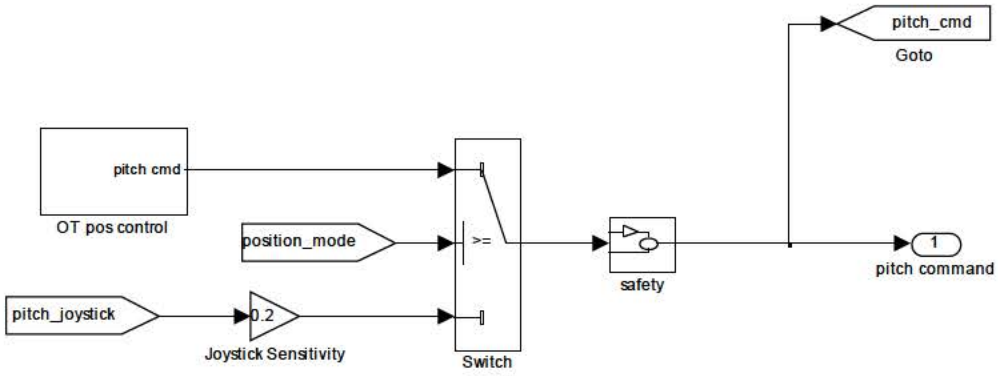

Figure 26. Pitch Command, part of the Pitch Controller subsystem.

The "OT Pos Control," or OptiTrack Position Control pictured in Figure 27, is the modified subsystem that determines the pitch command to be sent to the pitch controller when position mode is activated. When Targeting Mode is not active, the position command is calculated as the difference between the current position on the z-axis and the z-position commanded in the Position Commands subsystem. This difference represents the error that will be corrected by the controller. When Targeting Mode is active, the z-offset is inputted as the error. This allows the Qball-X4 to make the necessary corrections based on its position relative to the target. The error value is routed through a rate limited controller to prevent abrupt movements. Finally, the value is adjusted as necessary to account for variations in heading before being sent to the control subsystem.

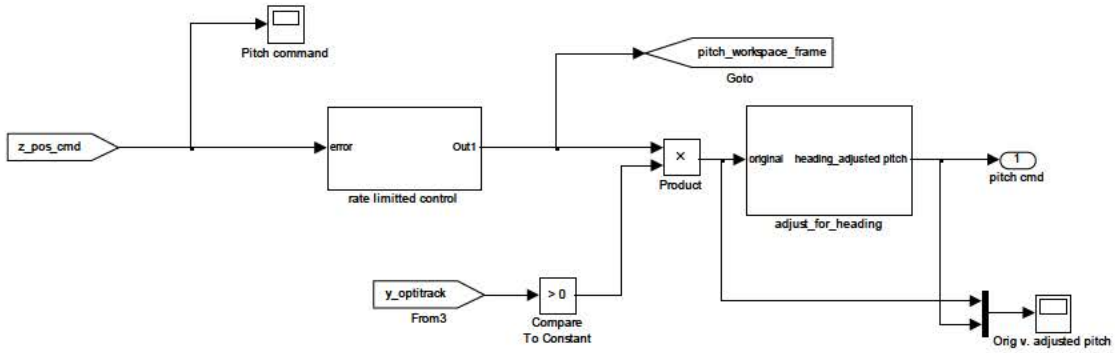

Figure 27. OT Pos Control, part of the Pitch Command subsystem.

(2) Pitch Control

This subsystem, shown in the illustration in Figure 28, takes the command input and calculates the control input for the Control Signals Mixing subsystem, in accordance with the control law that is loaded into the system. The main components of the control input are the pitch command, y-axis gyroscope setting, throttle command, and a feedback input.

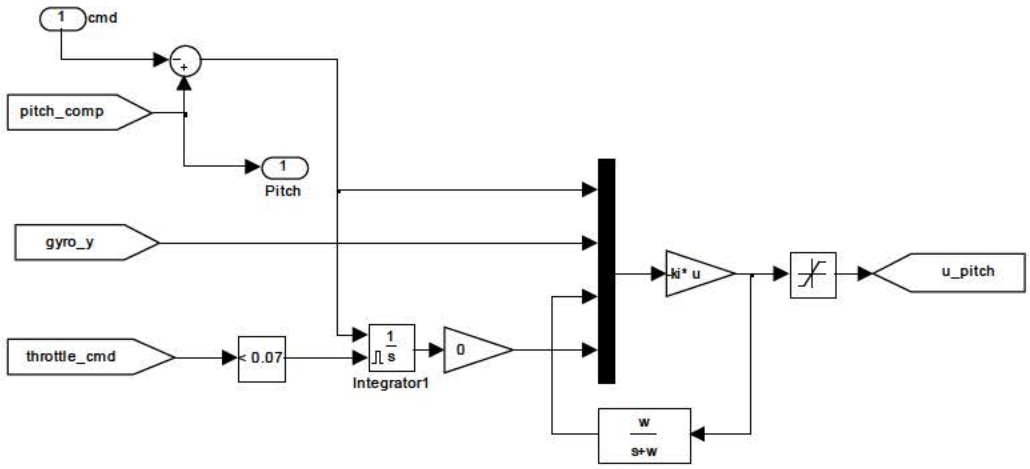

Figure 28. Pitch Control, part of the Pitch Controller subsystem.

7. Yaw Controller Subsystem

The Yaw Controller, which is displayed in Figure 29, is designed to turn the UAV to the desired heading, as set in the Position Commands subsystem. To accomplish this, the observed heading is computed and compared against the commanded heading to calculate the error. The z-axis of the gyroscope is also added to this error to account for how much the heading is changing. A gain is applied to each input, resulting in a command to change the throttles in the Control Signal Mixing subsystem. A saturation block is added to prevent the UAV from applying too much correction, as this may lead to uncontrolled flight.

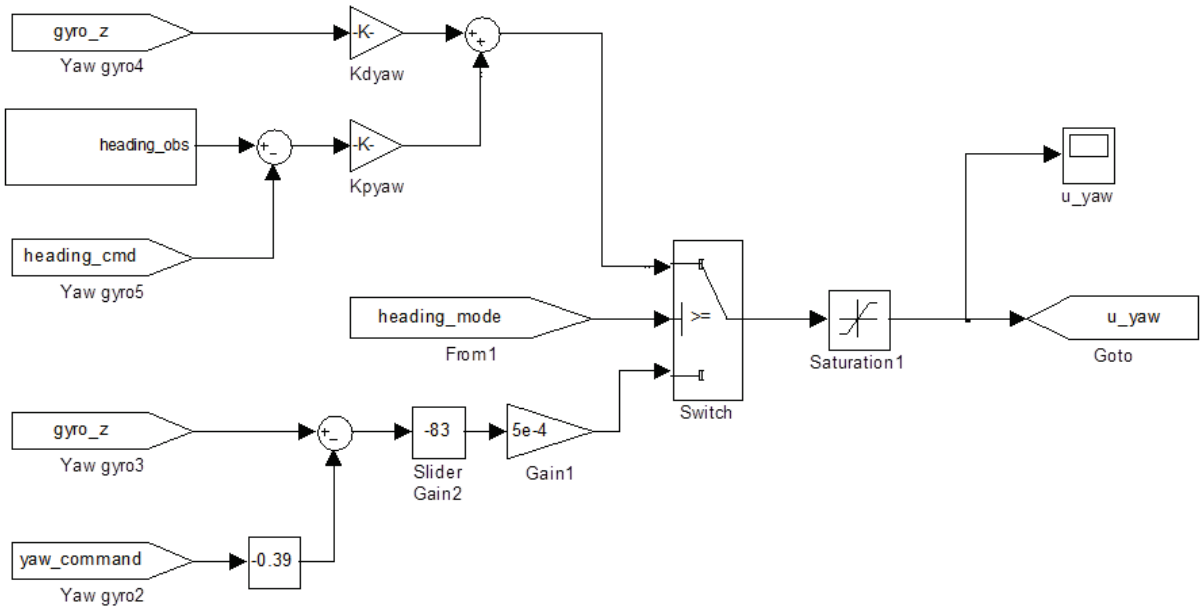

Figure 29. Yaw Controller subsystem.

This subsystem is also used when heading mode is set to manual, which activates the bottom portion of the subsystem. In this case, the gyroscope is compared to the yaw command of the joystick and the throttles are adjusted as necessary to match the two values. This experiment used automatic heading mode for all tests and did not require any modifications to this subsystem.

8. Control Signal Mixing Subsystem

After each controller computes their control command, the commands are sent to the Control Signal Mixing subsystem. This subsystem receives commands for roll, pitch, yaw, and throttle, combining them as necessary to translate the commands to motor output. The illustration in Figure 30 shows this subsystem, with the four motors representing the back, front, left, and right motors in that order. The pitch command gives opposing inputs to motors 1 and 2, while the roll command gives opposing inputs to motors 3 and 4. Any yaw command will give a certain input to the front two motors while giving the opposite input to the left and right motors. Finally, the throttle command is applied to all four motors equally. This subsystem, which matches the model in Chapter II, required no modifications.

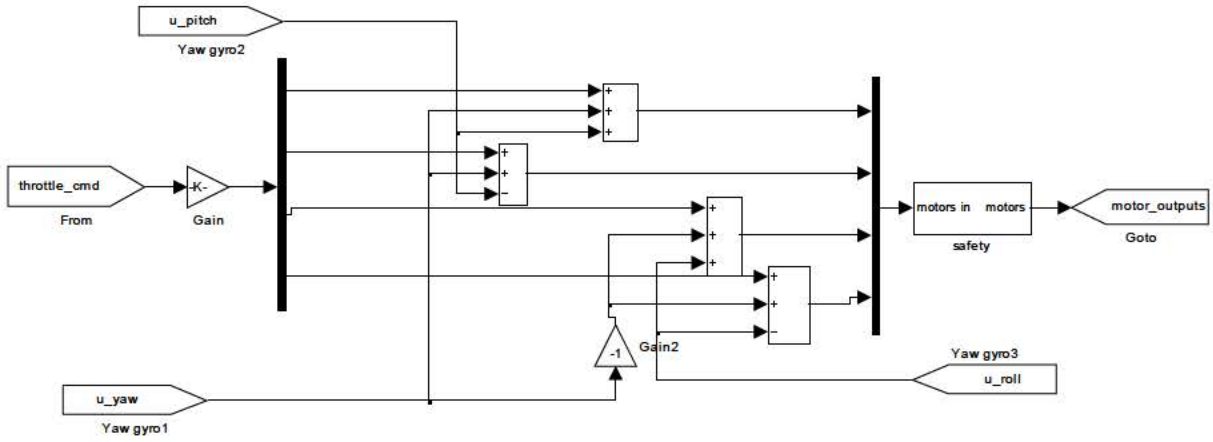

Figure 30. Control Signal Mixing subsystem

9. HiQ Subsystem

This subsystem, which is shown in the illustration in Figure 31, is the main interface between the ground station and the Qball-X4's data acquisition card. It takes the motor outputs from the Control Signal Mixing subsystem and sends them as a PWM signal to the Qball-X4, which then sends the inputs to the motors via the speed controllers. The "Gain" block allows the user to enable or disable the motors; setting the gain to zero allows for a test of all sensor systems without flying the UAV. This subsystem also receives the following sensor inputs from the UAV: gyroscope, accelerometer, magnetometer, sonar, and available battery voltage. The battery voltage block is used to send a message to the user if the battery is running low. The other sensors are used by the various controllers including the Targeting Solution controller.

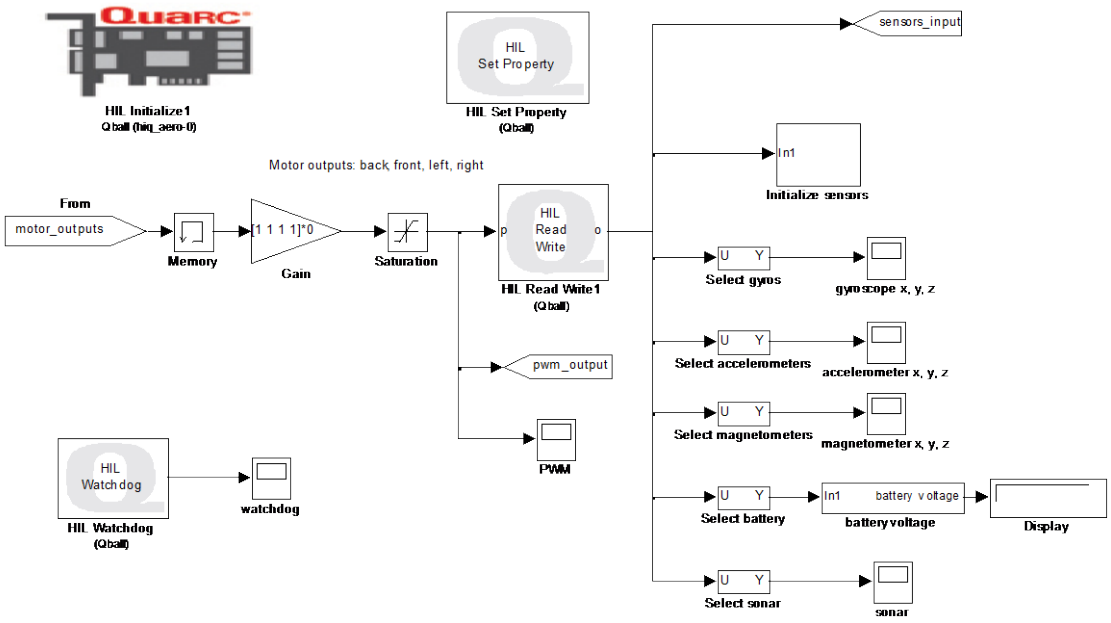

Figure 31. HiQ subsystem.

10. Save Data Subsystem

The Save Data subsystem is the "black box" of the control model. Figure 32 displays this subsystem, which saves all data including position data and angular position data to a file in the active Matlab directory. It also saves commands, such as joystick, OptiTrack, and targeting commands. The data is compiled into matrix form and includes a time stamp, allowing for easy analysis. The file is named using the current date and time to ensure that each flight test generates a unique data file.

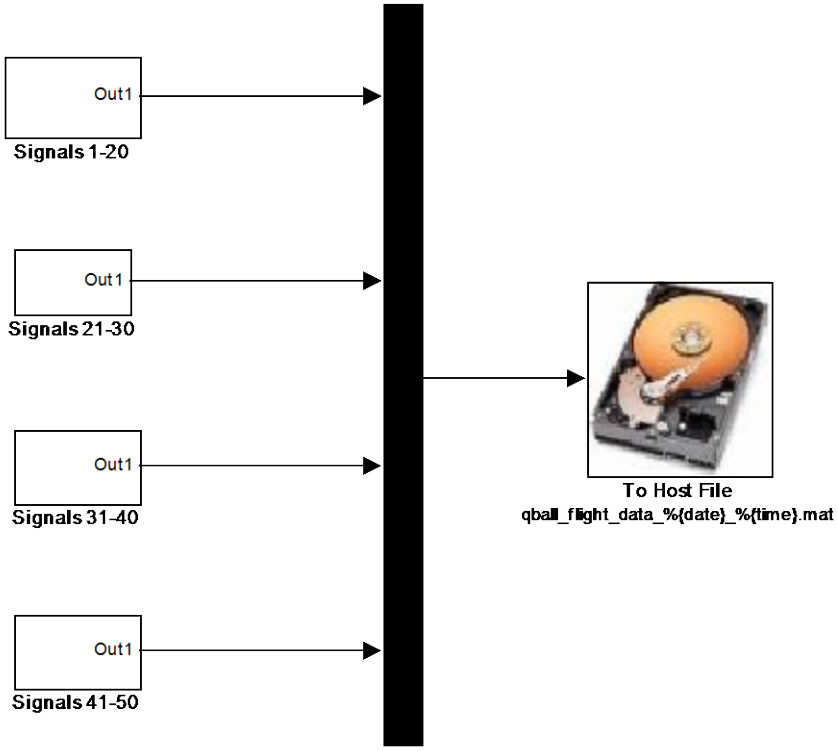

Figure 32. Save Data subsystem, the "black box" of the control model.

C. TARGETING SOLUTION SUBSYSTEM

While many parts of the control model were modified, an entirely new subsystem was created to convert the pixel indices to coordinates. The Targeting Solution subsystem is illustrated in Figure 33 and contains five inputs and two outputs. The first two inputs are the pixel indices, which are values between 0.01 and 2.40 that represent the pixel location of the target. Chapter IV discusses the process that obtains these values in detail. The third input is the sonar input, which represent the altitude of the UAV. It is important to note that this value is only accurate above 20 cm. The fourth and fifth inputs are the pitch and roll angles, which are used to correct the targeting angle to ensure that it remains accurate when the Qball-X4 pitches and banks. The two outputs represent how far the target is away from its goal position, in meters. These values are sent to the controllers to derive the necessary throttle connections, driving the UAV towards its desired offset position. Chapter IV details the calculations used to derive these offset values.

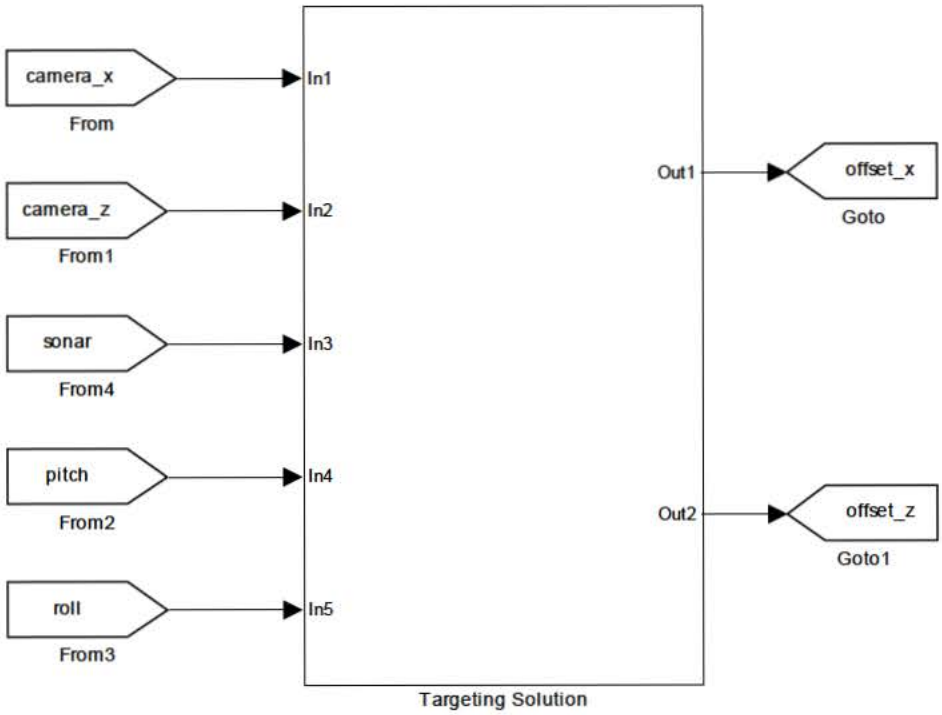

Figure 33. Targeting Solution subsystem main page.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. IMAGE PROCESSING AND REAL-TIME TRACKING SOLUTION

Digital imaging is the process where an image is converted to a set of numbers, allowing the image to be archived, altered, displayed, or printed. The most basic element of a digital image is the pixel, which is a short for picture element.[21] A pixel is a value that represents a color for one small point of the image. When a matrix of pixels is constructed, the pixels combine to create the image. The illustration in Figure 34 shows an example of a simple black-and-white image. The picture on the left can be reconstructed as the 10x10 matrix on the right, with "1" representing black and "0" representing white.

0	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	0	1
1	1	0	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	0

Figure 34. Representation of a black-and-white image as pixels.

The image in Figure 34 is considered a binary image, since every pixel must be fully black or fully white. A typical black-and-white image, however, consists of colors between the extremes, otherwise known as grayscale. Figure 35 contains a grayscale gradient, which consists of 256 different grays. By assigning a value of 0 to 255 to every pixel, each individual pixel can be stored in a single byte, or 8 bits, of memory. Additionally, the human eye only has the ability to distinguish approximately 200 gray levels, so there is no advantage to increasing the total number of levels.[21]

Figure 35. Example of a grayscale gradient (from http://www.crestock.com/uploads/blog/2011/calibration/good_gradient.JPG).

To create a color digital image, the RGB color system is commonly used. RGB, or red-green-blue, represents the primary colors of light. By combining two colors, a secondary color can be created: red and green combine to make yellow, red and blue combine to make magenta, and red and green combine to make cyan. A combination of all three colors results in white and the absence of all three colors results in black. Figure 36 shows these color combinations.

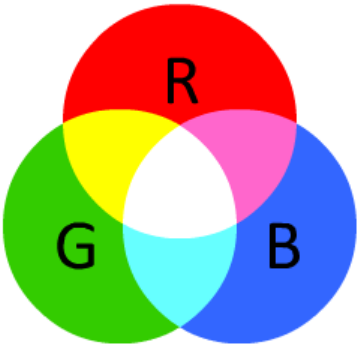

Figure 36. The primary and secondary colors of light (from http://www.kirupa.com/images/rgb_image.png).

Similar to the gray levels, each color can be divided into 256 levels, as shown in the illustration in Figure 37. When these colors are combined, there are about 16.7 million different possible colors.[21] Furthermore, each color can be stored using three bytes of storage, or 24 bits. To digitize this image, three matrices can be used: red, green, and blue. Each pixel requires three values, one for each color. These matrices can be combined into one three-dimensional matrix to represent the entire image.

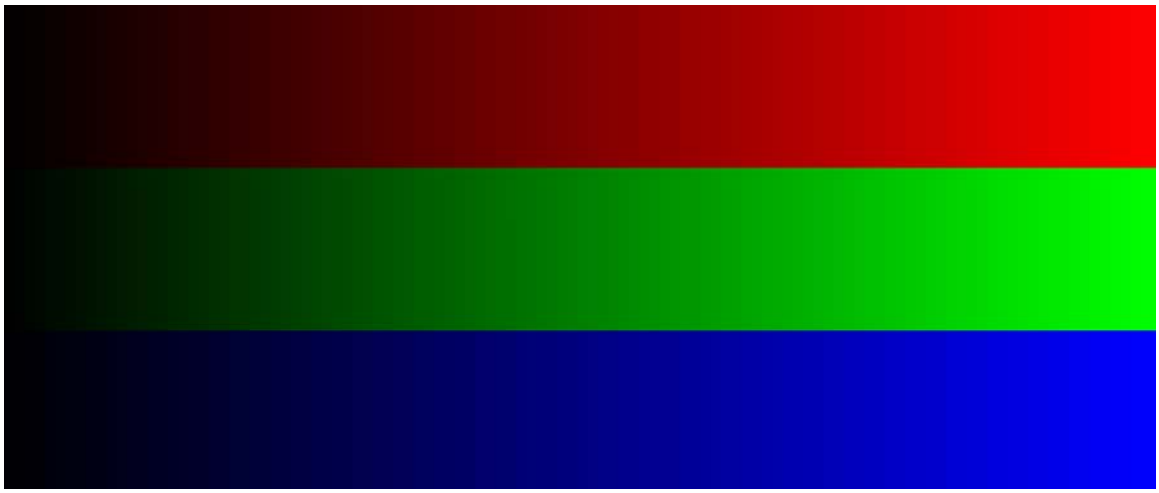

Figure 37. RGB levels (from http://www.guicarmail.it/Television/SMPTE%20ColourBars/8bit_full_grad_color.png).

The resolution of an image refers to the total number of pixels, represented as the number of rows and columns of the pixel matrix. For example, a color image may have a resolution of 640x480, meaning the image is represented as a 640x480x3 matrix, or 307,200 pixels per color. As a result, this image would require 921,600 values to be stored. A 320x240 image, on the other hand, only requires 230,400 total values. There are many factors such as available storage space and processing power that must be considered when selecting a resolution. For this experiment, the image resolution was 240x240 or 172,800 total values, which was sufficient for the target to be distinguished from its surroundings without exceeding the processing power limits of the experiment.

A. TARGET RECOGNITION—FROM IMAGE TO PIXEL INDICES

The first step in creating a targeting solution for the Qball-X4 is determining the relative bearing to the target using the onboard video camera. Since a video is a series of digital images, each image must be processed to determine the location of the target on that image. Furthermore, since the camera is in a fixed position on the front of the UAV, the target's location on a digital image can be used to calculate the exact bearing of the target. The diagram in Figure 38 shows the model used to capture the image and locate the target. Figure 39 illustrates the Target Recognition subsystem, which recognizes the

target and outputs the pixel indices. The indices are sent to the Qball-X4 Joystick Controller model using a TCP/IP connection.

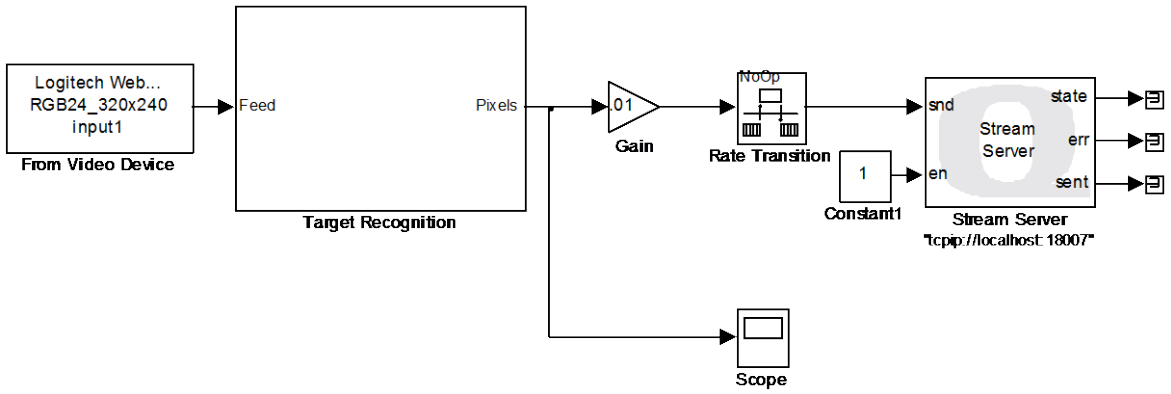

Figure 38. Image Capture model.

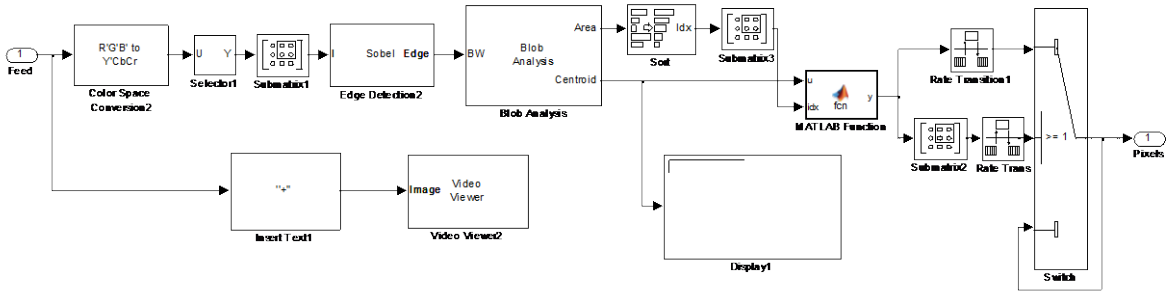

Figure 39. Target Recognition subsystem of the Image Capture model.

1. Defining the Target

For this experiment, the target is a small remote-controlled car. The car and remote controller are battery powered and do not interface with the Qball-X 4 or with the ground control station. The car, shown in the image in Figure 40, is covered in masking tape and has a green piece of tape on the top surface of the car. This helps the model to recognize the target without requiring excessive processing power.

Figure 40. Remote-controlled car, used as the target in this experiment.

Before the image can be processed, it must first be captured. This is accomplished using Simulink's Image Acquisition Toolbox, which brings the image into the model as a 24-bit, 320x240 digital image. The picture is a color RGB image that is represented in the model as a 320x240x3 matrix. Figure 41 shows an example of this image, with the target near the center of the camera's field of view. A subdued set of crosshairs was added to the image on the model to show the center of the camera's view, which is also the UAV's desired angle when tracking the target.

Figure 41. Original camera view prior to image processing.

2. Color Space Adjustment

The image is captured in RGB color space, which can be challenging for image processing. Figure 42 shows the image divided into separate red, green and blue channels, respectively. As the set of images shows, there is very little difference between the three channels. The green tape does appear much lighter in the center (green) channel, but the rest of the images look nearly identical.

Figure 42. Camera view partitioned to red, green, and blue channels.

To aid in target recognition, the image is then converted to Y'CbCr space. The following formula completes this transformation for each pixel.[22]

$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.369 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{18}$$

Y'CbCr color space places the luminance of the image in the Y matrix while placing the blue and red chrominance values in the Cb and Cr matrices, respectively. This color space is often used by photographers since the chrominance matrices tend to be easily compressible.[23] The color space can be valuable for image processing as well, since contrasts between certain colors can be easier to detect. Figure 43 shows the same set of images, but in the luminance and two chrominance channels.

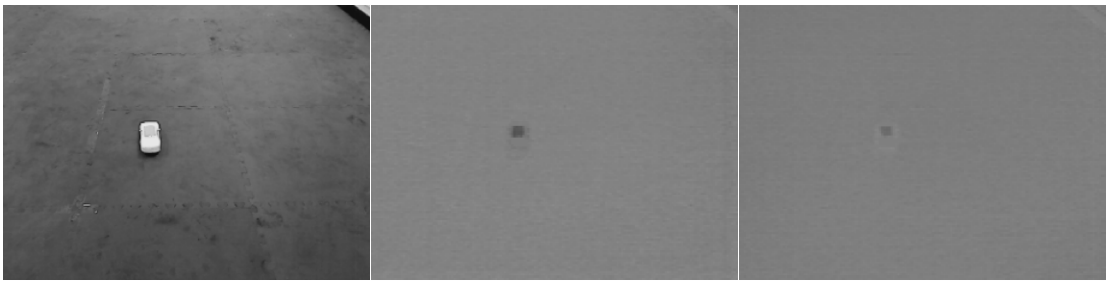

Figure 43. Camera view partitioned in Y, Cb, and Cr channels, respectively.

Both chrominance channels allow the target to be distinguishable while making the rest of the image blend together. The blue chrominance, or Cb, channel provides a very easy to detect target, which helps to improve the precision of the targeting solution while reducing the chances of false detections. For this reason, the Cb channel is isolated from the image. Additionally, the image is cropped from 320x240 to 240x240, reducing the image to a single matrix with 57,600 values. This allows for faster and smoother processing of the image.

3. Sobel Edge Detection

After the image is converted to the desired color space, it must be analyzed to locate the target. To accomplish this, edge detection is applied to the image. The purpose of edge detection is to highlight any discontinuity in image brightness, normally through the use of a filter.[24] One common method of edge detection is through the use of the Sobel operator, which is represented by the following two matrices.[25]

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \mathrm{G}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{19}$$

The Sobel method applies the operators to each pixel in the corrected image, deriving the gradient component in horizontal and vertical orientations. Next, the absolute magnitude is calculated for each pixel using these gradients.[26]

$$|G| = \sqrt{G_x^{\,2} + G_y^{\,2}} \tag{20}$$

The resulting matrix contains mostly low numbers where there is no color change, but also contains a few much higher numbers where the color difference is much larger. To

filter out the low values, the edge detector applies a threshold scale factor of 20 to the adjusted image. This is done by assigning a value of 1 to pixels with a gradient above 20 and assigning a value of zero to the rest of the pixels. Figure 44 contains the resulting binary image, with the high gradient pixels appearing as white.

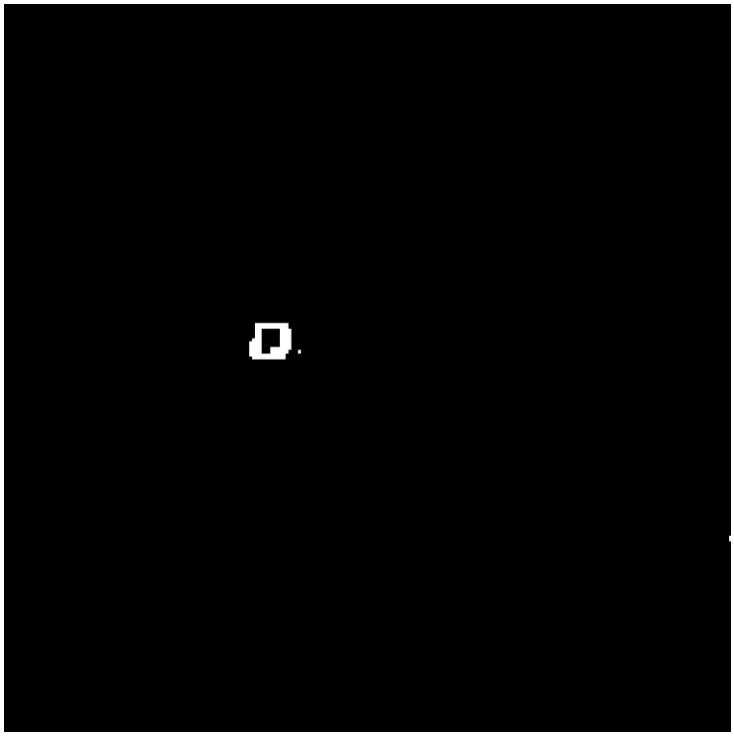

Figure 44. Camera view after Sobel edge detection and filtering.

4. Blob Analysis

To further analyze the image, the model implements Matlab's blob analysis function. The purpose of blob analysis is to compute various statistics for connected regions in any binary image.[27] While Matlab is capable of producing a number of statistics for the image, only centroid location and area are used for this model. Centroid location returns the center of all blobs that are detected , and area returns the area of the blob in pixels. The block is configured to not accept any blob that touches the edge of the image. Additionally, the block is configured to filter out any centroid that has an area of less than 50 pixels or more than 300 pixels.

In this example, the target is the only blob detected by the model that meets all criteria. Table 2 contains the pixel indices of the target as calculated by this model for 10 seconds. For both coordinates the precision is very good, as each pixel index fluctuates by no more than 0.4 pixels.

Coordinate	Average	Range
X	88.86	88.5-89.2
Y	111.8	111.5-112.0

Table 2. Pixel indices for target in the above camera view.

5. Other Considerations

In the case where the camera does not detect any blobs during a single frame, the model would normally reset to zero for each pixel count. This reading could cause the targeting solution model to momentarily place the target in the upper-left corner of the camera view. Since this could lead to a drastic correction by the controller, the model is designed to reuse the previous input if no new input is given.

The spike detection is an important safety feature in the model, designed to detect faults in the joystick and OptiTrack system by monitoring every signal for a reading greater than 10. To prevent the pixel indices from triggering the spike detection, both indices are divided by 100, resulting in a reading of 0.01 to 2.40 for each axis. Finally, the indices are sent to the Joystick Controller model via TCP/IP connection, where they are combined with the throttle, axis, and Optitrack camera inputs and sent to the control model.

B. TARGETING SOLUTION—FROM PIXEL INDICES TO OFFSET

The goal of this section is to convert the pixel indices of the target into a measurement of how far the target is from the center of the UAV's view at any given time. This is accomplished through a series of calculations that include some of Qball-X4's flight characteristics. The illustration in Figure 45 shows the Targeting Solution subsystem, which derives an accurate offset value, in meters, from the pixel indices, the sonar altitude measurement and the UAV's pitch and roll angle measurements.

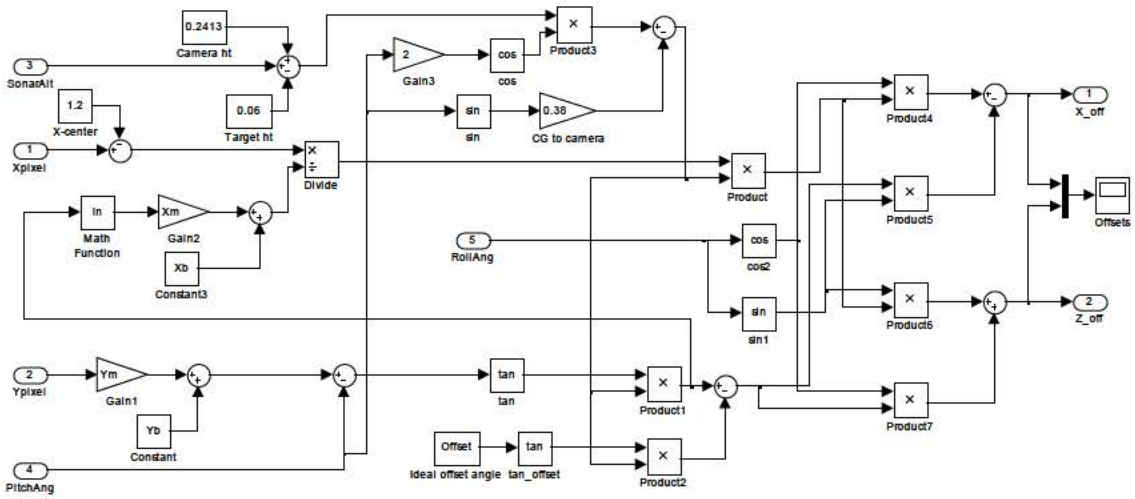

Figure 45. Targeting Solution subsystem.

1. Defining the Relative Position Space

The axes for the position space are defined differently than the model. Figure 46 shows the six degrees of freedom of the Qball-X4. The positive X direction protrudes from the right side of the UAV, while the positive Y direction points directly above it. Adhering to the right-hand rule, the Z direction points behind the Qball-X4. The angles are defined as follows: positive roll (ϕ) is a bank to the right, positive pitch (θ) is a forward pitching motion, and positive yaw (ψ) is a rightward turn. The angular directions match those measured by the Qball-X4's sensors.

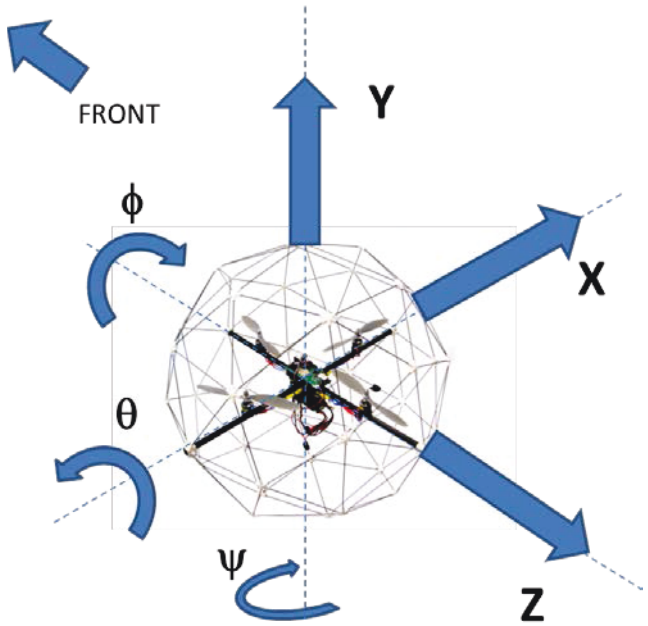

Figure 46. Position and rotational axes of the Qball-X4.

All definitions for the relative location of the target, shown in the illustration in Figure 47, are based on the relative origin, which is defined as the point on the ground that is directly below the Qball-X4's onboard camera and based on closest vertical spacing between the camera lens and the ground. The height above target (h) is the vertical distance between the camera lens and the height of the target and is explained in further detail later in this section. The Z-distance of the target (z) is the distance between the origin and the target's position on the Z-axis. The Z-angle (α) is defined as the angle between the Y-axis and the direct line-of-sight (LOS) between the camera and the target. The X-distance is defined as the horizontal distance between the Z-axis and the target's position on the X-axis.

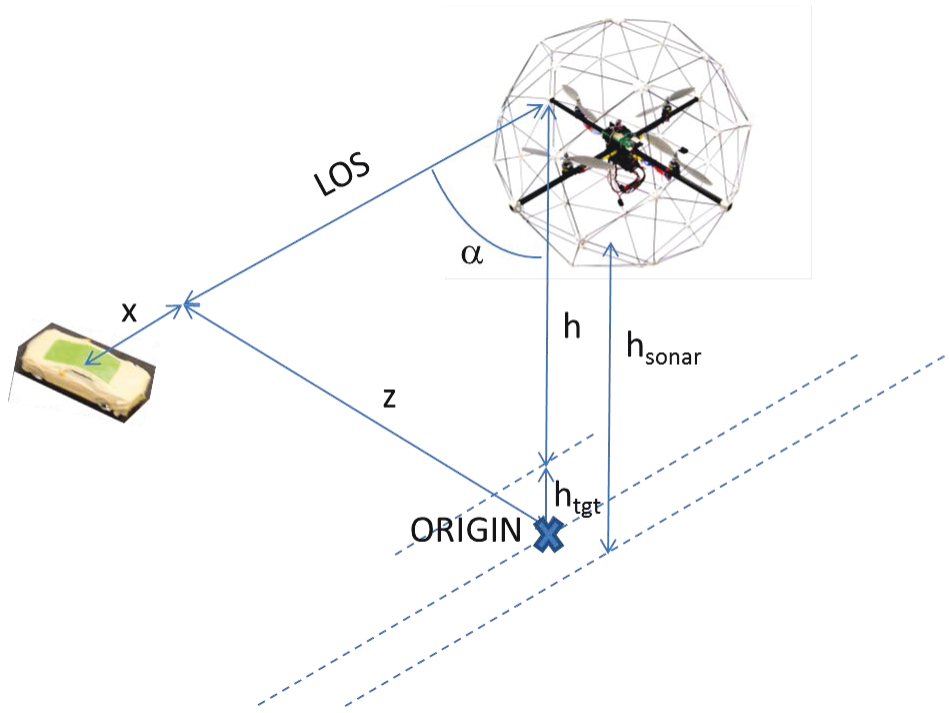

Figure 47. Relative position space of the Qball-X4 targeting solution.

Finally, the concept of desired position must be defined for this problem. The desired position refers to the target position that has a Z-angle (α) of 0.78 radians (45°) and an X-distance (x) of zero. The Z-distance (z) varies depending on the altitude of the UAV. The offset values, x_{offset} and z_{offset}, refer to the spacing between the desired position and the target position along the respective axes.

2. Calibration for Angle Based on Pixel Indices

This calibration was performed by placing the Qball-X4 at an altitude of 1 m and comparing measurements with their corresponding pixel indices. Each axis was calibrated separately, as outlined in this section.

a. Z-angle

To calculate the z-angle (α), which corresponds to the up/down or y-axis of the camera, 10 cm increments were measured along the horizontal center of the camera. First, the camera's 240x240 display was used to match the pixel index to each measurement.

Next, the following equation calculated the angle α of each increment, with zero representing a measurement directly below the camera.

$$\tan\alpha = \frac{z}{h} \tag{21}$$

Finally, the pixel index was divided by 100 to match the scale of the input and plotted against the angle, as shown in the plot in Figure 48. Using a linear curve-fitting model, the slope (Z_m) of the plot is -0.37 and the y-intercept (Z_b) is 1.28. When any y-pixel index is multiplied by Z_m and added to Z_b, the value derived is α. The center of the camera view was chosen as the desired z-offset, which corresponds to a Z-distance of 90 cm and a Z-angle (α) of 0.78 radians.

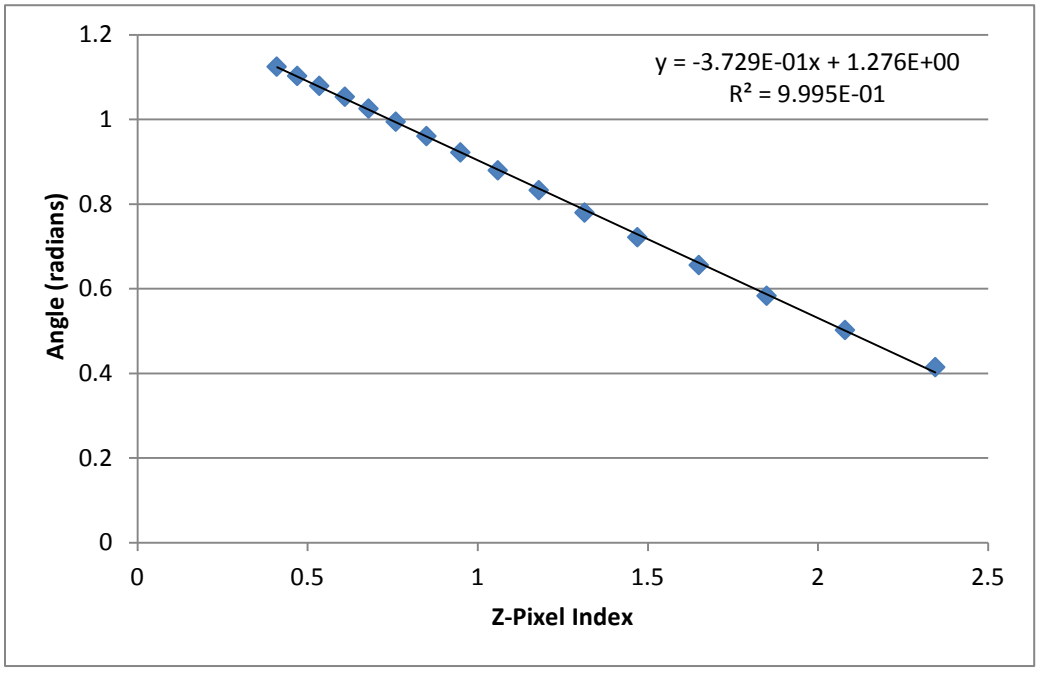

Figure 48. Z-angle calibration

b. X-offset When Z is Centered

This calibration required measurements along the x-axis of the camera view at its vertical center. Using the same method as the z-angle, pixel indices were recorded at 0.1m increments. Figure 49 contains a plot of the results, which derived values of X_m=0.55 and X_b=-0.66 for the model. These values are only accurate when z_{offset} is zero, meaning a different method was required to find X_m in the case that the target is not at the desired Z-angle.

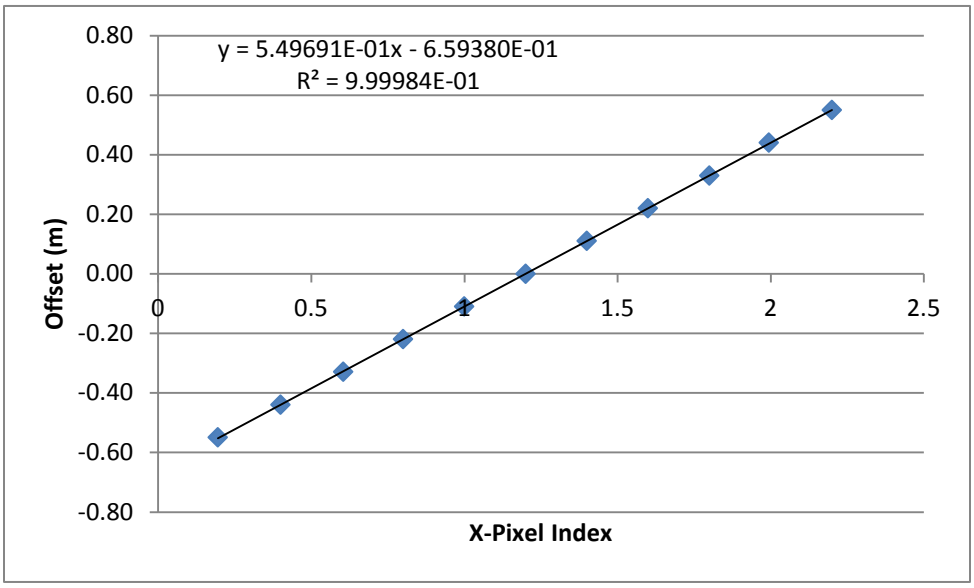

Figure 49. X-angle calibration at vertical center

c. X-offset When Z is not Centered

While the z-offset does not change for different values of x, the reverse does not hold. To quantify this, measurements were taken along a vertical line located 40 cm to the right of the horizontal center. The resulting pixel indices were used to calculate the "pixels per cm" for each value of z. The plot in Figure 50 shows that a logarithmic relationship exists between the Z measurement, in meters, and the "pixels per cm" value. This was used to calculate an accurate value of X_m in the model.

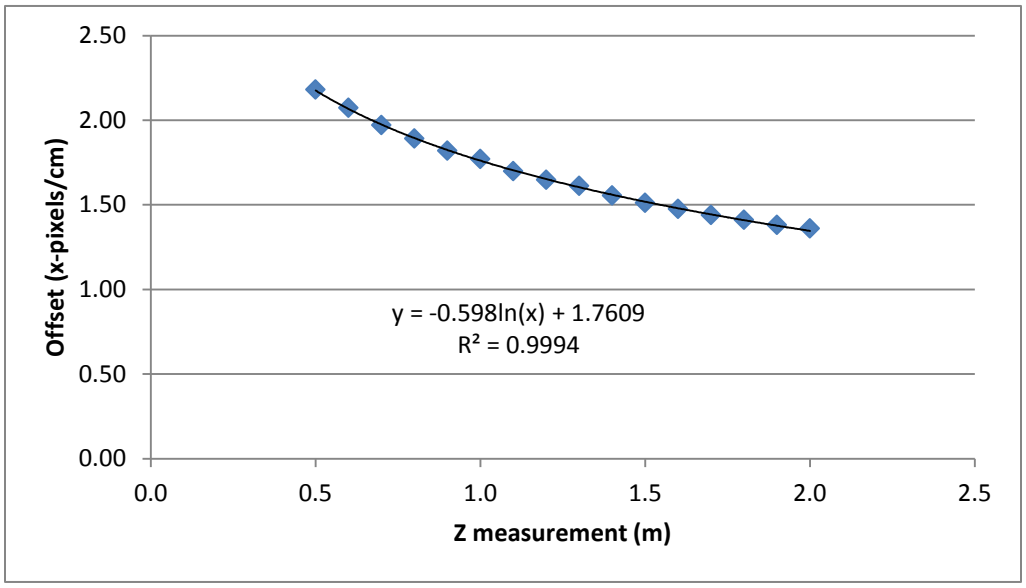

Figure 50. Graphical derivation of X_m for all Z values.

3. Using Targeting Height to Calculate Total Offset

Height above target is used to derive the x_{offset} and z_{offset} of the target. Table 3 lists the measurements used to calculate this height. The sonar, the only value on the table that varies with altitude, provides a measurement between the ground and the Qball-X4 base. The camera height represents the vertical component of the distance between the UAVbase and the center of the camera lens, and must be added to the sonar to provide the height of the camera above the ground. The height of the target must be subtracted from this sum to provide the "height above target," or h, for the targeting solution.

$$h = h_{sonar} + h_{camera} - h_{tgt} \tag{22}$$

Component	Value
Sonar height (h_{sonar})	Varies
Camera height (h_{camera})	0.2413 m
Target height (h_{tgt})	0.085 m

Table 3. Components used to calculate total camera height.

Once the height above target has been determined, it is used to calculate the offset values. For the x-offset, the height is simply multiplied by the offset derived above.

$$x_{offset} = hx \tag{23}$$

For the z-offset, the following equation is used, which is another form of (21).

$$z_{offset} = \mathrm{h} \tan \alpha \tag{24}$$

(24) is used twice, first to calculate the vertical position of the target, and then to calculate the *desired* position of the target based on the ideal angle of 0.78 radians. The difference between these two values is the z-offset of the target.

4. Pitch Adjustment

The offset calculations outlined in the previous section are sufficient when the Qball-X4 is parallel to the ground plane. When the UAV pitches forward or backward, however, the camera will move significantly, changing the perceived offset of the target. To account for pitch, three factors must be considered: pitch angle of the Qball-X4,

change in measured altitude due to a sonar that is tilted and not measuring directly below the UAV, and a change in camera height as a result of the pitch change.

The sensors of the Qball-X4 measure the pitch angle, θ, in radians, with a positive θ-value equating to a forward pitch. Because the model already measures the angle of the target with respect to the z-axis, the pitch correction must be made before the z-angle is converted to the z-offset. As the model shows, the pitch angle is subtracted from the z-angle measured from the camera before calculating the offset.

A sonar that is not pointing directly at the ground will always measure a higher altitude, since the ground appears to be farther away. To correct for this, the following adjustment is made.

$$h = h_{sonar}\cos(2\theta) \tag{25}$$

The small difference in height of the camera during pitch is still significant enough to require an adjustment in the calculations. The change in height of the camera can be calculated as follows, using 0.38m as the distance between the camera and the UAV's center of gravity.

$$\Delta h = 0.38\sin\theta \tag{26}$$

The change in height is subtracted from the altitude, since a positive θ would refer to a downward pitch and therefore reduce the height above target. This corrected altitude is then used for the x-offset and z-offset calculations..

5. Roll Adjustment

When the Qball-X4 rolls left or right, the target's location will move on the camera view. One major difference in the rolling motion is that when the target is in its desired position, its pixel indices will not change for any roll angle. When the target is away from its desired position, a roll will rotate the target around the desired position, meaning that it will always remain at a constant distance from the origin. Calculating the correct offset requires the implementation of a two-dimensional rotation matrix as follows.

$$\begin{bmatrix} x_{offset\_adj} \\ z_{offset\_adj} \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x_{offset} \\ z_{offset} \end{bmatrix} \tag{27}$$

where ϕ is the roll angle. These adjusted offsets are sent to the Position Commands subsystem to compute the roll and pitch commands.

C. SIMULATIONS

To verify the calibration, the Qball-X4 motors were first disabled in the HiQ subsystem. The UAV was then suspended approximately one meter above the ground, resting on a pair of metal rods placed between two shelves. This allowed the sonar to accurately measure altitude for the targeting solution calculations. Next, the target was placed at the desired position in front of the UAV, as confirmed by the camera view. Though the motors were inoperable, all sensors onboard the Qball-X4 were able to take measurements, testing the Targeting Solution model as well as the modifications that were made to the other models. Additionally, the "Black Box" recorded all data during these simulations. The photograph in Figure 51 shows this setup.

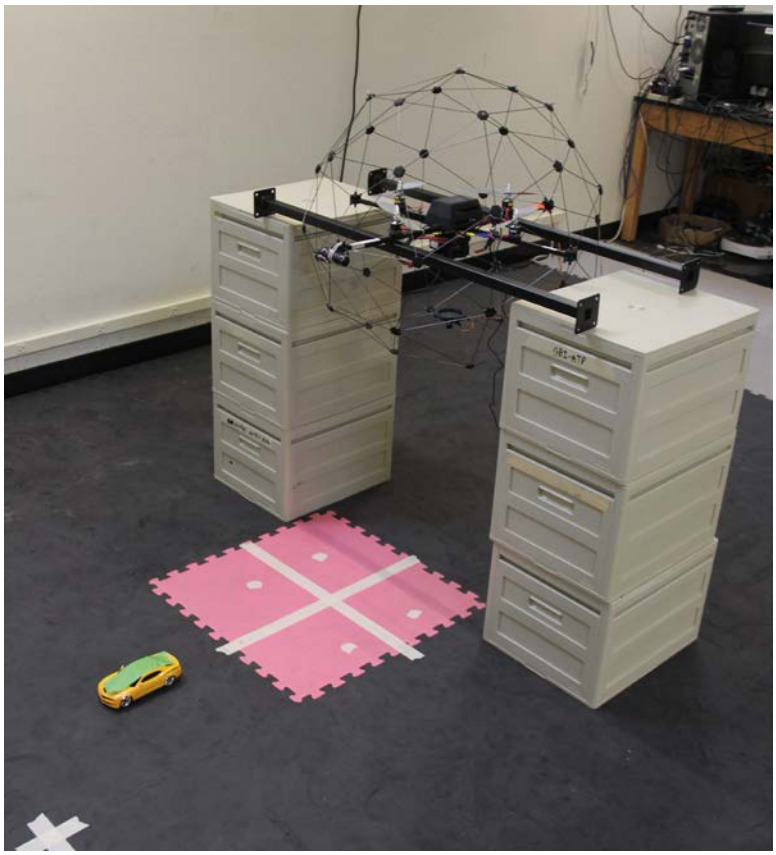

Figure 51. Setup of UAV for simulations.

To confirm the calibrations, four situations were simulated: target in desired position, target 30 cm in front of the desired position (z-axis correction), target 30 cm left of the desired position (x-axis correction), and 30 cm to the right of and behind the desired position (multi-axis correction). Additionally, the z-axis correction was tested with the UAV placed in a sustained forward pitch. For the first case, the noise of the X-offset and Z-offset was analyzed by comparing the variability with the OptiTrack position tracking. For the other cases, the following measurements were plotted for 15 seconds to confirm that the Qball-X4 was being controlled as expected:

- X-offset
- Z-offset
- Pitch Command
- Roll Command

1. Desired Position

The purpose of this test was to confirm two aspects of the calibration by placing the target in the center of the camera view. First, the Qball-X4 should not command any pitch or roll movements, as it was placed in its desired position. Second, any noise detected in the measurements should be comparable to the noise detected while on OptiTrack position mode. The plots in Figure 52 show the X-offset and Z-offset measured during this test.

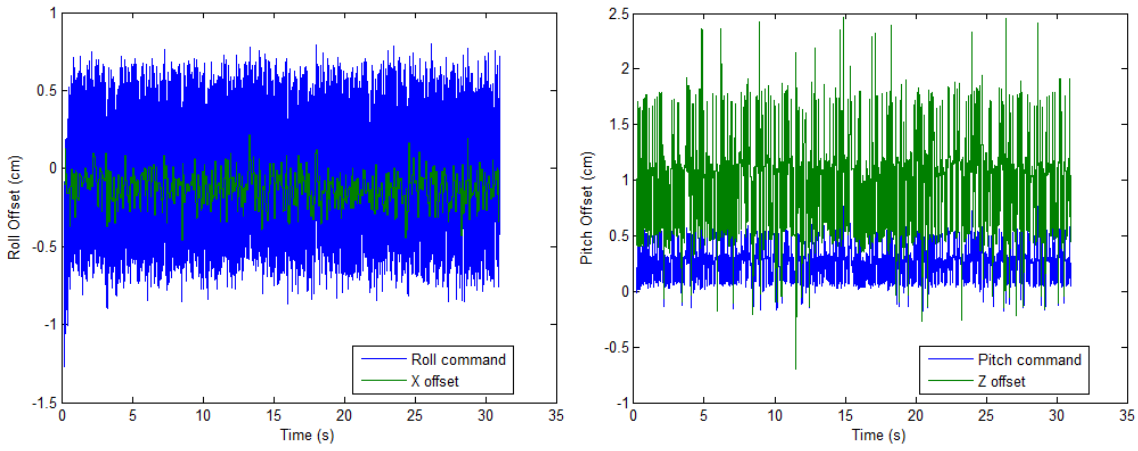

Figure 52. X-offset and Z-offset of the Qball-X4 during the desired position simulation.

As the data shows, the offsets were close to zero, indicating that the target was detected in the center of the camera view. Additionally, the roll and pitch commands were close to zero. This indicates that the Qball-X4 would hold its position in this case, which is the correct maneuver. Finally, the noise seen in the offsets was compared to the noise that would occur when position was controlled by the OptiTrack system. Figure 53 compares the above offset measurements to the measurements taken by the OptiTrack cameras. To better visualize this comparison, the means of each measurement were shifted to zero, aligning each plot along the x-axes.

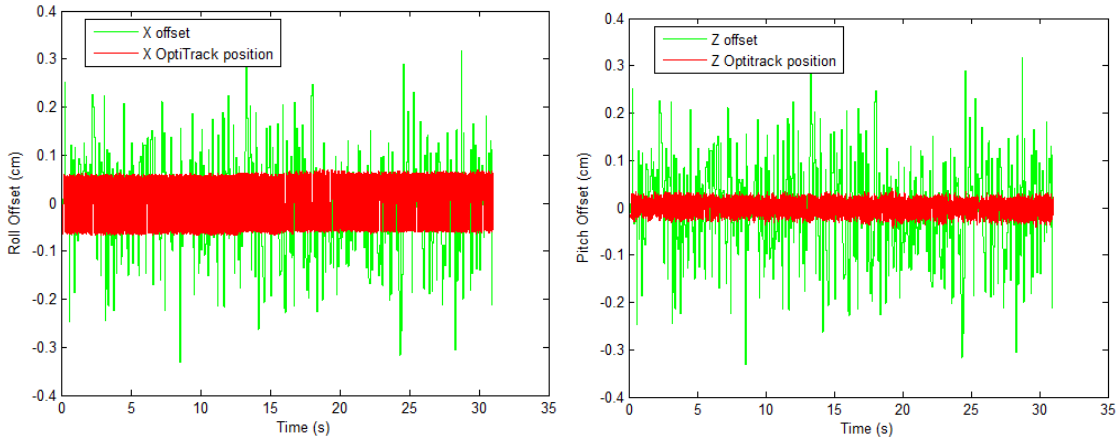

Figure 53. Comparison of offset positions and OptiTrack positions, normalized along the x-axis of the plot.

According the above plots, the Optitrack measurements were more precise than the camera measurements, The difference between the two, however, was very small, suggesting that the precision of the image recognition model is still very good. To quantify the noise, the standard deviation of all measurements was calculated using Matlab. The standard deviations of the OptiTrack measurements for the x- and z-axes were 0.59mm and 0.21mm, respectively. The standard deviations of the offsets for these axes, as derived through image processing, were 1.1mm and 4.1mm, respectively. While the offset measurements produced more noise, the level of noise was still extremely small, suggesting that the performance of the Qball-X4 would not deteriorate a great amount when the tracking algorithm is implemented.

2. Z-axis Correction

For this simulation, the target was placed 30cm forward of the desired position, or -30cm in the Z-direction. The plots in Figure 54 show the roll and pitch commands for this situation. As the left plot shows, the Qball-X4 did not command a roll in either direction, which makes sense considering that the target is centered along that axis. As the right plot shows, the UAV would command a pitch in the positive, or forward, direction, which would be the correct maneuver. Also, the mean of the z-offset was 29.1cm, which was very close to the actual offset of 30cm. This simulation proved that the tracking algorithm can correctly measure and correct for error in the z-direction.

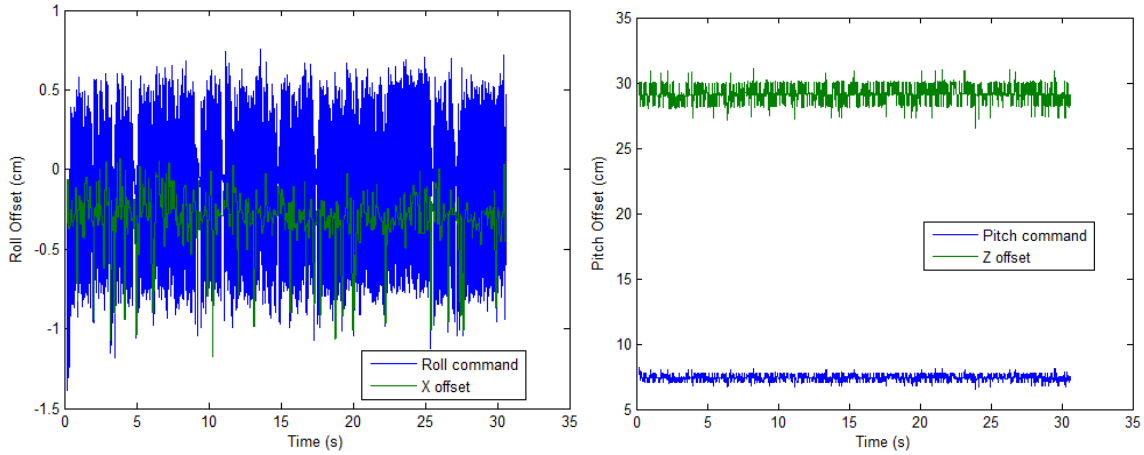

Figure 54. Roll and pitch commands for a target that is 30cm forward of the desired position.

3. Z-axis Correction during Forward Pitch

For this simulation, the target was placed in the same position as the previous simulation but the Q-ball-X4 was pitched forward about 6.5°. As the plot in Figure 55 shows, the measured z-offset was very similar to the offset measured in the previous example. To further quantify this measurement the mean of all offset measurements was measured using Matlab. In the previous simulation, the mean offset was 29.12cm; in this simulation, the mean offset was 29.96 cm. This proves that the pitch adjustment results in

an estimation that is very close to the actual z-offset measurement, suggesting that the effects to the UAV performance would be minimal when the pitch is not zero.

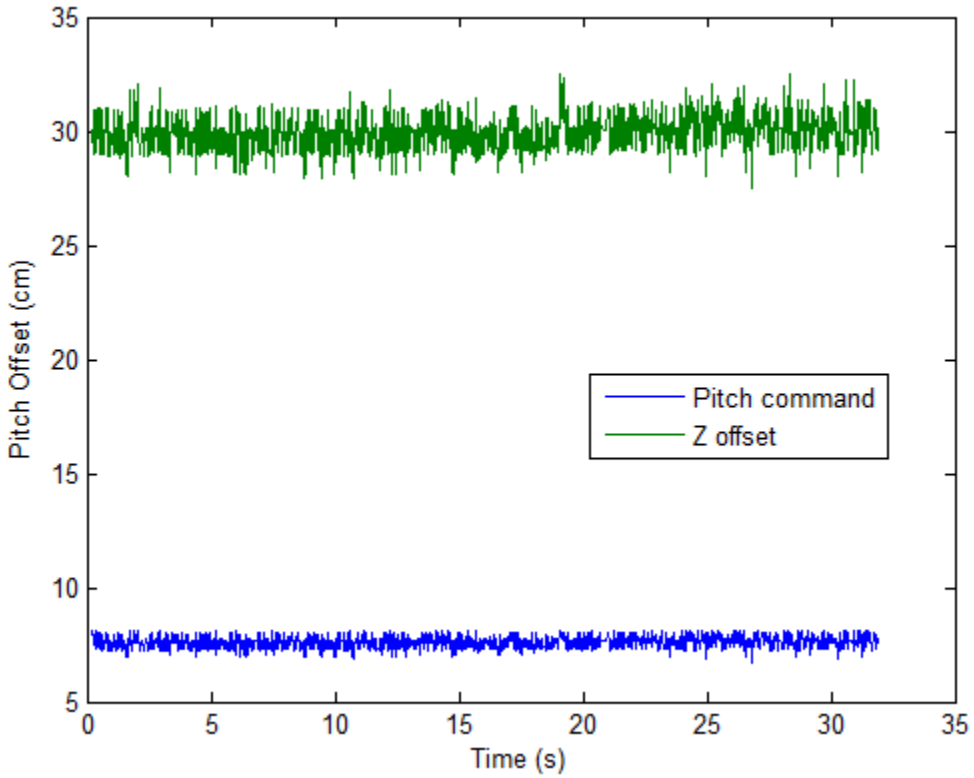

Figure 55. Pitch command for a target that is 30 cm forward of the desired position, with the Q-ball-X4 pitched forward 6.5°.

4. X-axis Correction

For this simulation, the target was placed 30 cm to the left of the desired position. As the plots in Figure 56 show, the mean of the target's position was calculated at -32.9 cm, close to the actual position. The roll command was the correct one, as the Qball-X4 was commanded to roll leftward to meet the target. The pitch measurement and pitch command were both close to zero, which was also expected.

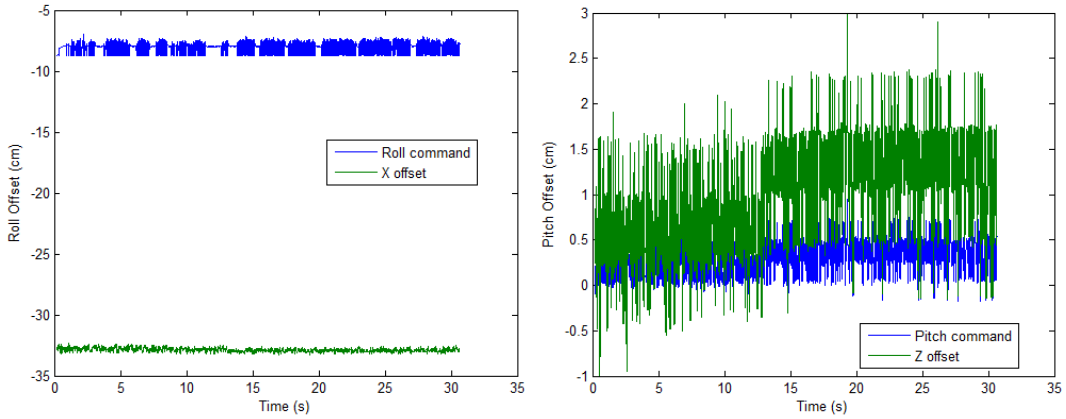

Figure 56. Roll and pitch commands for a target that is 30 cm left of the desired position.

5. Multi-axis Correction

For this simulation, the target was placed 30 cm right and 30 cm behind the desired position. As the plots in Figure 57 show, the measured X-offset was very similar to the previous simulation, as the mean was 32.9 cm. The Z-offset, however, showed a sizeable inaccuracy, as the offset was measured at -43.7 cm. In spite of this measurement issue, the performance of the Qball-X4 appears to be correct, since the commanded maneuvers are a roll to the right and a pitch upward, which would move the UAV closer to the desired position.

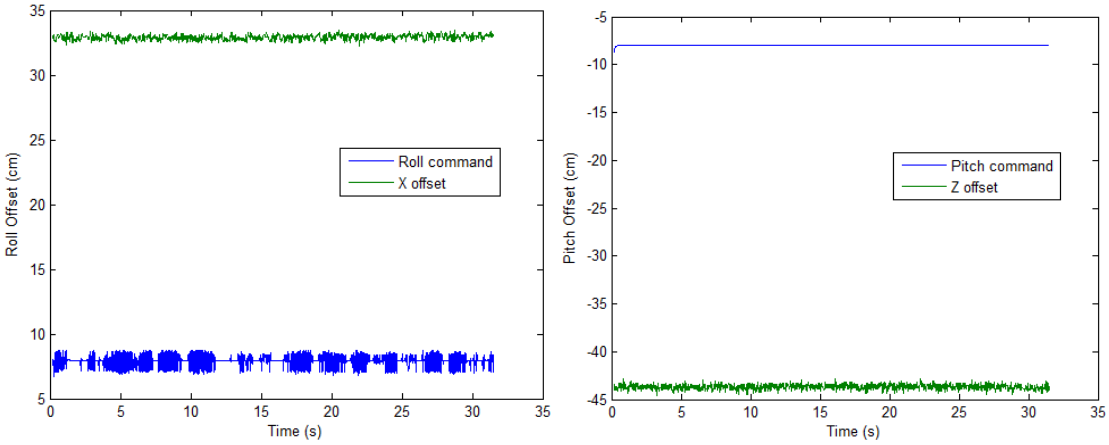

Figure 57. Roll and pitch commands for a target that is 30 cm right and 30 cm below the desired position.

THIS PAGE INTENTIONALLY LEFT BLANK

V. COOPERATIVE UGV/UAV EXPERIMENTS

All experiments were performed in a controlled laboratory environment. The experiments were designed to test the tracking ability of the Qball-X4 in a variety of scenarios. For each experiment, data was collected from the sensors and OptiTrack motion control system, allowing for thorough analysis of every aspect of each flight.

A. LABORATORY SETUP

The laboratory, as shown in the photograph in Figure 58, is contained in a 9 m by 8 m room, with 3 m in available altitude. The flight space is further reduced to 5.5 m by 4.3 m, and contains a foam floor to protect the Qball-X4 by softening its landings. The ground control station is located directly behind the flight space and contains the computer terminal, joystick, and USB connection for the onboard camera.

Figure 58. Laboratory used for all experiments. The ground control station is on the right side of the screen.

1. OptiTrack Motion Control System

The OptiTrack motion capture system consists of 10 cameras installed around the perimeter of the flight space. The system is designed to give the Qball-X4 multi-camera coverage throughout the flight space, allowing for an accurate three-dimensional representation of the UAV throughout each flight test. The system is connected to the ground control station through a set of USB cables and hubs, allowing for real-time data collection.

To calibrate the OptiTrack system, the manufacturer's Tracking Tools software was used. The first step in the calibration was to collect hundreds of measurements while waving a wand throughout the entire space. These measurements were performed until the calibration was rated "very high" by the software. Next, an T-shaped device was placed in the center of the space, determining the origin and the directions of the three axes. The illustration in Figure 59 shows two separate views of the entire OptiTrack system, with seven of the ten cameras measuring the Qball-X4 in the center of the flight space.

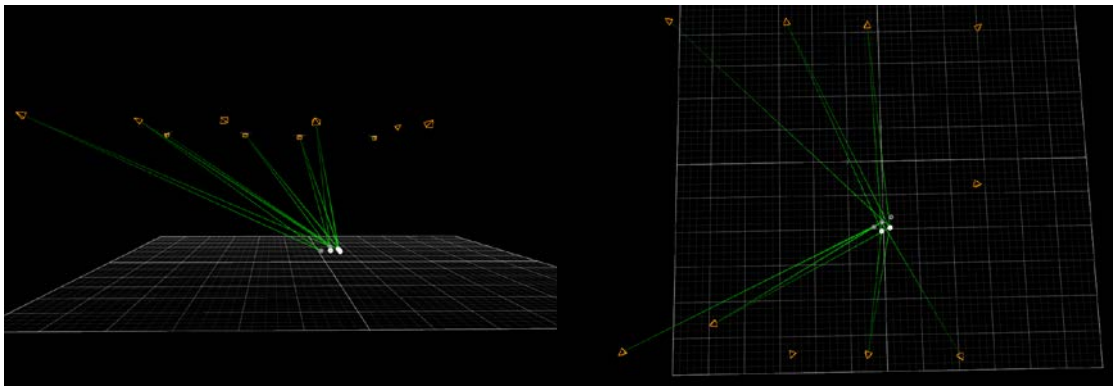

Figure 59. Side view and top view of the OptiTrack environment.

2. Safety Considerations

Though the laboratory is a controlled environment, there were still important safety considerations that must be made. First of all, the Qball-X4 and laboratory environment were thoroughly inspected before each flight, as is described in the next section. Also, the USB cable that connects the onboard camera to the ground control station was checked to ensure that it did not contact the propellers and that there was sufficient slack for the experiment. Finally, a warning sign was placed outside of the laboratory to alert people outside of the laboratory that a test is underway.

The experiments were conducted with two people in the laboratory. One person operated the throttle and focused solely on the safety of the UAV and the laboratory environment. The other person operated the control model by starting the model before the test and changing control modes as needed. Both operators participated in a full discussion of the experiment before setting up each flight.

3. Setup of Qball-X4 Flight

Before each Qball-X4 flight, a variety of steps were taken. First, the laboratory was scanned to ensure that no obstacles existed in or near the flight space. Also, the Qball-X4 was inspected to ensure that the propellers, components, and protective cage are all connected tightly. Following this inspection, the Qball-X4 was placed in the center of the flight space at the position matching the coordinates entered into the Mode Control subsystem. Next, a pair of charged batteries was attached to the center of the UAV with Velcro straps. After connecting the batteries to the Qball-X4, the system was then powered with two micro-switches. Finally, the target was placed about one meter in front of the UAV.

Once the hardware was set up, the software was then loaded on the ground control station. This was accomplished by opening the three Simulink models in succession. First, the Image Recognition model was loaded and run in normal mode. The camera view with subdued crosshairs appeared in a separate window, confirming that the model was operating correctly. Next, the Joystick Controller model was loaded and set for external mode. After starting the model, its operation was verified with three checks:

- Joystick Operation: scope was opened and checked to ensure joystick inputs
- OptiTrack Operation: it was confirmed that the cameras were each displaying a number and the "Trackables" block on the model showed "1"
- Camera Operation: the "Pixels" display was checked to see that it matched the location of the target on the camera view

Finally, the Qball-X4 control model was loaded. Before running this model, the wireless connection between the ground control station and the data acquisition card were made using the network settings. Also, the following settings require action prior to the flight:

- Mode Control subsystem: confirmed that altitude, position and heading modes were set to "1" and tracking mode was set to "0"
- HiQ subsystem: confirm that gain was set to [1 1 1 1]*1, enabling the motors
- Position Commands subsystem: confirmed that the heading command, altitude slider gain, and position slider gains were configured as required

Once a connection was confirmed, the code was compiled in the control model, loading the model onto the data acquisition card. Upon completion of this process, the model was connected and started, signaling that the Qball-X4 was ready for flight.

4. Flying the Qball-X4

The Qball-X4 flight, for these experiments, began in OptiTrack tracking mode. This allowed the model to automatically control the UAV to maintain an initial position, heading, and altitude on the OptiTrack system's coordinate grid. To launch the aircraft, the throttle input on the joystick was pushed upward to deactivate the safety switch, lifting the Qball-X4 off of the ground. The UAV then flew to the position commanded in the Position Commands subsystem, as the various controllers adjusted accordingly to maintain this position.

Once the Qball-X4 was stabilized in its position and the camera view was confirmed, tracking mode was activated by clicking the bottom switch in the Mode Control subsystem, beginning the experiment. Once the experiment was complete, the Qball-X4 was landed by placing the throttle control in the full down position. The UAV

continued to fly for a few seconds to stabilize its pitch and roll axes, then landed as its motors powered down.

B. EXPERIMENTS

To test the tracking ability of the Qball-X4, four scenarios were developed. For each scenario, the following characteristics are tested and analyzed:

- Accuracy and precision of offset measurement
- Stability of UAV while tracking
- Representation of target's movement while tracking
- Response of UAV to target movement

Additionally, any unusual behaviors, such as lost contact or false detections, were noted.

Each scenario began with the Qball-X4 climbing to its commanded altitude using the OptiTrack system, with the static target sitting near its desired position. Once the Qball-X4 was stable and at its correct altitude, tracking mode was selected. The target was then set into motion as directed by the individual experiment while the Qball-X4's behavior was monitored. Finally, once the maneuvers were completed and the ground below the UAV was checked for proper clearance, the Qball-X4 was landed by cutting the throttle on the joystick.

(1) Static Target Tracking

This experiment tested the ability of the Qball-X4 to maintain its desired offset behind the target, testing the ability of the controller to make proper and continuous error corrections. First, the UAV flew for 30 seconds at its position as determined by the OptiTrack cameras. Next, target mode was activated and the UAV captured the target's position, maintaining a constant offset from the target for 30 seconds. Finally, the target was landed using the joystick.

(2) Target Moving in Negative Z-direction

This experiment tested the ability of the Qball-X4 to track the target in the negative Z-direction, which would require a forward pitching motion. For this test, tracking mode was activated while the target was static, then the target advanced slowly for approximately three meters. The Qball-X4 was monitored to ensure continuous

tracking of the target. Following the test, the UAV was stabilized above the static target then landed using the same procedure as described earlier.

(3) Target Moving in Negative X-direction

This experiment tested the ability of the Qball-X4 to track the target in the negative X-direction, corresponding to a left rolling motion. During this test, the UAV was first stabilized above a static target, then the target moved leftward for approximately three meters. Similar to the previous tests, once the target motion was complete, the UAV stabilized over the static target prior to landing.

(4) Target Moving Along Both Axes

This experiment tested the ability of the Qball-X4 to track the target on both the X- and Z-axes. For this test, the Qball-X4 was first stabilized above a static target, which then advanced along a parabolic path as shown in Figure 60. Once the target finished advancing it was stopped, allowing the UAV to stabilize before landing.

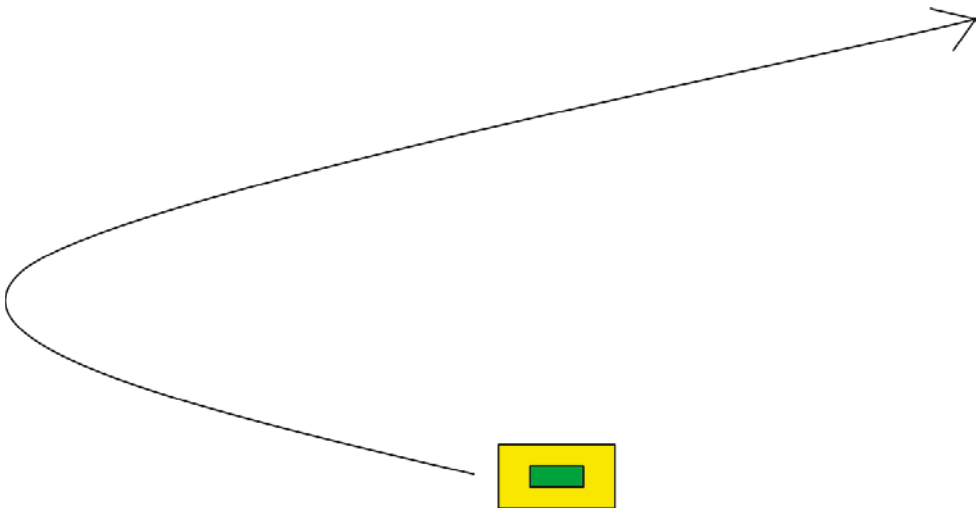

Figure 60. Path of the target during the fourth experiment.

VI. DISCUSSION OF EXPERIMENTAL RESULTS

A. STATIC TARGET TRACKING

The first experiment compared the characteristics of the Qball-X4 as its tracking mode was switched from OptiTrack to relative position tracking. The plots in Figure 61 show the X- and Z-positions of the Qball-X4 when relative tracking mode was off. As the plots show, the commanded positions were x=0m and z=3m. During the flight, the Qball-X4 maintained a position that was right of and in front of this commanded position.

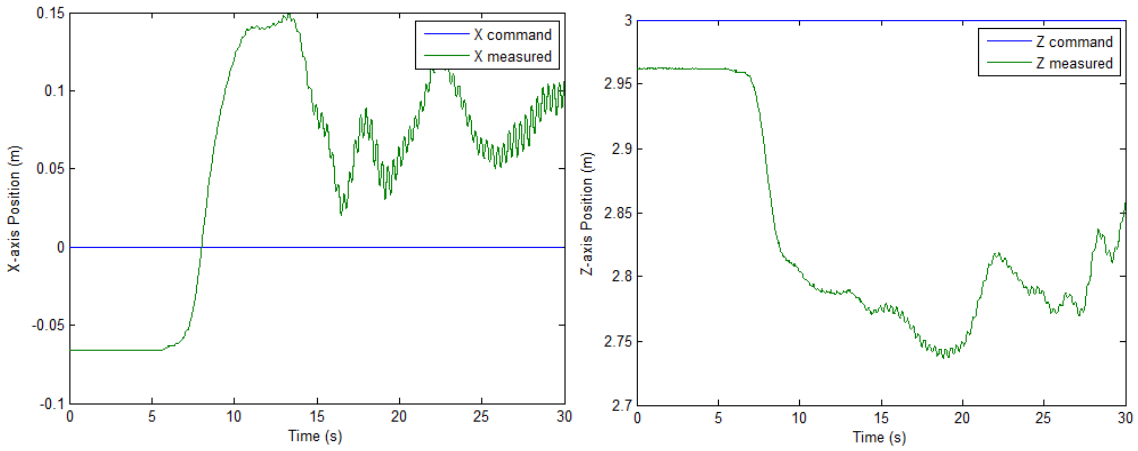

Figure 61. X and Z position of Qball-X4 when controlled by OptiTrack system.

The plots in Figure 62 show the X and Z positions of the Qball-X4 with relative tracking mode turned on. For this flight, tracking was turned on at the 15 second mark, which was easy to recognize as both position commands fluctuated around zero in response to the offset measurements. The commanded positions were mostly left and rearward, indicating that the Qball-X4 mostly maintained a position right of and in front of the desired position, matching the behavior in OptiTrack position mode. This suggests that a disturbance was present in the room which pushed the target in this direction.

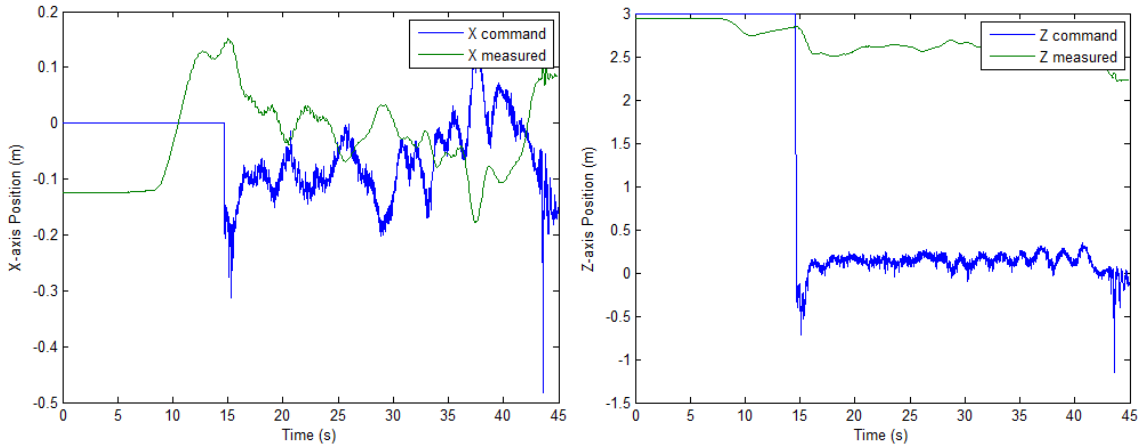

Figure 62. X and Z position of Qball-X4 when controlled by the Targeting Solution subsystem.

To gain more insight on the camera operation, the measured offsets are shown in Figure 63 for the period of time between 20 and 40 seconds. As these plots show, the X-offset fluctuated between 5 and 20 cm for most of the flight, then returned to zero as the UAV corrected itself along this axis. The Z-offset fluctuated between 0 and 20 cm, showing that the Qball-X4 maintained a position that was in front of the desired position.

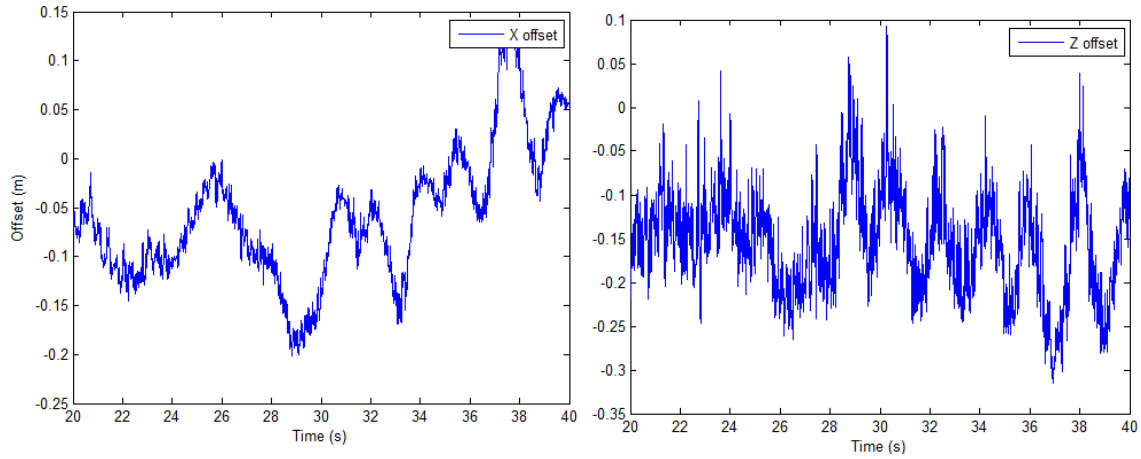

Figure 63. X and Z offsets, as detected by the camera attached to the Qball-X4.

Table 4 compares the performance of the targeting solution to that of the OptiTrack system. The mean position represents the UAV's total deviation from its

commanded position, and was calculated using the offset values for the targeting solution and using the OptiTrack position measurement for the OptiTrack tracking mode. The standard deviation represents the precision of the aircraft, and was calculated using the offset and OptiTrack position measurements. For each flight, a 15-second sample was used: the 20-35 second range for the targeting solution flight, and the 15-30 second range was used for the OptiTrack flight. As the data shows, the performance of the Qball-X4 was very similar for both modes. The camera actually placed the UAV closer to the target with regards to the z-axis, better overcoming the disturbance. The standard deviation was lower for OptiTrack mode for both axes, but only slightly lower.

Tracking System	Targeting Solution	OptiTrack
Mean Position (x)	+8.78 cm	+7.45 cm
Mean Position (z)	-14.43 cm	-21.74 cm
Standard Deviation (x)	3.03 cm	2.30 cm
Standard Deviation (z)	3.03 cm	2.77 cm

Table 4. Comparison of performance between Targeting Solution subsystem and OptiTrack position mode.

B. Z-DIRECTION MOVEMENT

In this experiment, after the Qball-X4 achieved a targeting solution on the target, the target was then moved in the negative Z-direction. The plot in Figure 64 shows the motion for 50 seconds of flight time. The Qball-X4 position was derived from OptiTrack data, while the target position was calculated using the Qball-X4 position, measured offset, and altitude. The target began moving forward (negative Z-direction) at the 25-second mark, and proceeded 2.5 meters in 15 seconds, coming to rest at the 40-second mark. As the plot shows, the Qball-X4 recognized this motion and moved in the same direction, successfully tracking the target.

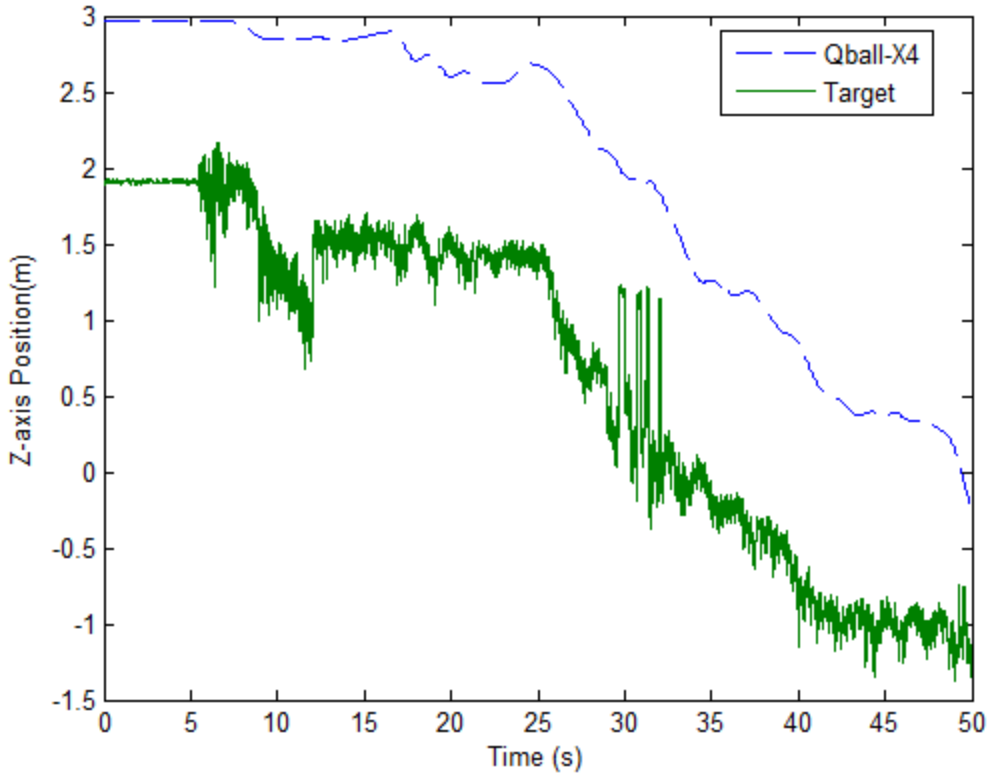

Figure 64. The progression of the target and Qball-X4 along the negative z-axis, showing a successful low-speed tracking solution.

The target's position contained a noticeable amount of noise. This was due to the noise in the camera's initial target recognition, which had small but significant error in its precision. The noise from this error propagates through the model, producing a greater amount of noise for the targeting solution. The plot in Figure 65 shows a two-dimensional representation of the Qball-X4 and target positions between the 20-second mark and the 50-second mark. As the plot shows, the estimation of the target's position contained noise in both directions, especially during the first stages of the target motion. The noise did not have a major effect on the path of the UAV, since the aircraft successfully followed the target as the car proceeded forward while maintaining stable flight. On the plot, the sign of the Z-axis is reversed to provide a "bird's eye" view of the target and tracking vehicle motion.

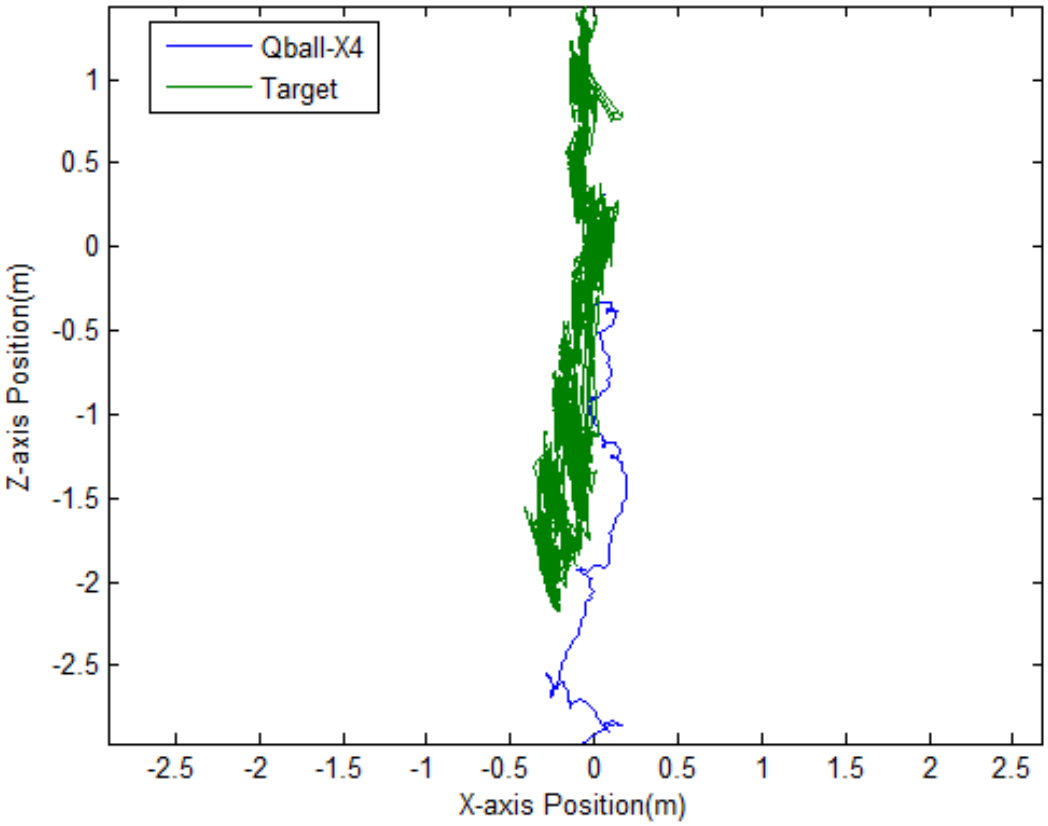

Figure 65. Two-dimensional position plot of the Qball-X4 and target position for the Z-direction movement.

To further analyze the tracking ability of the Qball-X4, the plot in Figure 66 shows the offset in the Z-direction between 20 and 50 seconds. As the data proves, the target was consistently 0-40 cm in front of the desired position during its forward motion. This was expected, as the Qball-X4 was programmed to react to the target's relative position but not to estimate its motion.

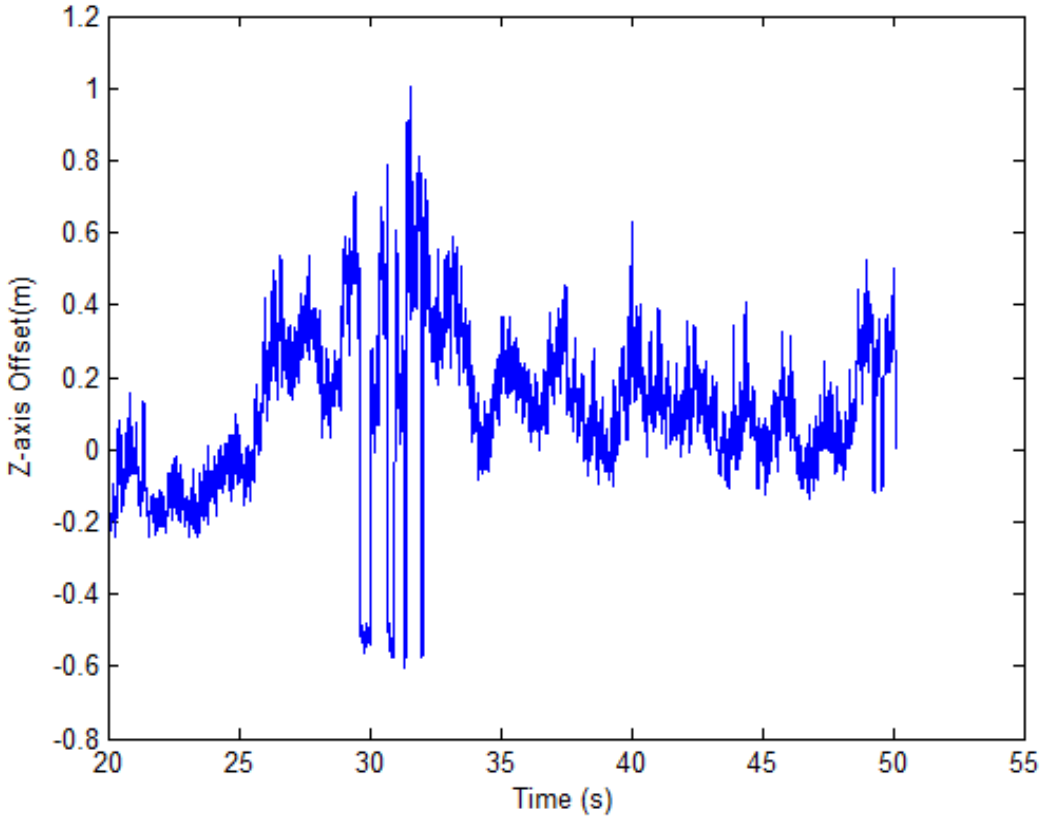

Figure 66. Z-axis offset of the target, as measured from the camera onboard the Qball-X4 during the second experiment.

According to the plot in Figure 66, the measured offset moved erratically between 29 and 32 seconds. This was a result of the onboard camera temporarily losing focus. In Figure 64, the target's estimated position was inaccurate during this span as well. Despite these erratic measurements, the aircraft did not maneuver erratically in response to the loss of focus. This relatively stable behavior by the Qball-X4 occurred because of the saturation and smoothing blocks in the pitch and roll controllers.

Finally, the performance of the Qball-X4 during the forward motion was measured by comparing the offset data during different three-second time periods: static target, beginning of target motion, further motion (after 10 seconds), and static target following motion. As the data in Table 5 shows, the Qball-X4 remained about 20 cm behind the desired position throughout the motion. The standard deviation was initially very high, likely due to the target's initial reaction to the motion, but was much lower

once the motion of the target was established. This suggests that the tracking algorithm can successfully track a target moving at a steady speed, but may have difficulty tracking a target that moves erratically.

Tracking Motion	Static (21-24 sec)	Initial Motion (25-28 sec)	Further Motion (35-38 sec)	Static (45-48 sec)
Mean Offset (z)	-13.50 cm	19.90 cm	19.08 cm	4.06 cm
Standard Deviation (z)	6.44 cm	16.36 cm	9.06 cm	8.57 cm

Table 5. Performance of Qball-X4 during negative Z-direction target movement.

C. X-DIRECTION MOVEMENT

For this experiment, the Qball-X4 tracked a target that moved in the negative X-direction, testing its ability to respond with rolling motion. The plot in Figure 67 shows the movement of both the target and the aircraft along the X-axis, calculated in a similar manner as the previous experiment. As the plot indicates, the target movement began at the 23-second mark. The target then moved to the left for 1.8 meters in the next 17 seconds, coming to a stop around the 40-second mark. The Qball-X4 appeared to be to the right of its desired position throughout the experiment. This behavior was constant for all experiments, suggesting that a disturbance was responsible for this trend.

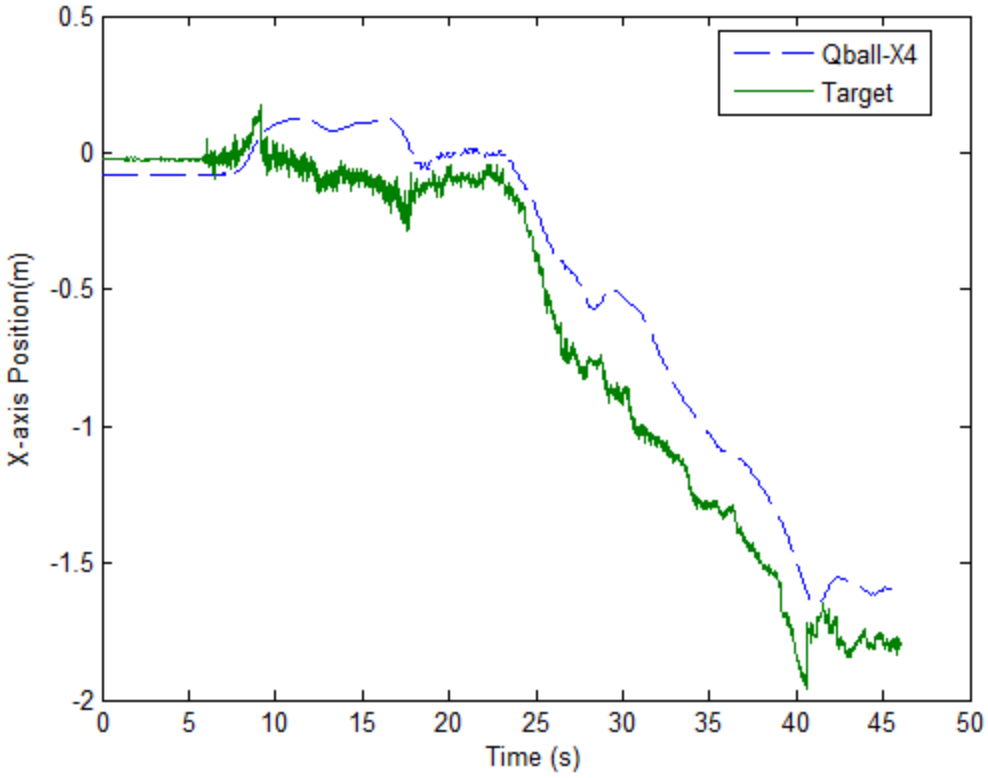

Figure 67. The progression of the target and Q-ball during movement in the negative X-direction, showing a successful tracking solution.

The plot in Figure 68 is a two-dimensional representation that shows the movement of the aircraft and the target between the 20- and 40-second marks. Similar to the previous experiment, the estimation of the target's position contained error along both axes. The aircraft did not behave erratically in response to this error, instead moving smoothly to the left.

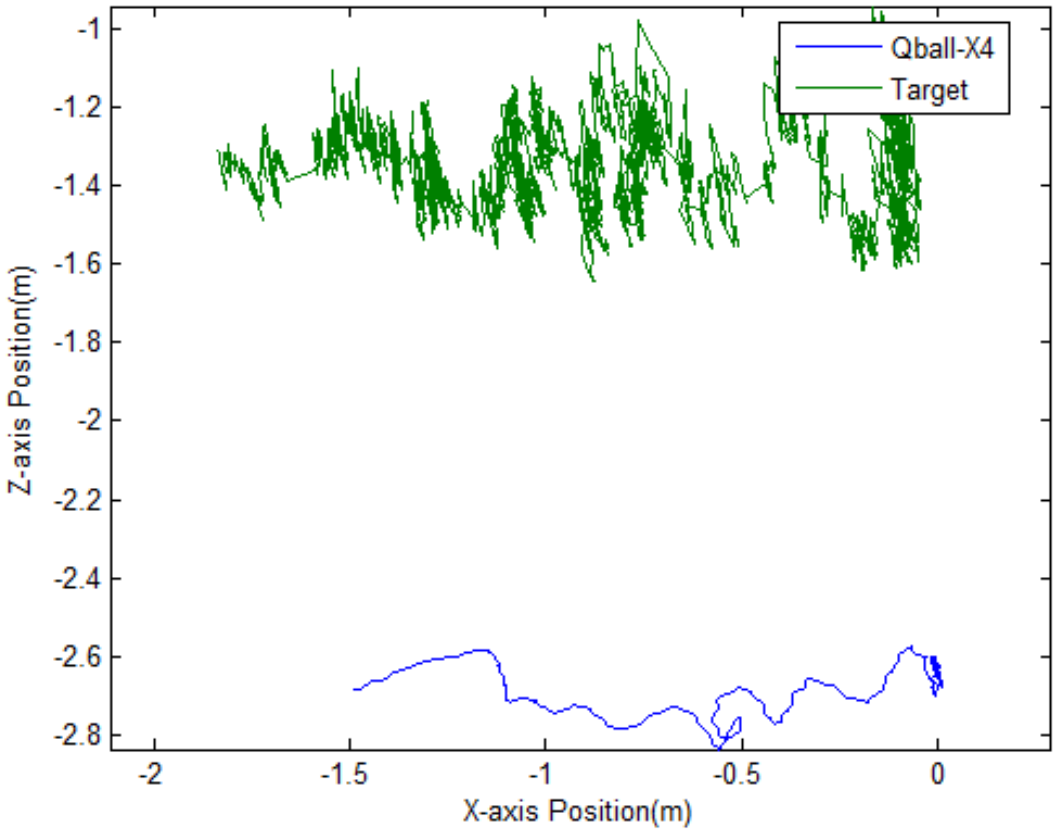

Figure 68. Two-dimensional position of the target and Qball-X4 during the third experiment. The sign of the Z-axis is reversed to create a bird's eye view of this representation.

Figure 69 contains a plot of the offset in the X-direction against time, allowing for further analysis of the tracking ability in this direction. The data proves that the target was consistently left of the camera's crosshairs. The aircraft was 10 cm to the right of the target in the beginning, then 40 cm to the right after the motion begin. The Qball-X4 was able to make the correct adjustments, ending the tracking portion of the flight 25 cm to the right of the desired position.

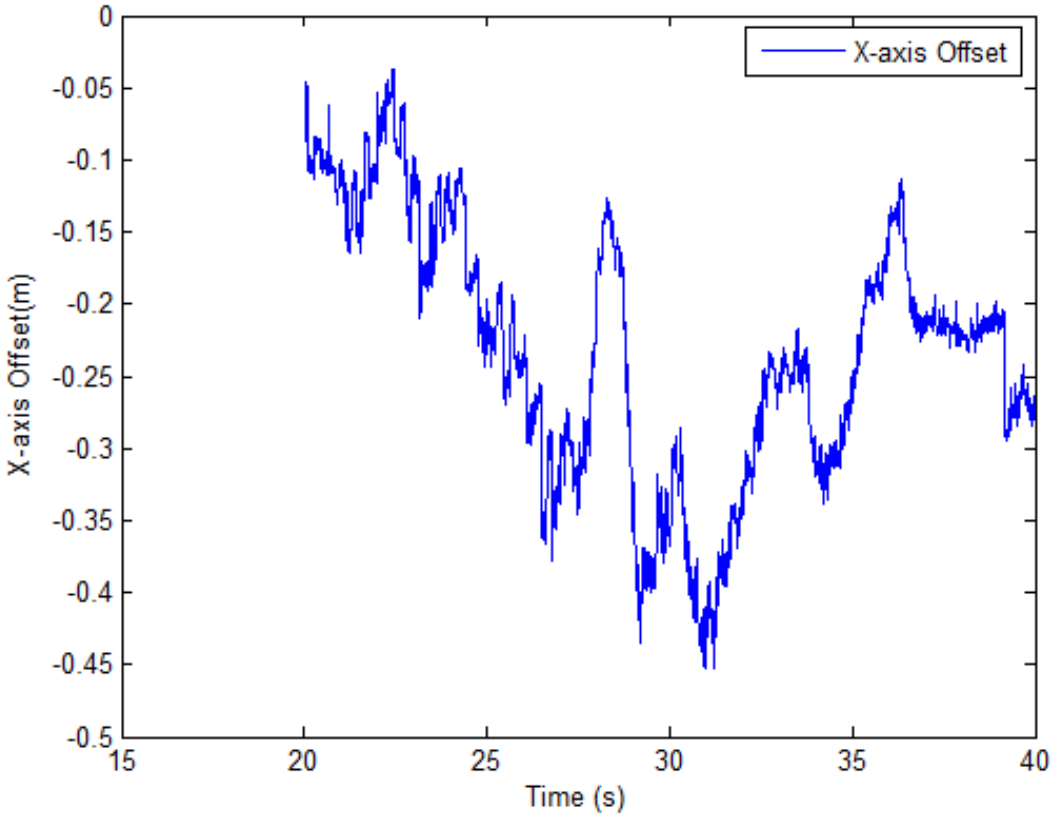

Figure 69. X-axis offset of the target during the target's leftward movement. The offset varied between 10 cm and 40 cm, but remained left of the desired position throughout the flight.

To analyze the performance of the Qball-X4 during lateral motion, the offset of the Qball-X4 was measured in a manner similar to the previous experiment. The same four situations were used, though the time intervals were adjusted to match the data for this experiment. Table 6 contains the mean and standard deviation for the offset during different phases of the flight. As the data shows, the mean offset was about 11 cm left when the target was static, and moved further to the left while the target was in motion. This is similar as the above experiment, as the aircraft always lagged the moving target. The standard deviation of the offset, however, was much lower than in the previous experiment. Surprisingly, even upon initial motion the deviation along the x-axis was less than it was along the z-axis during the previous test.

Tracking Motion	Static (19-22 sec)	Initial Motion (25-28 sec)	Further Motion (35-38 sec)	Static (43-46 sec)
Mean Offset (x)	-11.48 cm	-27.34 cm	-19.68 cm	-11.38 cm
Standard Deviation (x)	2.51 cm	4.48 cm	3.13 cm	3.28 cm

Table 6. Performance of the Qball-X4 during target movement along the negative x-axis. The mean and standard deviation were calculated during four distinct time intervals.

D. MULTI-AXIS MOVEMENT

This experiment tested the performance of the Qball-X4 when tracking the target along multiple axes. The setup of this experiment was similar to the previous test with the target beginning its motion in the negative X-direction. Instead of proceeding in a straight line, the target started a slow right turn, following a parabolic pattern. The plot in Figure 70 compares the Qball-X4 position and the target position along each axis, similar to the previous experiments. Along the X-axis the target leads the UAV when moving in either direction, while the Qball-X4 appears to hold a consistent offset in the Z-direction. Between 40 and 45 seconds there are two sudden peaks. This was a result of the camera temporarily losing focus, and did not lead to any erratic behavior from the UAV.

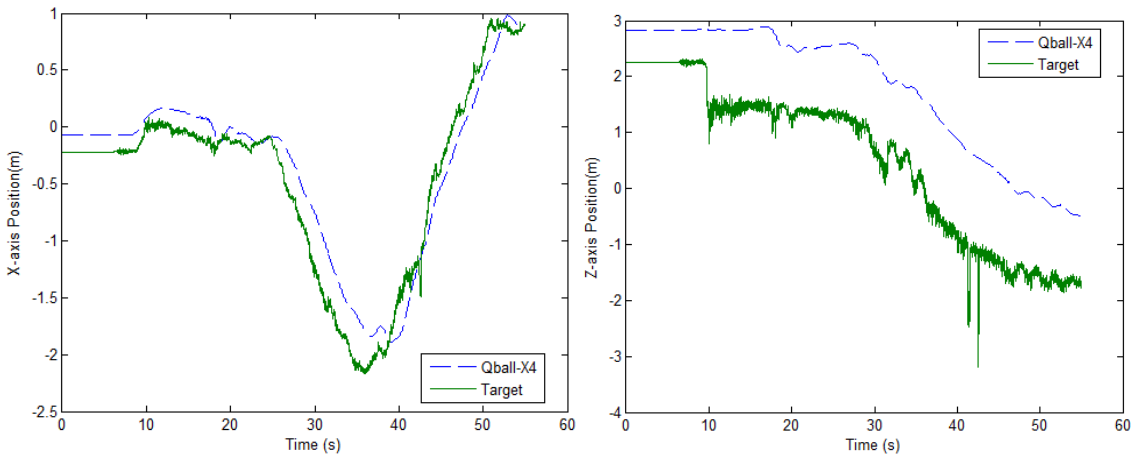

Figure 70. The progression of the target and Qball-X4 in the X and Z directions during the multi-axis test.

Figure 71 shows a multi-axis representation of this experiment, with the sign of the Z-axis position reversed to represent a birds-eye view. The loss of focus that occurred around the 40-second mark is easy to see, but the rest of the target's path appears to be relatively precise.

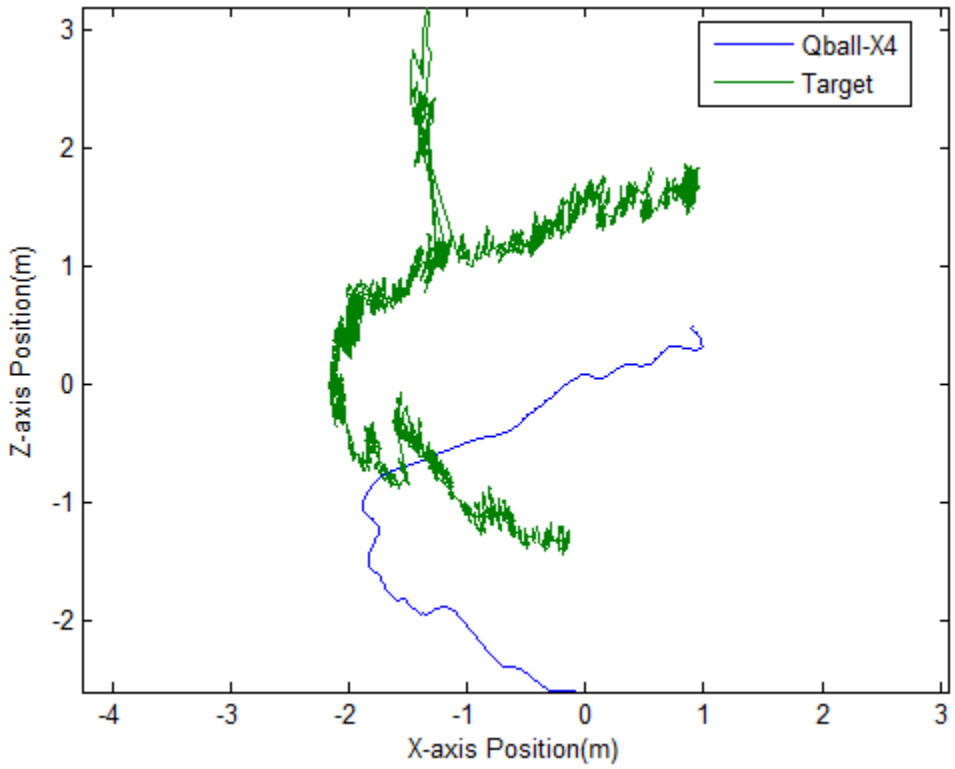

Figure 71. Progression of target and Qball-X4 during the multi-axis experiment.

VII. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

The following conclusions were reached through this research:

- A successful model for a targeting solution on a quadrotor UAV was built using the aircraft's sensors and an onboard camera. This targeting solution resolved a "desired position" of the target, giving the aircraft a goal position to maintain throughout the tracking task.
- Using image recognition, the target's location was represented in the model first as a pair of pixel indices, then as a distance from the desired position. The model accounted for altitude, pitch angle, and roll angle of the UAV in its determination of this position. The target's position was then compared to the desired position to determine the offset, which represented position error along two axes.
- Measurements from the model were incorporated directly into the UAV's pitch and roll controllers. The controllers, which were programmed to calculate error as the difference between actual and commanded position along two axes, successfully replaced this error with the offset error.
- Multiple tests were run, testing the targeting solution against a static target, a target moving in vertical and horizontal directions, and a target moving along a parabolic path. In each case, the targeting solution allowed the aircraft to successfully track the target, validating the use of relative position as a primary source of navigation.
- The experimental data was analyzed to evaluate the performance of the data. In both cases, the Qball-X4 lagged about 20-30 cm behind the target in terms of its desired position, which was expected for a reactive tracking algorithm. The precision along the lateral axis proved to be superior to that along the vertical axis, as the standard deviation of the offset was three times smaller for the lateral axis measurements.

B. RECOMMENDATIONS

Suggestions for future research include the following:

- Incorporation of the target's orientation in the tracking solution. For this research, the aircraft was programmed to hold a constant heading while tracking the target. If the onboard camera could calculate the orientation of the target, it could match the tracking vehicle's heading to the heading of the target, allowing the aircraft to track the target along its primary flight axis. A wireless camera would probably be needed if the Qball-X4 was

permitted to change heading, since the current configuration could lead to a tangled wire that could crash the UAV.

- Estimation of the "range to target" to back up or replace the sonar's contribution to the algorithm. Range can be estimated by measuring the size of the target on the scope, as a larger target indicates a smaller range to target. Range would definitely add value to the algorithm providing a layer of redundancy as well as allowing the UAV to track a ground vehicle on uneven terrain.
- Addition of cameras along secondary axes. With one camera, the tracking vehicle can easily lose a target that quickly changes direction, due to the tracking aircraft's limited field of view. By adding additional cameras, the UAV can increase its field of view and successfully maintain visual contact on the target. It would also minimize any problems that could occur if one camera temporarily loses focus.
- Incorporation of relative motion estimation. Though the tracking platform successfully follows the target, it tends to lag behind the target's movement. By estimating the relative motion of the target, the tracking vehicle can anticipate changes in the target's motion, allowing for a more precise tracking solution. This would be ideal for tracking a non-evasive target.
- Use of relative position tracking for formation flight. Using this model of relative position tracking, a "wingman" aircraft can maintain a desired position behind a lead aircraft with little deviation. This could allow for three, four, five, or more aircraft to travel together with a similar magnitude of precision. Numerous researchers are testing the concept of a "swarm" of unmanned aircraft flying in unison; relative position tracking could allow for a very large number of aircraft to fly together safely.

APPENDIX

Matlab Scripts

This script contains the controllers that were loaded into the Qball-X4 before each test. Most of the script was developed by Quanser. Modifications include the addition of the first five variables and the tuning of the LQR gains. These controllers are explained in more detail in Chapter II.

```
% This file contains all the controller parameters and LQR gains for
the Qball.

% Calibration values for camera
% X returns offset position, Y returns offset angle, both from pixels
Xm = -0.59;    % x-slope
Xb = 1.76;     % x-intercept
Ym = -0.3729;  % y-slope
Yb = 1.2764;     % y-intercept
Offset = 0.827;  % Goal offset, in radians

% PITCH and ROLL
wnom = 15;
L = 0.2;
w = wnom;
K = 120;
J = 0.03;
Jyaw = 0.04;
CLimit = 0.025;
M = 1.45;  % Camera weight 0.05 kg added
g = 9.8;

Am = [0 1 0
    0 0 2*K*L/J
    0 0 -w];
Bm = [0 0 w]';
Aobs = Am' ;
Bobs = eye(3);
Qobs = diag([.001 10000 .01]);

Robs = diag([ 1 1 1 ])*1;
Kobs = lqr(Aobs,Bobs,Qobs,Robs)
Kobs = Kobs';
Aobs = Aobs'-Kobs*Bobs';
eig(Aobs)
Bobs = [Bm Kobs]
Cobs = eye(3)
Dobs = [ 0 0 0 0
    0 0 0 0
```

```
    0 0 0 0];

% augment with integrator
Ai = [Am [0 0 0 ]'
      1 0 0 0 ];
Bi = [Bm' 0]';
Ci = eye(4);
Di = [0 0 0 0 ]';
Q = diag([70 0 22000 10]); %old [100 0 22000 10]
R = 30000;

ki = lqr(Ai,Bi,Q,R);
rp_eig = eig(Ai-Bi*ki);
fprintf ('********************************************** \n');
fprintf('ROLL, PITCH DESIGN \n');
fprintf( 'P = %5.3f D = %5.3f Actuator = %5.3f I = %5.3f \n\n',ki(1),
ki(2),ki(3),ki(4));
for i = 1:4
fprintf('  %5.3f + %5.3f i \n  ',real(rp_eig(i)), imag(rp_eig(i)));
end;


%POSITION CONTROLLER (C2)
% XZ travel

tlimit = 5*pi/180; %max pitch cmd radians
%tlimit = 15*pi/180; %max pitch cmd radians
vlimit = 0.3; % max speed cmd in m/sec
%vlimit = 0.5; % max speed cmd in m/sec
Tau_theta = 1/7; % closed loop time constant for pitch response
wt =1/Tau_theta; %closed loop theta bandwidth
kt = 1;
a = [0 1 0 0
    0 0 g 0
    0 0 -wt 0
    1 0 0 0 ];
b = [0 0 wt 0 ]';

q = diag([ 5 2 0 0.1]);
%q = diag([ 5 2 0 0.1]);
%r = 50;
r = 50;

k = lqr(a,b,q,r);

ac = a-b*k;
xy_eig = eig(a-b*k);
Kp = k(1);
Kd = k(2);
Ki = k(4);
Kw = k(3);
fprintf('\n\n X Y Design \n');
fprintf( 'P = %5.3f D = %5.3f Actuator = %5.3f I = %5.3f \n\n',k(1),
k(2),k(3),k(4));
```

```
for i = 1:4
fprintf('  %5.3f + %5.3f i \n  ',real(xy_eig(i)), imag(xy_eig(i)));
end;

% Z axis without actuator

vlimith = 0.1;
Amh = [0 1
    0 0 ]
Bmh = [0 4*K/M]';
Cmh = [1 0];
Dmh = 0;

% augment with integrator
Aih = [Amh [0  0 ]'
      1  0 0 ];
Bih = [Bmh' 0]';

Cih = eye(3);
Dih = [0 0 0]';

Q = diag([1 0 50]);
R = 5000000;
kh = lqr(Aih,Bih,Q,R);
h_eig = eig(Aih-Bih*kh);
fprintf ('********************************************** \n');
fprintf('Z DESIGN \n');
fprintf( 'P = %5.3f D = %5.3f I = %5.3f \n\n',kh(1), kh(2),kh(3));
for i = 1:3
fprintf('  %5.3f + %5.3f i \n  ',real(h_eig(i)), imag(h_eig(i)));
end;
Kph = kh(1);
Kdh = kh(2);
Kwh = 0;
Kih = kh(3);

% yaw axis

Ky = 4;
Jy = 0.032;

Amy = [0 1
    0 0 ];
Bmy = [0 4*Ky/Jy]';
Cmy = eye(2);
Dmy = [0;0];

Qy = diag([1 0.1]);
Ry = 1000;
ky = lqr(Amy,Bmy,Qy,Ry);
h_eigy = eig(Amy-Bmy*ky);
Kpyaw = ky(1);
Kdyaw = ky(2);
```

```
Bih = [Bih,[0 1 0]'];

Dih = [Dih, [0 0 0]'];
```

This script contains the specific filter transfer functions used in the pitch and roll controllers. It was developed by Quanser.

```
t=10;
s = tf('s');
Gg = t^2*s/(t*s+1)^2
Gi = (2*t*s+1)/(t*s+1)^2
```

This script created the plots used in the simulations (Chapter IV), and was developed specifically for this research.

```
clc
close all
t = qball_data(1,40:end);
x_off = qball_data(48,40:end)*100;
z_off = qball_data(49,40:end)*100;
roll_cmd = qball_data(20,40:end)*100;
pitch_cmd = qball_data(21,40:end)*100;

figure(1);
plot(t, roll_cmd,t, x_off);
legend('Roll command','X offset','location','Best');
xlabel('Time (s)')
ylabel('Roll Offset (cm)')

figure(2);
plot(t, pitch_cmd, t, z_off);
legend('Pitch command','Z offset','location','Best');
xlabel('Time (s)')
ylabel('Pitch Offset (cm)')
```

This script created the plots used in experiment 1 (Chapter VI), and was developed specifically for this research.

```
clc
close all
t = qball_data(1,:);
```

```
x_cmd = qball_data(27,:);
y_cmd = qball_data(29,:);
z_cmd = qball_data(28,:);
x = qball_data(30,:);
y = qball_data(31,:);
z = qball_data(32,:);

t = t(:,11:9000);
x = x(:,11:9000);
z = z(:,11:9000);
x_cmd = x_cmd(:,11:9000);
z_cmd = z_cmd(:,11:9000);


figure(1);
plot(t, x_cmd, t, x);
legend('X command', 'X measured');
xlabel('Time (s)');
ylabel('X-axis Position (m)');

%figure(2);
%plot(t, y_cmd, t, y);
%legend('Y command', 'Y measured');

figure(3);
plot(t, z_cmd, t, z);
legend('Z command', 'Z measured');
xlabel('Time (s)');
ylabel('Z-axis Position (m)');
```

This script created the plots used in experiments 2, 3, and 4 (Chapter VI), and was developed specifically for this research.

```
clc
close all

t_init = 0.1;  % Start time (sec) of data collect
t_final = 55;   % End time (sec) of data collect
t_start = t_init * 200; % Derives start of each data array
t_end = t_final * 200;  % Derives end of each data array

t = qball_data(1,t_start:t_end);
x_off = qball_data(48,t_start:t_end);
z_off = qball_data(49,t_start:t_end);
roll_cmd = qball_data(20,t_start:t_end);
pitch_cmd = qball_data(21,t_start:t_end);
x_OT = qball_data(30,t_start:t_end);
z_OT = qball_data(31,t_start:t_end);
x = qball_data(30,t_start:t_end);
y = qball_data(31,t_start:t_end);
```

```
z = qball_data(32,t_start:t_end);
heading = qball_data(25,t_start:t_end);

tz = z - 0.9 * y - z_off;
tx = x + x_off - 0.9 * y .* sin(heading);

figure(1);
plot(t, x_off);
legend('X-axis Offset','location','Best');
xlabel('Time (s)')
ylabel('X-axis Offset(m)')

figure(2);
plot(t, z_off);
legend('Z-axis Offset','location','Best');
xlabel('Time (s)')
ylabel('Z-axis Offset(m)')

figure(3);
plot(t, z, '--', t, tz);
legend('Qball-X4', 'Target','location','Best');
xlabel('Time (s)')
ylabel('Z-axis Position(m)')

figure(4);
plot(t, x, '--', t, tx);
legend('Qball-X4', 'Target','location','Best');
xlabel('Time (s)')
ylabel('X-axis Position(m)')

figure(5);
plot(x, -z, tx, -tz);
legend('Qball-X4', 'Target','location','Best');
xlabel('X-axis Position(m)')
ylabel('Z-axis Position(m)')
axis equal
```

LIST OF REFERENCES

[1] National Research Council of the National Academies, *Autonomous Vehicles in Support of Naval Operations*, The National Academies Press, Washington, DC, 2005, p. 13.

[2] Lin, F., Dong, X., Chen, B.M., Lum, K.Y., and Lee, T.H., "A Robust Real-Time Embedded Vision System on an Unmanned Rotorcraft for Ground Target Following," *IEEE Transactions on Industrial Electronics*, Vol. 59, No. 2, 2012.

[3] Autonomous Systems Technologies Research and Integration Laboratory, "Vision based GPS-denied Object Tracking and Following for Unmanned Aerial Vehicles," URL: http://robotics.asu.edu/ardrone2_ibvs/ [cited 2 March 2015].

[4] Rafi, F., Khan, S., Shafiq, K., and Shah, M., "Autonomous Target Following by Unmanned Aerial Vehicles," University of Central Florida, Orlando, FL, URL: http://vision.eecs.ucf.edu/papers/Fahd_SPIE_06.pdf [cited 2 March 2015].

[5] Hewgley, C.W., "Pose and Wind Estimation for Autonomous Parafoils," Ph.D. Dissertation, Electrical Engineering Dept., Naval Postgraduate School, Monterey, CA, 2014.

[6] Martin, M.W., "Global Versus Reactive Navigation for Joint UAV-UGV Missions in a Cluttered Environment," M.S., Thesis, Mechanical Engineering Dept., Naval Postgraduate School, Monterey, CA, 2012.

[7] Jones, L., "Coordination and Control for Multi-Quadrotor UAV Missions," M.S., Thesis, Mechanical Engineering Dept., Naval Postgraduate School, Monterey, CA, 2012.

[8] Newton, I., "Absolute and Relative Space, Time, and Motion," originally published in *Philosophiae Naturalis Principia Mathematica,* Bk. 1, 1689, translated by Andrew Motte, University of California Press, Berkeley, CA, 1934.

[9] Jain, R., Kasturi, R., and Schunck, B.G., *Machine Vision*, McGraw-Hill, New York, 1995, pp. 459, 467–468.

[10] Lais, S., "Optical Character Recognition," *Computer World*, Jul. 29, 2002, URL: http://www.computerworld.com/article/2577868/app-development/optical-character-recognition.html [cited 24 February 2015].

[11] Markoff, J., "Researchers Announce Advance in Image-Recognition Software," *New York Times*, 17 Nov. 2014, URL: http://www.nytimes.com/2014/11/18/science/researchers-announce-breakthrough-in-content-recognition-software.html?_r=0 [cited 14 January 2015].

[12] Ahrens, S.G., "Vision-Based Guidance and Control of a Hovering Vehicle in Unknown Environments," M.S. Thesis, Mechanical Engineering Dept., Massachusetts Institute of Technology, Cambridge, MA, 2008, pp. 25–26.

[13]Federal Aviation Administration, *Introduction to TCAS II*, U.S. Department of Transportation, 2011, p. 5.

[14] Landsberg, B., "Safety Pilot: Safety Envelope," Aircraft Owners and Pilots Association, 2008, URL: http://www.aopa.org/News-and-Video/All-News/2008/September/1/Safety-Pilot-Safety-envelope [cited 3 March 2015].

[15] Quanser, "About Quanser," URL: http://www.quanser.com/about [cited 18 January 2015].

[16] Quanser, *Quanser Qball-X4: User Manual*, Doc. 888, Rev. 2.

[17] Quanser, "Qball 2," URL: http://www.quanser.com/products/qball2 [cited 19 January 2015].

[18] NaturalPoint, "About Optitrack," URL: http://www.naturalpoint.com/optitrack/about/ [cited 30 January 2015].

[19] NaturalPoint, "Flex 3 – An Affordable Motion Capture Camera – OptiTrack," URL: http://www.naturalpoint.com/optitrack/products/flex-3/ [cited 30 January 2015].

[20] Logitech, "Logitech Support, Webcam Pro 9000," URL: http://support.logitech.com/product/webcam-pro-9000 [cited 6 February 2015].

[21] Sachs, J., *Digital Image Basics*, Digital Light and Color, 1999, URL: http://www.dl-c.com/basics.pdf [cited 6 February 2015].

[22] Systementwicklung Equasys, "Color Conversion – equasys," URL: http://www.equasys.de/colorconversion.html [cited 3 February 2015].

[23] Erik Van Bilsen Audiovisuality, "Advanced Image Coding," URL: http://www.bilsen.com/aic/colorconversion.shtml [cited 3 February 2015].

[24] Lindeberg, T., "Edge Detection," *Encyclopedia of Mathematics*, URL: http://www.encyclopediaofmath.org/index.php?title=Edge_detection [cited 4 February 2015].

[25] Sobel, I., "History and Definition of the Sobel Operator," [online correspondence], 6 Feb. 1989, URL: http://www.researchgate.net/publication/239398674_An_Isotropic_3_3_Image_Gradient_Operator [cited 4 February 2015].

[26] Fisher, R., Perkins, S., Walker, A., and Wolfart, E., "Feature Detectors – Sobel Edge Detector," http://homepages.inf.ed.ac.uk/rbf/HIPR2/hipr_top.htm [cited 4 February 2015].

[27] MathWorks, "Properties of Connected Regions," http://www.mathworks.com/help/vision/ref/vision.blobanalysis-class.html [cited 4 February 2015].

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California