Dynamics, Modeling, Simulation and Control of Mid-Flight Coupling of Quadrotors

by

Daniel Larsson

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2016 by the
Graduate Supervisory Committee:

Panagiotis Artemiadis, Chair
Hamidreza Marvi
Spring Berman

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

Unmanned aerial vehicles have received increased attention in the last decade due to their versatility, as well as the availability of inexpensive sensors (e.g. GPS, IMU) for their navigation and control. Multirotor vehicles, specifically quadrotors, have formed a fast growing field in robotics, with the range of applications spanning from surveillance and reconnaissance to agriculture and large area mapping. Although in most applications single quadrotors are used, there is an increasing interest in architectures controlling multiple quadrotors executing a collaborative task. This thesis introduces a new concept of control involving more than one quadrotors, according to which two quadrotors can be physically coupled in mid-flight. This concept equips the quadrotors with new capabilities, e.g. increased payload or pursuit and capturing of other quadrotors. A comprehensive simulation of the approach is built to simulate coupled quadrotors. The dynamics and modeling of the coupled system is presented together with a discussion regarding the coupling mechanism, impact modeling and additional considerations that have been investigated. Simulation results are presented for cases of static coupling as well as enemy quadrotor pursuit and capture, together with an analysis of control methodology and gain tuning. Practical implementations are introduced as results show the feasibility of this design.

ACKNOWLEDGEMENTS

I would like to thank Dr. Artemiadis for being an excellent research advisor and for continuously challenging me with new problems - I would not be the person I am today without your help. I would also like to thank Dr. Marvi and Dr. Berman for their support and willingness to serve on the thesis committee.

To close friends, family and my parents, Bengt and Rosemarie, for their unrelenting support and encouragement. Finally, I want to thank Dr. Wells for being the best mentor a student can ask for.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

In recent years, quadrotors have received much attention thanks to their versatility in academic research, military operations and private, commercial, applications. In many places around the world, these maneuverable air vehicles are being used in a variety of applications. As discussed in Bloss (2014), these applications range from commercial agricultural data collection and aerial imaging to public security surveillance and border protection. Quadrotors have also, according to Ali *et al.* (2015), been considered for delivering medical supplies to those in need following natural disasters or, as mentioned in Nemati and Kumar (2014), assisting authorities with hazardous material spills. Others, such as Thrun *et al.* (2000) and Michael *et al.* (2012) have suggested that quadrotor use can be extended to assist first responders with search and rescue operations, especially in areas that are particularly perilous to humans. Such example can be found in Michael *et al.* (2012), where a team of robots, including quadrotors, were used to search partially collapsed buildings after earthquakes, potentially preventing further loss of life. Additional applications listed by Ostojic *et al.* (2015) include using quadrotors for aerial recognition, industrial monitoring, and by the United States for use in recognizance missions over several countries. The attractiveness of the quadrotor-type flight vehicle is attributed to different characteristics from one publication to the next. For example, according to Nemati and Kumar (2014), the popularity of the flight vehicle stems from its ability to maneuver in tight spaces whereas Sa and Corke (2012) state that it is in part thanks to the mechanical simplicity of this type of unmanned air vehicle (UAV) that it has become so ubiquitous. It is also stated by Sa and Corke (2011), that is is not

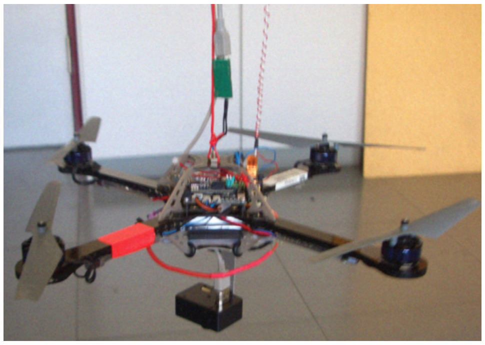

Figure 1.1: Quadrotor with Onboard Surveillance Equipment. © 2009 IEEE

only the versatility and size of the vehicle that make it attractive to researchers and the general public, but also its low cost and availability. The quadrotor vehicle exists in a number of configurations and from a variety of manufactures worldwide. Public, hobby, models can be purchased in many cases for less than $500. Other, academic, professional, and research oriented offerings exist from manufactures such as Ascending Technologies GMBH and 3DR robotics. There are other, mid-cost, models utilized by both professional and private users. One such product offering is that of the Parrot AR drone. Although there are commercial, ready to fly options available, it is not uncommon to see cost or task optimized solutions presented by various individuals or universities. For example, Schmidt (2011) discusses, in part, the custom design of a "combat-theater" quadrotor and Sa and Corke (2011) discusses the cost benefit and control methods to purchasing readily available open source vehicles. The quadrotor is an adaptable platform, consisting of a frame, four total motors and propellers as

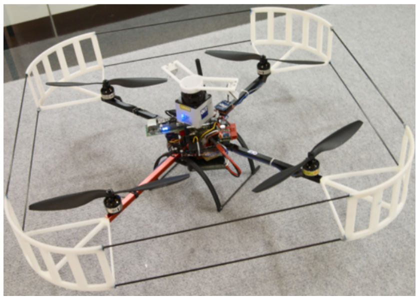

Figure 1.2: Quadrotor with Mounted Laser Scanner. © 2012 IEEE

Figure 1.3: Parrot AR Drone. © 2012 IEEE

well as an inertial measurment/autopilot unit. The global position of the system is controlled by changing the roll and pitch angles, also known as the vehicle attitude, in order to direct the thrust vector in the direction of desired travel. This results in a net force imbalance, and therefore a net acceleration, thereby causing a movement in inertial space. The motors and propellers are configured in such a manner that two of the four propellers rotate counterclockwise whereas the remaining two rotate clockwise. Assuming that all the propellers are identical and rotate at the close to the same rotational velocity in equilibrium, there will be no net torque transmitted to the flight vehicle. The yaw angle is controlled through differential motor rotational velocities, as a net torque is applied to the vehicle from the aerodynamic drag experienced by the propellers, and will be opposite the rotational direction. Therefore, by knowing the direction of the resulting motor torque, it becomes possible to rotate the quadrotor about its z axis, thereby changing the yaw.

Considering their popularity, it comes as no surprise that much work has been done to explore and study various aspects of the system, both from a dynamics and a control/feedback point of view. One example of this is presented in Schmid *et al.* (2013), where a quadrotor is equipped with sensors in order to allow the flight vehicle to autonomously navigate both in and outdoor, follow waypoints provided to it by a user, perform obstacle avoidance onboard and to create 3D maps of its environment. Additional studies have been conducted by Blosch *et al.* (2010) in related applications, working with vision based navigation of quadrotors in unknown, unstructured environments. Other researchers have used MAVs together with ground robots in order to collaboratively complete a given task. Examples of this are provided in the works of Michael *et al.* (2012) and Rudol *et al.* (2008). In Rudol *et al.* (2008), an MAV (quadrotor) vehicle is used together with a ground robot in order to establish autonomous indoor navigation. This work is similar to the previously mentioned publication by Michael *et al.* (2012), where a team of ground and arial robots are used in order to map and to explore an earthquake-damaged building. Other quadrotor vehicles are designed specifically to be deployed in combat situations, as is discussed in Schmidt (2011).

Other publications have focused on the aerodynamic interaction and modeling of a quadrotor. Such work includes that presented by Huang *et al.* (2009) and Hoffmann *et al.* (2007), where a study of the aerodynamic effects outside of the hover flight regime is conducted, and the development of compensatory control algorithms are presented to alleviate some of the adverse effects. Additionally, Huang *et al.* (2009) considers the dynamics of blade flapping, which is done to attenuate the resulting moment applied to a rotary wing vehicle in forward flight due to a non-symmetric flow field experienced by the propellers. It should be noted that blade flapping as applied to quadrotors has also been studied in Mahony *et al.* (2012) and is applied to rotary wing vehicles in Leishman (2006). In addition to studying the forward flight aerodynamics, publications such as Brandt and Selig (2011) focus on the study of propeller efficiency as a function of Reynolds number while Simoes (NA) has examined co-axial propeller configurations for quadrotors and performed empirical testing and data collection to compare with theoretical, control volume derived results such as those found in Leishman (2006). There has also been an abundance of research done in regard to quadrotor control and sensory feedback methods that can be utilized to either track the flight vehicle, or be adapted onboard to serve as mission equipment.

Examples of this type of work can be found in Achtelik *et al.* (2009), where a motion feedback system is utilized in order to track the flight vehicle in inertial space and an onboard inertial measurement system is used to set the desired quadrotor attitude. Others, such as Nemati and Kumar (2014) have explored the possibility of adding additional control actuators to tilt only the rotors, instead of the entire vehicle, in an attempt to eliminate the under actuation of the system.

The present thesis presents a new approach: modeling the coupling dynamics and control of two quadrotors. This new approach presents a host of new issues, which include coupling mechanics and methods, influences of one quadrotor on the other during flight as well as controller gain tuning to produce desired system response characteristics. Presentation and discussion of a systematic way of mathematically modeling impact forces, quadrotor equations of motion in both the single and coupled states as well as discuss some other complications that may be important to consider in practical implementation. A magnet-based way of mechanically coupling the quadrotors is proposed. Finally, a simulation has been built to model the dynamics of coupled flight in various situations including (1) stationary coupling, (2) intercept coupling with no target maneuver and (3) intercept coupling with target maneuvering. Simulation results are presented together with a discussion regarding controller gain tuning and feasibility of implementation.

The idea of coupled quadrotors presents a new path for many current applications to be extended. It also opens up new avenues that previously have not been considered, such as the pursuit and capture of enemy quadrotors. Also, this idea may extend and improve team lift operations allowing both vehicles to lift a single standard object at the center of gravity, simultaneously improving the inherent stability of the system - a concept that may become more important as the quadrotor begins to be considered, according to Ali *et al.* (2015), as a package delivery platform.

Chapter 2

QUADROTOR DYNAMIC MODELING

2.1 Current Dynamic Model of a Quadrotor

The popularity of the quadrotor has made the single system dynamics and control a relatively well studied field. It is addressed in many publications, such as Mahony *et al.* (2012); Mellinger *et al.* (2012); Mellinger (2012), and is here included for the sake of completeness. Although the dynamics of the quadrotor may be relatively well studied in isolation, a few important distinctions for the case of a coupled system are vital for a comprehensive understanding of the entire simulation process.

Considering the special orthogonal group in three space $SO(3)$, a quadrotor has a total of six degrees of freedom. These are the x, y, z linear forces/accelerations as well as the moments about each of these three axes. To begin, we consider the free body diagram shown in figure (2.1). When considering the forces acting upon a body in three dimensional space, it is often easier to express the forces in the body-fixed frame of reference. A vector in this body-fixed frame of reference will be denoted by $[x, y, z]^b$ while quantities expressed in the inertial frame will be displayed as $[x, y, z]^i$. All quantities written in the body frame must be related to the inertial system in order to write the equations of motion from Newton's second law. In order to do this, a rotation matrix is constructed utilizing the roll-pitch-yaw Euler angles. Using this set of rotations, the body frame is first yawed about the original z-axis. The resulting intermediate system is then pitched about its y-axis and finally the second intermediate system is rolled about its x-axis. Assuming that the inertial and body frames were originally aligned, this set of three consecutive rotations will construct a

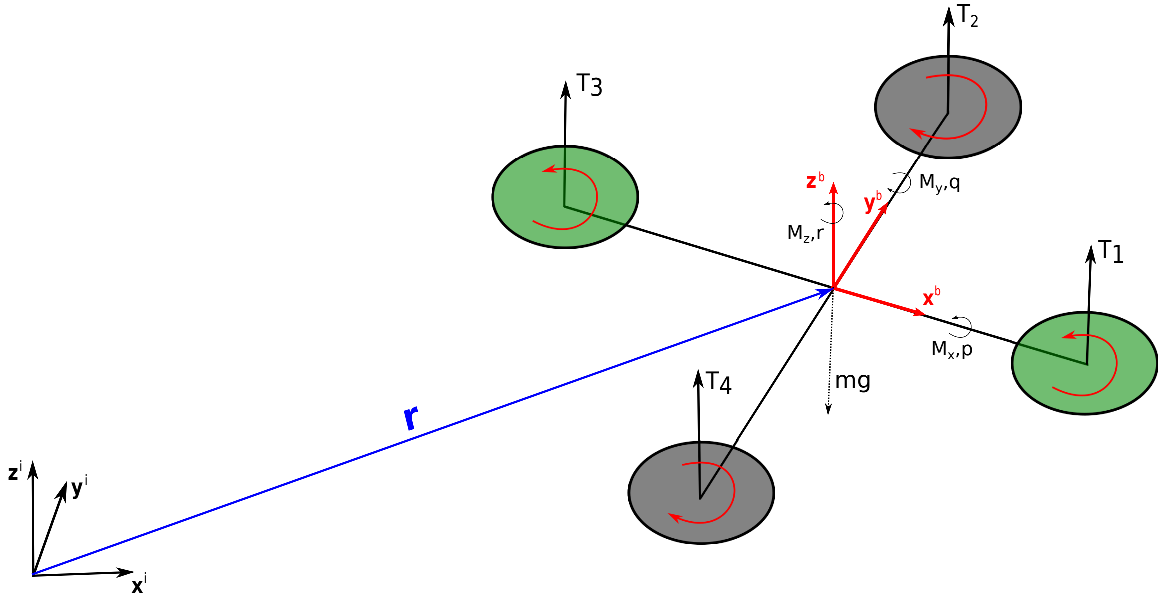

Figure 2.1: Free Body Diagram for the Single Quadrotor.

transformation from the inertial to body frame and vice versa. The resulting rotation matrix is shown below. Note that this representation uses the shorthand notation $cos(x) = c_x$ and $sin(x) = s_x$:

$$\mathbf{R}_b^i = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{2.1}$$

With the above rotation matrix, it becomes possible to express quantities originally in the body frame in the inertial reference system. Therefore, the translational equations of motion become:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}^i = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}^i + \mathbf{R}_b^i \begin{bmatrix} 0 \\ 0 \\ \sum\limits_{n=1}^{4} T_n \end{bmatrix}^b \tag{2.2}$$

Here, m represents the mass of the quadrotor, T_n is the thrust of motor n, g is the gravitational acceleration and $[\ddot{x}, \ddot{y}, \ddot{z}]^T$ are the inertial x, y and z accelerations.

Equation (2.2) provides a set of three non-linear state equations that, when solved, will give inertial position and velocity. While this information is essential to the position control of any air vehicle, it does not form a compete set for controlling a quadrotor. In order to form this complete set, we must consider the rotation dynamics arising due to the applied torques on the vehicle. Once again reviewing the free body diagram in figure (2.1), we obtain that the applied moments for the single quadrotor case must follow the equation below:

$$\mathbf{M} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}^b = \begin{bmatrix} l(T_2 - T_4) \\ l(T_3 - T_1) \\ \tau_2 + \tau_4 - \tau_1 - \tau_3 \end{bmatrix}^b \tag{2.3}$$

In the above equation, $[M_x, M_y, M_z]^T$ are the applied moments along the *body* x, y, z axes, l is the distance from the center of gravity to the center of rotation of any motor and T_n is the thrust of motor n. Also, in this representation, τ_n represents the equal and opposite moment applied to the quadrotor due to the torque applied to the propeller shaft. This torque is dependent on the rotational direction of the shaft and is here shown to be consistent with that shown in the free body diagram in figure (2.1).

While the expression for the applied moments has been found, one must also note that this result is written in the body frame reference system. If Newton's second law were to be directly applied to this relation, an incorrect result would be obtained. Instead, we must consider the time rate of change of angular momentum and accounting for the non-inertial coordinate frame appropriately. Assuming that the coordinate system shown in the free body diagram is approximately the principle axes, the angular momentum can be written as:

$$\mathbf{H} = \mathbf{I}^b \omega^b \tag{2.4}$$

Above, $\mathbf{H}$ is the angular momentum vector, $\mathbf{I}^b$ is the moment of inertia tensor and ω^b is the rotational rate of the quadrotor with all quantities expressed in the body frame. Taking the time derivative of this expression and considering that the body frame components of angular rates will be denoted p, q, r for roll, pitch yaw, respectively, we obtain the attitude, or rotational, dynamics of the system. This set of equations can be rearranged to yield another collection of three non-linear state equations.

$$\mathbf{I}^b \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}^b + \begin{bmatrix} p \\ q \\ r \end{bmatrix}^b \times \mathbf{I}^b \begin{bmatrix} p \\ q \\ r \end{bmatrix}^b = \mathbf{M} \tag{2.5}$$

Completing the dynamics of a quadrotor will be the equations governing the rate of change of the Euler angles. This must be considered as any rotational action in the body frame will result in a change in quadrotor attitude and therefore its Euler angles. When deriving the expression relating the body reference angular rates to that of the Euler angles, we must consider the order of rotation and the respective angular velocity of each axis. Doing this for the order of rotation discussed here, which can also be found in Etkin and Reid (1996), the following is obtained:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.6}$$

The above differential equations, construct a system of twelve non-linear, coupled, first order state equations that can then be solved given initial conditions. These state equations are solved using the MATLAB built-in ordinary differential equations solver ode45. The input to the quadrotor is modeled as the rotational frequency of each motor, thereby constructing a multiple-input-multiple-output (MIMO) system. The rotational frequency of each motor is commanded by a proportional integral derivative

(PID) controller, using feedback of the current orientation (attitude), position and velocity of the vehicle. These considerations are discussed in a later section.

2.2 Extended Dynamic Model for the Coupled Configuration

The coupled quadrotor case presents a variety of added issues that must be considered in order to properly model and apply the dynamics in simulation. Some of the questions that must be answered are:

1. How is the connection to be done?
2. How will having an additional quadrotor attached to the first influence the dynamics of the system?
3. Are there any additional complications that may arise due to a specific design or configuration?

Many of these questions are answered in this section and we begin by modifying the already presented equations of motion to depict the change in dynamics.

The equations of motion that are to be presented here make the following assumptions:

1. The location of the added quadrotor is known (by design) to the first vehicle (see figure (2.2)).
2. The geometry of both quadrotors is known.
3. The system behaves as a rigid body after coupling.
4. The connection between the two vehicles is instantaneous.
5. The yaw angle is controllable onbat least one vehicle.

The assumption of an instantaneous connection is applied only to the moment of switched dynamics. That is, the time instance in which two quadrotors exist as separately controllable bodies to that when they behave as a single rigid body is assumed to be instantaneous. The modeling of the connection forces and disturbances is to be discussed.

Considering the free body diagram for the coupled system (figure 2.2), the modified equations of motion for the translational dynamics can be written as:

$$m_T \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}^i = \begin{bmatrix} 0 \\ 0 \\ -m_T g \end{bmatrix}^i + \mathbf{R}_b^i \begin{bmatrix} 0 \\ 0 \\ \sum\limits_{n=1}^{8} T_n \end{bmatrix}^b \tag{2.7}$$

where $m_T = m_1 + m_2$ is used to represent the total mass of the system. Equation (2.7) also assumes that control over the second quadrotor is possible after the connection is performed. If this is not the case, the summation term in equation (2.7) would remain the same as that in equation (2.2). However, the content presented in here assumes that the system transforms from a four to an eight input system at the time of impact.

The more difficult challenge lies in modeling the rotational motion of the quadrotor. Namely, the second quadrotor's weight will cause a moment on the first in three dimensional space (figure (2.2)). This moment must be appropriately modeled in order for an accurate depiction of reality in simulation to be accomplished. To model this force and resulting moment, it is practical to first must consider that the weight vector of the second mass always points downward in the *inertial* frame. Furthermore, the moment that this added mass applies to the first is highly dependent on (1) the location of the second mass relative to the first, and (2) the orientation of the vehicle in space. In addition, one must consider that the rotational dynamics equation

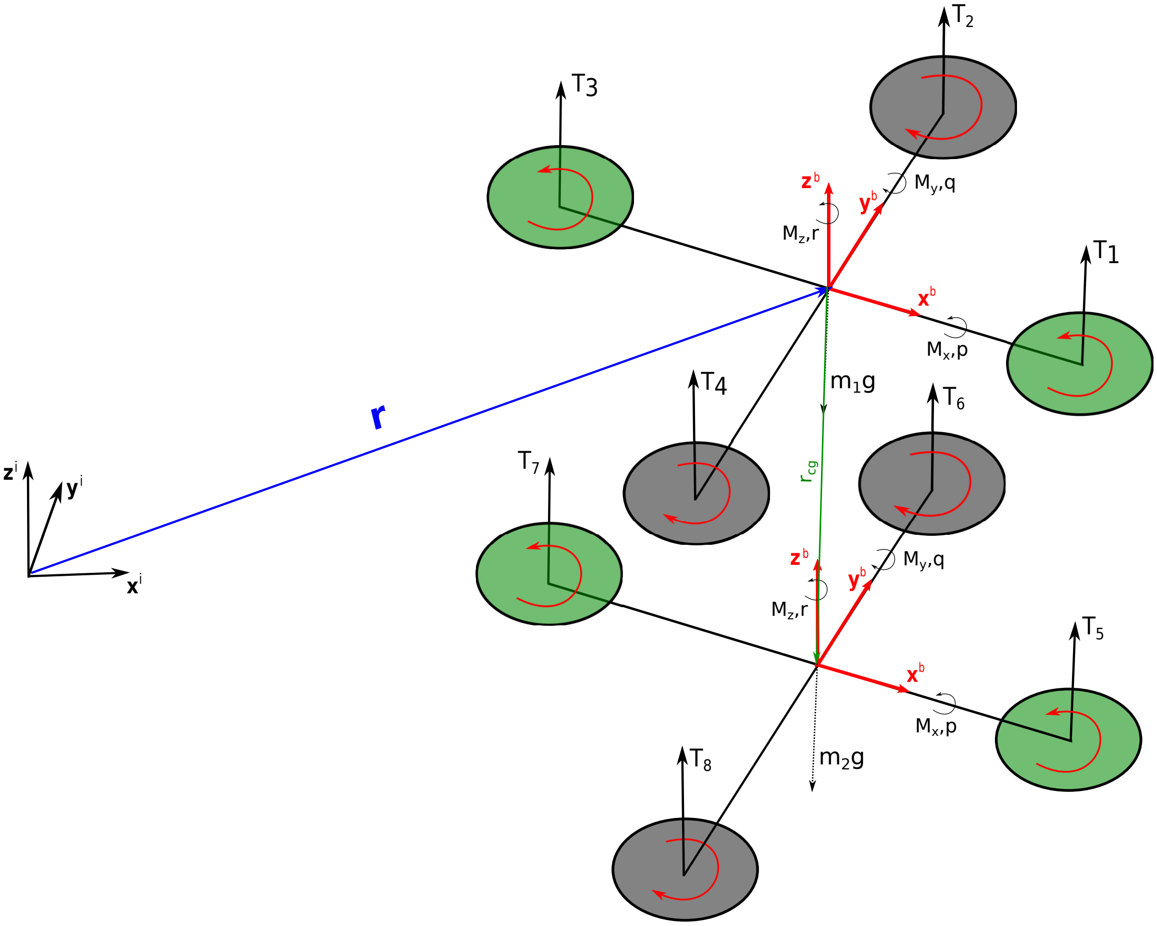

Figure 2.2: Free Body Diagram for the Coupled Quadrotor System.

derived for the single quadrotor case is valid in the body frame, since the appropriate actions have been taken to compensate for this non-inertial frame of reference. As such, it is suggested that the moment on the first mass due to the second be modeled as:

$$\mathbf{M_2} = \mathbf{r_{cg}}^b \times \mathbf{F_{cg}}^b \tag{2.8}$$

where $\mathbf{r_{cg}}^b$ is a vector, expressed in the body frame of quadrotor one, defining the location of the center of gravity of quadrotor two relative to that of one. Using this numbering method, it is assumed that the chase, or upper, quadrotor is referred to as quadrotor one, with the lower then being quadrotor two. With this in mind, $\mathbf{F_{cg}}^b$

is defined by the following equation:

$$\mathbf{F_{cg}}^b = (\mathbf{R}_b^i)^T \begin{bmatrix} 0 \\ 0 \\ -m_2 g \end{bmatrix}^i \tag{2.9}$$

In simulation, this calculation was simplified through the use of skew-symmetric matrix operations. Any vector with components $[v_x, v_y, v_z]^T$ has a skew-symmetric representation as follows:

$$[\mathbf{V}\times] = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \tag{2.10}$$

It is now possible to consolidate the cross product in equation (2.8) with the expression in (2.8) to a single mathematical relation that will both account for the attitude of the vehicle and the location of the second mass, thereby making this model practical for general applications. This is done as follows:

$$\mathbf{M_2} = [\mathbf{r_{cg}}^b \times](\mathbf{R}_b^i)^T \begin{bmatrix} 0 \\ 0 \\ -m_2 g \end{bmatrix}^i \tag{2.11}$$

Modifying the moment of inertial and accounting for the added motor thrusts, the complete set of equations describing the rotational dynamics of the coupled system is obtained as follows:

$$\mathbf{M_1} = \begin{bmatrix} l_1(T_2 - T_4) + l_2(T_6 - T_8) \\ l_1(T_3 - T_1) + l_2(T_7 - T_5) \\ \tau_2 + \tau_4 + \tau_6 + \tau_8 - \tau_1 - \tau_3 - \tau_5 - \tau_7 \end{bmatrix}^b \tag{2.12}$$

where $\mathbf{M_1}$ is used to denote the applied moment due to differential thrust and l_1 and l_2 are the moment arms of quadrotors one and two, respectively. Then, with $\mathbf{I}_2^b$ being the body frame moment of inertia of the coupled system, the final equation can be obtained, as shown below. Equation (2.13) describes the rotational dynamics of the coupled system.

$$\mathbf{I}_2^b \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}^b + \begin{bmatrix} p \\ q \\ r \end{bmatrix}^b \times \mathbf{I}_2^b \begin{bmatrix} p \\ q \\ r \end{bmatrix}^b = \mathbf{M_1} + \mathbf{M_2} \tag{2.13}$$

Modified equations (2.7) and (2.13) together with the differential relations for the Euler angles creates another set of twelve non-linear, coupled, state equations that can be solved with appropriate initial conditions.

2.3 Additional Dynamic Considerations

2.3.1 Proposed Coupling Mechanism

In order to couple the two quadrotors in mid-flight, a mechanical structure must be constructed to provide and house the attachment mechanism. This mechanism can be created to be passive, examples of which would be utilizing magnets, adhesives, or Velcro-type products in order to provide an attachment surface. Alternatively, this mechanism can be built as an active system element, such as a gripping tool or electro magnet configuration. Considering that an active element will introduce another power consuming component to the system, and the adverse effects this has on the flight endurance time, it was decided that a passive, magnet based approach would best suit the system.

The design process of the coupling structure now turns to the selection of the individual magnets and the resulting mechanical structure. The main considerations are:

1. The distance between quadrotors.

2. Individual Magnet strength/pull force.

3. Accommodation of Vicon markers.

4. Ability to couple at generic relative yaw angles.

5. Provide an adequate coupling area/surface.

In the pursuit of satisfying (1) - (5) above, a generic structure was constructed, as shown in figure (2.3). However, prior to the building process, the type and strength of magnets to be used must be established. There are a number of commercially available magnets, however most of which have the shortcoming of either being too small, or excessively strong in pull force. The first aforementioned deficiency runs the risk of not providing an adequate coupling surface and/or area. While this may not seem detrimental to the system as the quadrotors are position controlled, real-world dynamics will cause small imperfections in the positioning of the vehicle, making it beneficial to increase the coupling area. The second flaw, namely selecting magnets that are too strong, has the adverse effect of causing a large disturbance when the vehicles are coupled, placing an increased strain on both the high-level and low-level controllers in order to maintain stability. Additionally, the system is not designed to handle extremely aggressive maneuvers when in the coupled configuration, further reducing the need for strong magnets. Finally, the desired characteristic that the mechanism be able to couple at varying relative yaw angles dictates that the magnet(s) be centrally mounted and be either circular or square in shape. The desire to couple at varying relative yaw angles has benefits when considering the chase-and-couple scenarios as well as when considering some potential aerodynamic effects that are to be discussed.

Considering the parameters and discussion above, a pair of $1\frac{1}{8}$ inch diameter, neodymium magnets were selected. These magnets weigh less than 8 grams and provide a pull force of approximately 7 pounds. The magnets are mounted on a fiber glass-based structure which is attached to the quadrotor by two #8-32 treaded rods with corresponding locking nuts to hold the mechanism in place. The lower locking nuts can be moved up and down the rod, thereby adjusting the height at which the magnet is above the quadrotor inertial measurement unit (IMU) or below the vehicle battery (upper quadrotor). Additionally, the lower coupling structure is extended in order to potentially accommodate additional Vicon markers. Each assembly weighs approximately 50 grams and is shown in figure (2.3). A description of components in figure (2.3) can be found in table (2.1).

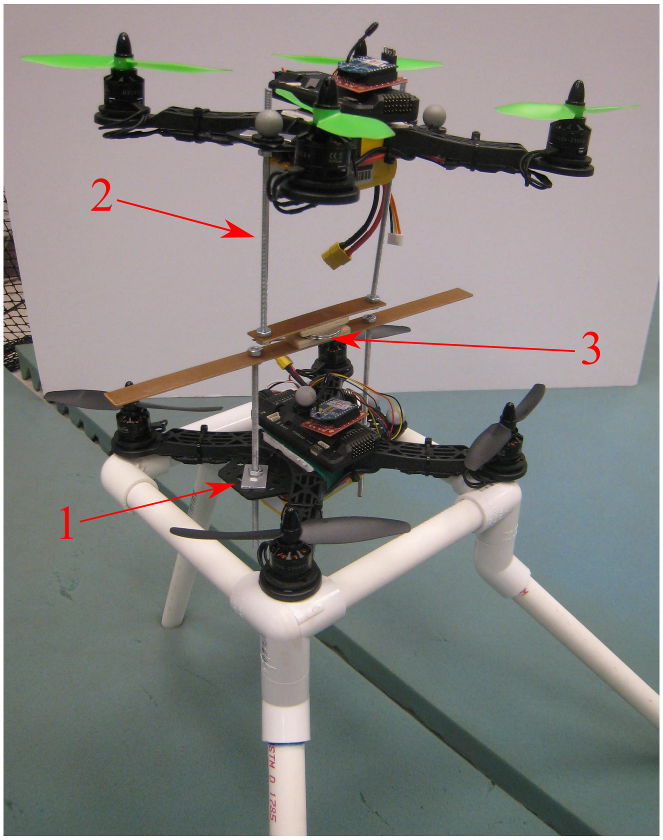

Figure 2.3: The Quadrotor Coupling Mechanism.

Table 2.1: Coupling Mechanism Components.

Label Number	**Component**
1	Aluminum attachments to quadrotor frame.
2	#8-32 Threaded Support Rod/Structure.
3	Magnet Location

2.3.2 Coupling Perturbance Modeling

The complexity of the magnetic force and its dependency on geometry severely complicate the issue of modeling the disturbance at the time of coupling. Many publications, such as Akoun and Yonnet (1984); Furlani (1993), derive mathematical expressions that relate the magnetic force to the detailed geometry of the two permanent magnets considered. This level of detail is beyond the current scope, however according to Young and Freedman (2007a), if it is assumed that the magnets are small enough and relatively at rest, they can be modeled as point charges and Coulomb's law can be implemented. This approach is employed here as the force between two point charges, as in Young and Freedman (2007a), is given by the following equation:

$$F_{mag} = \frac{|q_1 q_2|}{4\pi\epsilon_0\delta^2} \tag{2.14}$$

In the above expression, δ represents the distance between the two magnets, q_1 and q_2 are the electric charge strengths and ϵ_0 the permeability of free space. Given the non-linearity of the magnetic force in equation (2.14), it is likely that modeling this interaction as linear will not provide for an accurate result. Instead, it is assumed that the upper quadrotor is able to maintain a hover state *relative* to the lower. This can be validated by considering that an increase in thrust may compensate for the added magnetic force, especially noting that the force on quadrotor one is downward in the

z^b direction. This, however, is not the case for the lower quadrotor in that it will not be able resist a force greater than its weight to remain in equilibrium. Therefore, for impact modeling, it is assumed that the lower vehicle non-linearly accelerates toward the upper given that the net force acting on it is that of the magnets. Further, since the force is dependent on the separation distance between the two vehicles, it was decided that, instead of modeling a disturbance force, use Newton's conservation of momentum expressed by the equation below:

$$m_1\mathbf{V}_1^i + m_2\mathbf{V}_2^i = (m_1 + m_2)\mathbf{V}_3^i \tag{2.15}$$

Here, m_1 and m_2 are the masses of quadrotors one and two, $\mathbf{V}_1^i$, $\mathbf{V}_2^i$, $\mathbf{V}_3^i$ are the inertial velocity vectors of vehicles one, two and the combined (post coupling) configuration, respectively. In this representation, $\mathbf{V}_2^i$ includes not only the inertial velocity of quadrotor two, but also the disturbance velocity applied to the system due to the non-linear magnetic force and resulting acceleration prior to coupling, as discussed below.

The challenge is now to find $\mathbf{V}_2^i$, as it is likely to be changed due to the non-linear acceleration discussed above. Therefore, we model this velocity term as follows:

$$\mathbf{V}_2^i = \mathbf{v}_2^i + \mathbf{v}_d^i \tag{2.16}$$

The disturbance velocity, $\mathbf{v}_d^i$, is found through a secondary simulation that finds the velocity of equal mass to the second quadrotor at the time of impact. This result can be further confirmed by employing the work-energy theorem in Young and Freedman (2007b), although using this method will only provide a scalar velocity. However, it is known that the increase in velocity must be in the z^b direction, considering this is the direction of the net force acting on the object. For the purposes of calculating this disturbance velocity, a second, temporary, inertial frame is constructed to be aligned in the same orientation as the body frame. Then, expressing this velocity gain in the original (world) inertial frame, it is concluded that:

$$\mathbf{v}_d^i = \mathbf{R}_b^i \mathbf{v}_d^b \tag{2.17}$$

Having obtained this result, it then becomes possible to solve for the vector quantity $\mathbf{V}_3^i$ and feed the result in as an initial condition into the coupled simulation. This is done within the simulation and accounts for both moving and stationary coupling instances.

2.3.3 Aerodynamic Effects of Coupling

In addition to the mechanical disturbances due to the magnetic force discussed above, consideration must be given to potential aerodynamic effects as, in the proposed coupling configuration, the lower quadrotor will be operating in the downwash of the first (upper) vehicle. Considering that at the present moment we are most concerned with the consequences of any additional constraints due to the altered flow field surrounding the lower quadrotor, a momentum analysis is performed. This method is well documented in a variety of professional publications Leishman (2006) and is applied directly to rotary wing vehicles in Huang *et al.* (2009) and Simoes (NA). Here, consider and discuss the process and results presented in Leishman (2006) and its applicability to the present dynamic system.

Considering the control volume shown in figure (2.4) for the quadrotor system, the governing equations are those of conservation of mass, momentum and energy as applied to fluids. These equations are listed in the order mentioned, in equations (2.18) - (2.20). The equations assume steady, incompressible and inviscid flow.

$$\dot{m} = \iint_S \rho \vec{\mathbf{V}} \cdot d\vec{\mathbf{S}} \tag{2.18}$$

$$\vec{\mathbf{F}} = \iint_S p \ d\vec{\mathbf{S}} + \iint_S \rho(\vec{\mathbf{V}} \cdot d\vec{\mathbf{S}})\vec{\mathbf{V}} \tag{2.19}$$

$$W = \iint_S \frac{1}{2}(\rho\vec{\mathbf{V}} \cdot d\vec{\mathbf{S}})|\vec{\mathbf{V}}|^2 \tag{2.20}$$

In the above equations $\dot{m}$ represents the mass flow rate across surface S, $\vec{\mathbf{V}}$ is the local velocity vector, p is the static pressure, W is the power required, ρ the air density, and $d\vec{\mathbf{S}}$ the incremental unit vector perpendicular to the control volume surface at all points. Figure (2.4) has been included to help in the visualization of some of these variables. In addition to the aforementioned assumptions, it is also assumed that the flow is one dimensional across all control volume surfaces.

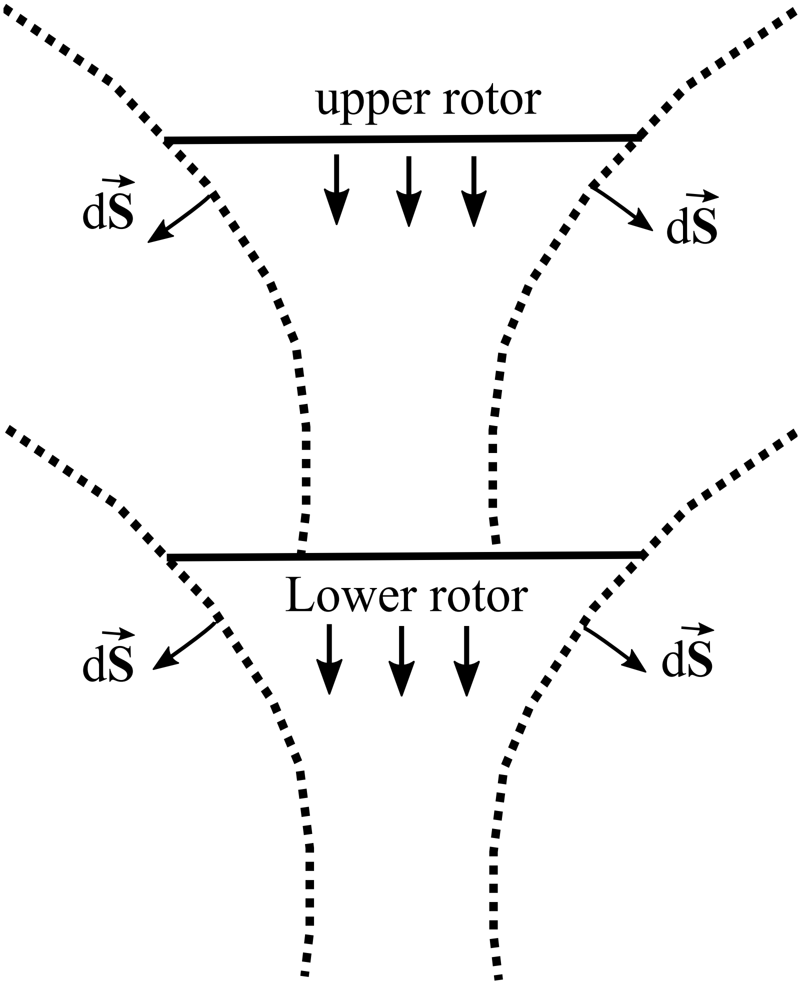

Figure 2.4: Control Volume for the Coupled Configuration.

Applying these conservation equations to a streamtube control volume, assuming that the "actuator disk" area be the same as that of the propeller, and that the rotor is open and not ducted, the results shown in equations (2.21) and (2.22) are obtained for the resulting thrust and power required to hover. Note that the assumption of a streamtube defined control volume implies that there is no mass flow across the sides of the system boundary.

$$T = 2\rho A_{disk}(V + v_i) \tag{2.21}$$

$$P = \frac{T^{3/2}}{\sqrt{2\rho A_{disk}}} \tag{2.22}$$

Where, in the above equations, A_{disk} is the circular rotor area, V is the incoming free stream velocity and v_i is the *induced* velocity at the plane of the rotor. Note that the vector superscripts have been omitted as the assumption of one dimensional flow dictates that these velocities and forces must be scalar, not vector, quantities.

The analysis of the coupled quadrotor configuration is similar to the work presented in Leishman (2006) and Simoes (NA) for the coaxial rotor case. In the reference text Leishman (2006), the coaxial rotor case was studied as when mounted on a single vehicle, as is the case for full size helicopters and coaxial propulsion quadrotors, this becomes a necessity to balance the reaction torque generated by the aerodynamic drag force. The relative rotation of the coaxial rotor system that is generated when the quadrotors are in the coupled configuration can easily be modified to the counter rotating rotor case by offsetting the yaw angles by ± 90 degrees. Alternatively, the yaw angles can be directed to be "in phase" and have two consecutive rotors in the vertical direction spin in the same direction, as shown in figure (2.2). The control logic presented in the above section as well as in chapter 4 require no modification regardless of the relative coupling yaw angle. The differences lie in the computed desired roll and pitch angles that are ultimately sent to the onboard autopilot and also assumes that the current yaw angle of each vehicle can be uniquely determined. A more in-depth discussion regarding the practical implementation is discussed in chapter 4.

Prior to applying the equations discussed above to a coaxial rotor system, baseline values must be established for the value of the ambient density ρ, the required thrust, T and the area of the rotor disk, A_{disk}. To calculate the ambient density, the ideal gas law is employed, shown in equation (2.23).

$$\rho = \frac{p}{RT} \tag{2.23}$$

In the above equation, p represents the ambient air pressure, R the specific gas constant and T the ambient air temperature. The pressure is assumed to be that of a standard day, while air temperature is measured in the laboratory space. Calculated values for the above mentioned variables are tabulated in table (2.2). It should also be stated that the current analysis assumes that the dual quadrotor system is in the hover state, as this will allow slight simplifications in the analysis. Additionally, it is

Table 2.2: Values for Aerodynamic Control Volume Computation.

Variable	**Value**
Ambient density, ρ	1.14 $\frac{kg}{m^3}$
Ambient Pressure, p	97.8 kPa
Ambient Temperature, T	25°C
Specific Gas Constant, R	287 $\frac{J}{Kg \cdot K}$
Rotor Diameter, d_{disk}	0.127 m
Rotor Disk Area, A_{disk}	0.0127 m^2
Total Hover Thrust, T_{Th}	5.1 N
Hover Thrust per Motor, T_h	1.28 N

assumed that the coupling offset distance is great enough so that the lower vehicle is operating in the *verna contracta* of the upper. The verna contracta is the constricting portion of the flow field down stream of the rotor as shown in figure (2.4). It can be shown by employing equations (2.18) - (2.20) that the exit area of the control volume be half that of the disk (A_{disk}) to meet mass conservation. This effect can be seen in figure (2.4), where half of the lower rotor area experiences the downwash from the upper vehicle. Therefore, from an aerodynamics point of view, the inner half of the lower rotor will experience a flow field identical to that of a constant climb at the velocity of the down wash in the verna contracta. Considering that it has been assumed that the system be in hover, the induced velocity term (v_i) can be calculated by utilizing equation (2.21), and can also be found in Leishman (2006) and Huang *et al.* (2009).

$$v_i = \frac{mg/4}{2\rho A_{disk}} \tag{2.24}$$

Using the values in table (2.2) together with equation (2.24), it is found that the induced velocity at the rotor is approximately 7 m/s, making the velocity in the verna contracta 14 m/s. With this information, attention can now be focused on analyzing the lower quadrotor.

Considering that the flow field is now vastly different from the single quadrotor hover case, and not constant in the radial direction of the rotor, re-application of equations (2.18) - (2.20) is necessary. Carrying through this calculation for mass flow, the following is obtained.

$$\dot{m} = \rho A_{disk}(v_i + v_{il}) \tag{2.25}$$

Where, in the above equation, the new variable v_{il} represents the induced velocity that the lower rotor imparts on the air. Here, it is assumed that the lower and upper quadotor produce enough thrust to separately maintain hover. In other words, each vehicle is producing the same amount of thrust. With this assumption, the task becomes to determine the velocity through the disk in order to calculate the required power. This can be done with a similar process to that discussed above, namely by applying conservation of mass, momentum and energy to the lower control volume. In doing this, it becomes possible to determine the required power for the lower quadrotor to maintain a hover state. These calculations show that a 28% increase in the total power required by the system is experience, a result confirmed by Leishman (2006), as compared to two separate quadrotors hovering in quiescent air. It should also be noted that according to Coleman (1997) and Leishman (2006), the separation distance between the two rotors does not cause a marked efficiency increase or decrease. This trend holds as long as the two rotors are not placed *directly* behind each other, as discussed in Leishman (2006). Additionally, in Leishman (2006), it is stated that the estimation of power required, in this case a 28% increase, is normally considered an over predication, which is based on comparison to experimental results.

Chapter 3

QUADROTOR SIMULATION DEVELOPMENT AND STRUCTURE

3.1 Simulation Mechanics and Flow Chart

In order to test the feasibility of the concepts proposed in chapter 2, a comprehensive simulation has been built to propagate the equations of motion. The simulation utilizes a two-stage controller (High Level and Low Level) to simulate quadcopter translations from one position to the next. An overall flow chart of the simulator is given in figure (3.1).

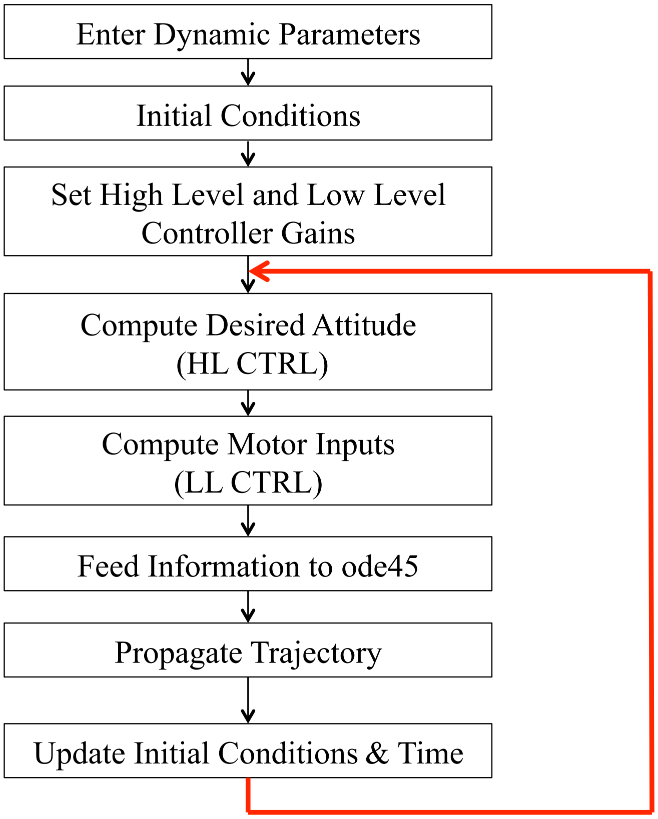
Enter Dynamic Parameters
Initial Conditions
Set High Level and Low Level Controller Gains
Compute Desired Attitude (HL CTRL)
Compute Motor Inputs (LL CTRL)
Feed Information to ode45
Propagate Trajectory
Update Initial Conditions & Time

Figure 3.1: Flow Chart Overview of the Quadrotor Simulator.

Before the simulation is initiated, the complete set of dynamical parameters must be known and provided. The discussion of these parameters is to follow, as the exact number, or type, of values provided is dependent on the dynamic model employed by the simulator. In this case, a modeling similar to that of Mahony *et al.* (2012), Mellinger (2012), and Mellinger *et al.* (2012) is used to relate the electric motor rotational velocities to the forces and torques produced. This thrust and torque model is shown in equations (3.1) and (3.2), respectively.

$$T_i = k_T \omega_i^2 \tag{3.1}$$

$$\tau_i = k_\tau \omega_i^2 \tag{3.2}$$

In the above equations, T_i , τ_i are the thrust and torque generated by motor i, ω is the rotational velocity in radians per second of a given motor, and k_T and k_τ are the characteristic motor thrust and torque coefficients. According to Mahony *et al.* (2012), the characteristic thrust constant, k_T is normally experimentally determined by measuring the rotational rate of shaft i and the resulting thrust produced. This shaft velocity is then used to back-solve equation (3.1) for k_T. On the contrary, it is also stated in Mellinger *et al.* (2012) that the determination of k_τ is a more complicated process, and usually involves both a combination of empirical testing and simulation. This modeling was selected as it is a relatively standard form to mathematically relate the rotational velocity of the rotor shaft to that of the forces and torques applied to the body. Another benefit to using this thrust and torque model, as according to Mahony *et al.* (2012), is that this form of "lumped parameter" model incorporates some of the airframe drag due to the downwash from the rotors. In addition, thanks to the popularity of this model, the validation procedure, which is to be discussed, is greatly simplified and a higher fidelity simulation can be achieved.

The simulator, as shown in figure (3.1), receives inertial frame x, y, z set points and uses a high-level and low-level proportional derivative (PD) controller to drive the position error to zero. In doing so, the high level PD controller calculates desired inertial acceleration by considering the error in position and velocity, as shown in equation (3.3).

$$\ddot{\mathbf{r}}_{\mathbf{des}} = K_p \mathbf{e}_{loc} + K_d \mathbf{e}_{vel} \tag{3.3}$$

The resulting desired linear accelerations are then used in a linearized dynamic model derived from (2.2). Assuming that the roll and pitch angles remain relatively small, and that the yaw angle is known, the translational equations of motion can then be used to relate the desired inertial space acceleration given by (3.3) to that of the quadrotor desired roll, pitch and thrust values. The linearization assumes that the yaw angle is constant, or at least known, at every time instance that the simulation is to propagate the trajectory. These linearized relations are shown in equations (3.4) - (3.6).

$$\theta_{des} = \frac{1}{g}(\ddot{r}_{x,des} \cos\psi + \ddot{r}_{y,des} \sin\psi) \tag{3.4}$$

$$\phi_{des} = \frac{1}{g}(\ddot{r}_{x,des} \cos\psi - \ddot{r}_{y,des} \sin\psi) \tag{3.5}$$

$$T_{des} = m(g + \ddot{r}_{z,des}) \tag{3.6}$$

In the above equations, m is the mass of the quadrotor, g is the acceleration due to gravity, θ_{des}, ϕ_{des} are the desired pitch and roll angles, T_{des} is the required thrust, ψ is the current yaw angle, and $\ddot{r}_{i,des}$ is the desired linear inertial acceleration in direction i, as provided by the high level controller in equation (3.3).

In order to set the desired flight vehicle attitude, it is necessary to employ another controller, as shown in equation (3.7) - (3.10), which is also done in Mellinger (2012). Note that in this representation, the input shown in equation (3.7) is identical to that in equation (3.6) and is repeated here for clarity.

$$u_1 = m(g + \ddot{r}_{z,des}) \tag{3.7}$$

$$u_2 = k_{p,\phi}e_\phi + k_{d,\phi}e_p \tag{3.8}$$

$$u_3 = k_{p,\theta}e_\theta + k_{d,\theta}e_q \tag{3.9}$$

$$u_4 = k_{p,\psi}e_\psi + k_{d,\psi}e_r \tag{3.10}$$

Here, u_1, u_2, u_3, u_4 are the desired forces and moments to be applied to the quadrotor, $k_{p,i}$, $k_{d,i}$ are the proportional and derivative gains for direction i and e_i is the error in variable (or direction) i. The role of the lower level controller (equations (3.7) - (3.10)) is to provide appropriate values for the desired moments in order to achieve a satisfactory response. To relate the desired forces and moments provided by the low level controller above, a mapping must be done which utilizes the relationships shown in equations (3.1) - (3.2). This mapping is shown below, and is also shown in Mellinger (2012).

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} k_T & k_T & k_T & k_T \\ 0 & k_T l & 0 & -k_T l \\ -k_T l & 0 & k_T l & 0 \\ -k_\tau & k_\tau & -k_\tau & k_\tau \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{3.11}$$

The linear transformation matrix above can then be inverted to obtain the desired motor rotational velocities. An overall block diagram of the aforementioned control system is shown in figure (3.2). Note that in figure (3.2), $\ddot{\mathbf{r}}_{p,des}$ has been used to represent position and inertial velocity set points, in an effort to preclude confusion with the body frame angular rates p, q, r.

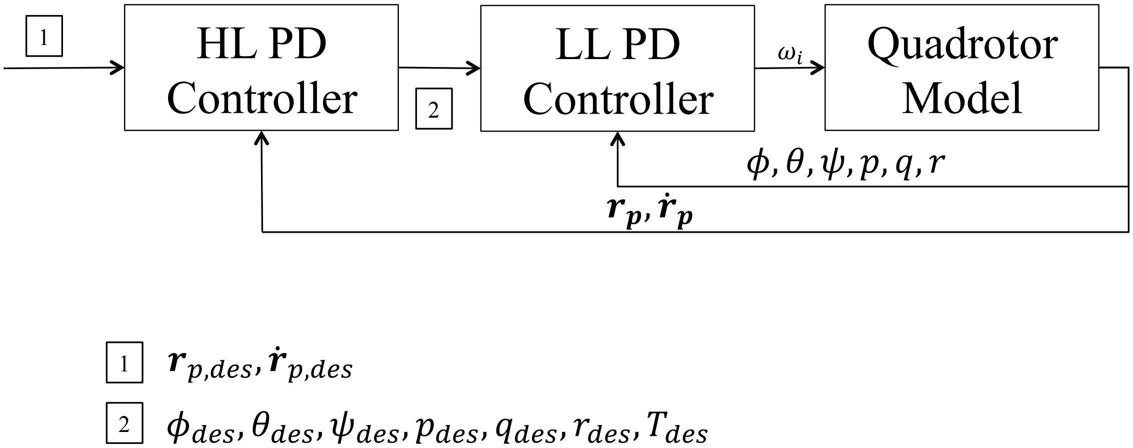
1
HL PD Controller
2
LL PD Controller
ω_i
Quadrotor Model
$\phi, \theta, \psi, p, q, r$
$\boldsymbol{r_p}, \dot{\boldsymbol{r}}_{\boldsymbol{p}}$
1 $\boldsymbol{r}_{p,des}, \dot{\boldsymbol{r}}_{p,des}$
2 $\phi_{des}, \theta_{des}, \psi_{des}, p_{des}, q_{des}, r_{des}, T_{des}$

Figure 3.2: Block Diagram of Simulator Controller.

The quadrotor model shown in figure (3.2) represents the dynamics of the single or coupled system according to the equations of motion derived in chapter 2. The system model is not linearized, and the full non-linear system dynamics are propagated forward in time by MATLAB's ode45 ordinary differential equation solver. The ode45 function utilizes a six-stage, FSAL numerical scheme to solve the system of equations. In this case, a standard state space form is required, as shown in equation (3.12).

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{3.12}$$

Above, $\mathbf{x}$ is the system state vector, $\mathbf{u}$ represents the system input vector, $\mathbf{f}$ is the state equation(s) and $\dot{\mathbf{x}}$ is the time derivative of the system state vector. Considering that the dynamics are second order, a transformation must be made in order reduce the order of the system of equations.

The resulting system consists of 12 non-linear, coupled state equations, which are shown next. Note that, in the equations that follow, the state vector is assumed to be $[x, y, z, v_x, v_y, v_z, \phi, \theta, \psi, p, q, r]^T$ and the abbreviations $c_x = \cos(x)$, $s_x = \sin(x)$, $ta_x = \tan(x)$, and $se_x = \sec(x)$ are employed.

$$\dot{x}_1 = x_4 \tag{3.13}$$

$$\dot{x}_2 = x_5 \tag{3.14}$$

$$\dot{x}_3 = x_6 \tag{3.15}$$

$$\dot{x}_4 = (c_{x_7} s_{x_8} c_{x_9} + s_{x_7} s_{x_9}) \frac{1}{m} \sum_{n=1}^{4} T_n \tag{3.16}$$

$$\dot{x}_5 = (c_{x_7} s_{x_8} s_{x_9} - s_{x_7} c_{x_9}) \frac{1}{m} \sum_{n=1}^{4} T_n \tag{3.17}$$

$$\dot{x}_6 = -g + (c_{x_7} c_{x_8}) \frac{1}{m} \sum_{n=1}^{4} T_n \tag{3.18}$$

$$\dot{x}_7 = x_{10} + s_{x_7} ta_{x_8} x_{11} + c_{x_7} ta_{x_8} x_{12} \tag{3.19}$$

$$\dot{x}_8 = c_{x_7} x_{11} - s_{x_7} x_{12} \tag{3.20}$$

$$\dot{x}_9 = s_{x_7} se_{x_8} x_{11} + c_{x_7} se_{x_8} x_{12} \tag{3.21}$$

$$\dot{x}_{10} = \frac{1}{I_{xx}} (l(T_2 - T_4) + I_{yy} x_{11} x_{12} - I_{zz} x_{11} x_{12}) \tag{3.22}$$

$$\dot{x}_{11} = \frac{1}{I_{yy}} (l(T_3 - T_1) + I_{zz} x_{10} x_{12} - I_{xx} x_{10} x_{12}) \tag{3.23}$$

$$\dot{x}_{12} = \frac{1}{I_{zz}} (\tau_2 + \tau_4 - \tau_1 - \tau_3) + I_{xx} x_{10} x_{11} - I_{yy} x_{10} x_{11}) \tag{3.24}$$

In the above equations, m is the mass of the quadrotor, I_{ii} is the mass moment of inertia around principle axis i and l is the distance from the vehicle center of gravity to the center of rotation of the motor. In the equations above, it is additionally assumed that the quadrotor is symmetric with respect to the x and y body coordinates. As discussed, the formulation of the twelve equations above is based on the dynamic modeling presented in chapter two. Once the simulation has been build with this dynamic model, a validation procedure must be followed in order to verify that the simulator outputs can be trusted. This was done using existing data, and is the topic of discussion in the next section.

3.2 Simulation Validation

Prior to discussing any results obtained by the simulation and before a feasibility analysis can be performed, the dynamic model and simulator must be validated. Validation can be a relatively difficult process, as the exact knowledge of dynamic parameters is usually not available unless a system identification process has been performed. At this time, however, it is the general behavior of the simulator that is under scrutiny, as applied to a generic quadrotor. Using this logic, as long as the dynamic parameters are known for a given quadrotor, then the simulator will output high fidelity results as long as it has been validated against a known solution. As has been discussed, there is a plethora of research publications that study quadrotors, however most are concerned with the response and control of the vehicle given new control algorithms and often do not disclose all the required parameters. It should be further noted that, even if the full list of parameters is disclosed, the input-output data necessary in order to confirm the behavior of the model and simulation is often not. The exception to this is work found in Luukkonen (2011), where a dynamic model and simulator has been build for the single quadrotor case, and input-output data has been disclosed. The list of dynamic parameters is listed in table (3.1). The dynamic modeling presented in Luukkonen (2011) differs slightly from that presented in equations (3.13) - (3.24). Namely, the simulation and modeling presented here neglects aerodynamic drag. Aerodynamic drag is a highly non-linear force, scaling with the square of the forward flight velocity. The work presented in Luukkonen

Table 3.1: Quadrotor Dynamic Properties for Simulation Validation, from Luukkonen (2011)

Parameter	**Value**
m	0.468 kg
k_T	$2.980 \cdot 10^{-6}$ N/rad^2
k_τ	$1.140 \cdot 10^{-7}$ N-m/rad^2
I_{xx}	$4.856 \cdot 10^{-3}$ kg-m^2
I_{yy}	$4.856 \cdot 10^{-3}$ kg-m^2
I_{zz}	$8.801 \cdot 10^{-3}$ kg-m^2

(2011) includes a rudimentary, linear modeling of this force. This linear type of modeling is nearly over simplistic considering the complicated aerodynamic interactions between the quadrotor body, propellers and other components on the flight vehicle which are non-constant and are functions of the vehicle velocity. Additionally, it is generally difficult to construct an all-inclusive simulation which captures these complex aerodynamic interactions throughout the flight envelope, as discussed in Huang *et al.* (2009). It should, however, be noted that quadrotor aerodynamic modeling has been done in a number of ways, as shown in Ostojic *et al.* (2015). Additionally, in order to accurately incorporate the drag force, knowledge of the center of pressure (or drag) must be know, which is a function of the aerodynamic force distributions on the individual components of the flight vehicle.

As previously discussed, in order to validate the developed dynamic model and simulation, known input-output data must be known for the system with parameters listed in table (3.1). An excerpt from Luukkonen (2011) is shown in figure (3.3), displaying the motor rotational velocity input (ω_i) versus time. The resulting simulation output for inertial x,y,z position as well as roll, pitch and yaw angles, according to

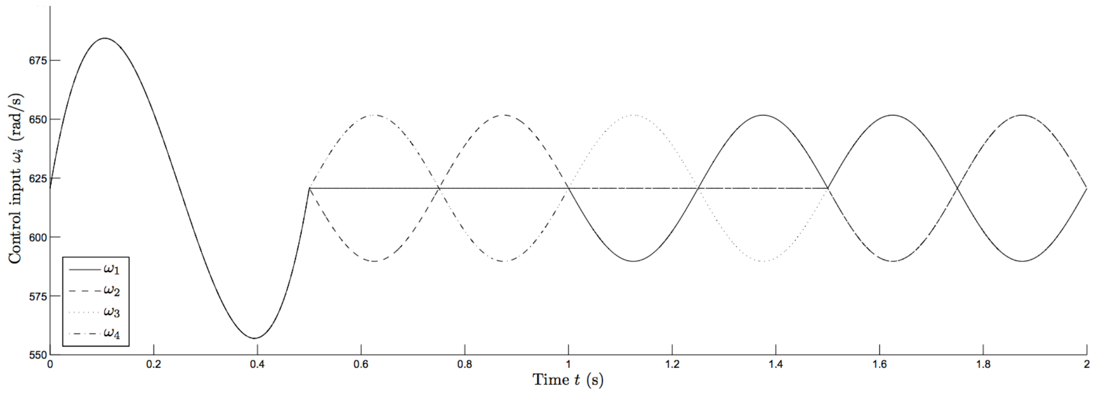

Figure 3.3: Quadrotor Input for Simulation Validation, from Luukkonen (2011)

Luukkonen (2011), is displayed in figure (3.4) and figure (3.5). In order to validate the simulation, reconstruction of figures (3.3) - (3.5) is desired. This process is considerably complicated due to the lack of access to exact motor frequencies shown in figure (3.3). In order to accomplish this recreation of figure (3.3), two possible methods are possible: (1) attempt to discretize the original data, interpolate and feed into the dynamic model or (2) consider only the amplitudes of the wave and phase-shift the signal in time. Considering the sinusoidal shape of the input signal, option (2) was selected. This was done as the amount of data extracted from figure (3.3) is minimized as opposed to option (1), and consists of mainly the approximate time and amplitude of input changes. Knowing the characteristic shape of the input,

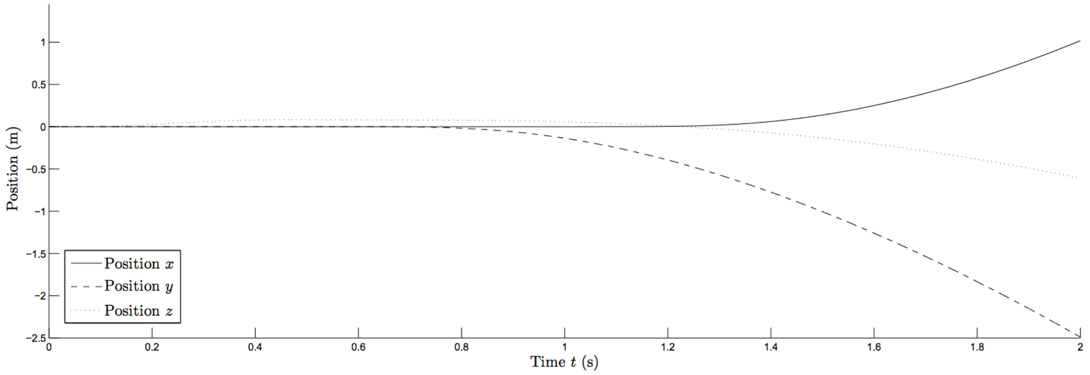

Figure 3.4: Quadrotor x,y,z Inertial Position Output for Simulation Validation, from Luukkonen (2011)

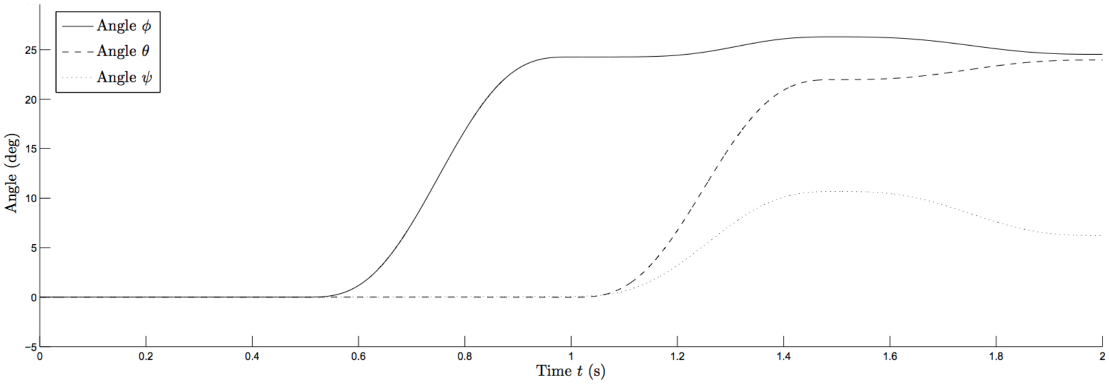

Figure 3.5: Quadrotor Attitude Output for Simulation Validation, from Luukkonen (2011)

the approximate time and amplitude of changes allows for a phase-shift method to be employed that will mimic the signal shown in figure (3.3). The result obtained from the present simulator is shown in figure (3.6). Comparing trends shown in figure (3.6) to that presented in figure (3.3) one can see that the two are nearly identical. Upon obtaining the reconstruction of the input motor angular velocities, the simulation was propagated forward in time and the inertial position and quadrotor attitude were

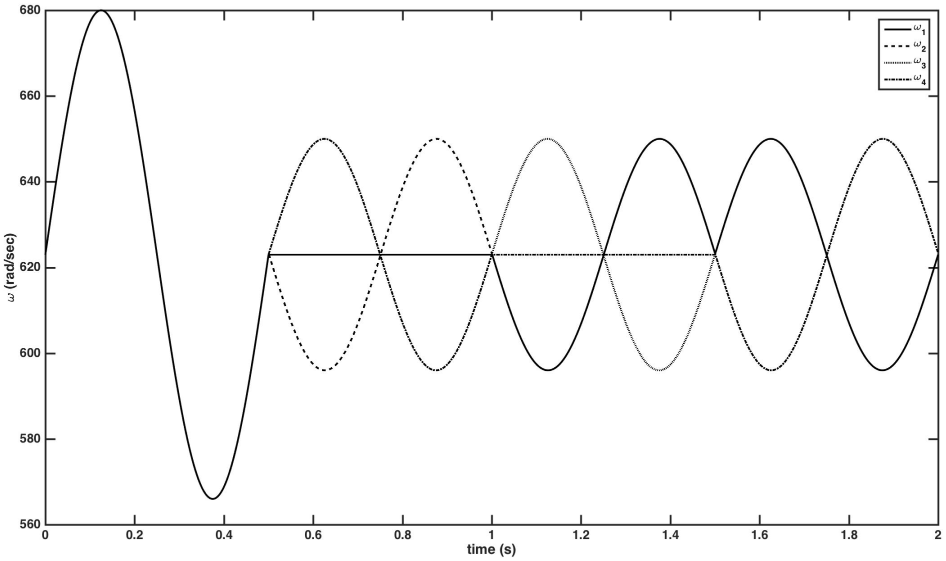

Figure 3.6: Quadrotor Input Obtained from User Simulation.

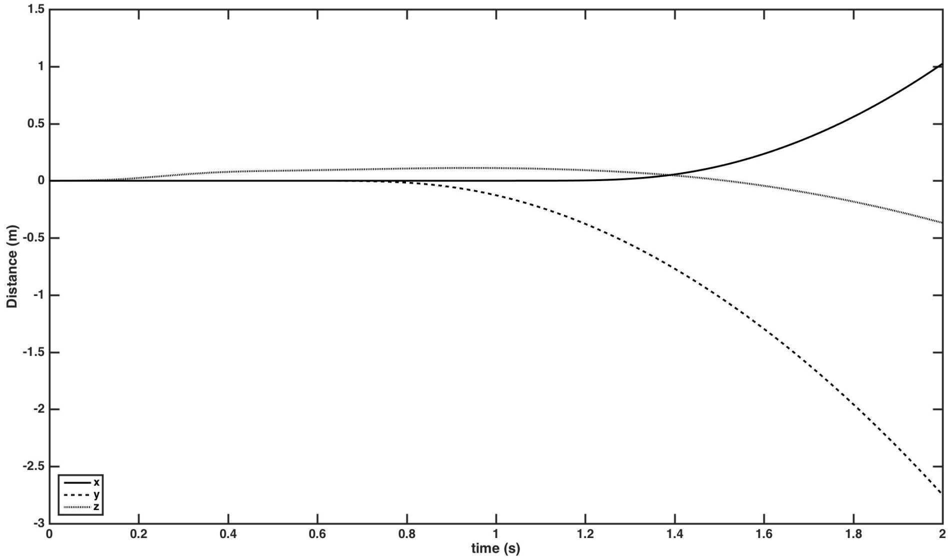

Figure 3.7: Quadrotor x,y,z Output Obtained from User Simulation.

then compared to the representations in figures (3.4) and (3.5). Simulation obtained results are presented in figures (3.7) and (3.8). Comparing figures (3.4) and (3.5) to that shown in (3.7) and (3.8), it is observed that the developed simulation produces a nearly identical output to that used as a reference for validation. However, slight differences are to be expected considering the slight modeling differences and assumptions discussed previously.

3.3 Simulation Results

3.3.1 Controller Methodology & Results - Static Coupling

A preliminary simulation was conducted to show static coupling. This portion assumes that each quadrotor has a controller identical to that presented in equations (3.3) - (3.10). Using this controller, each vehicle is started from an arbitrary position in inertial space then commanded to fly to the same location and maintain hover. The first (upper) quadrotor's position is adjusted to account for the linkage distance

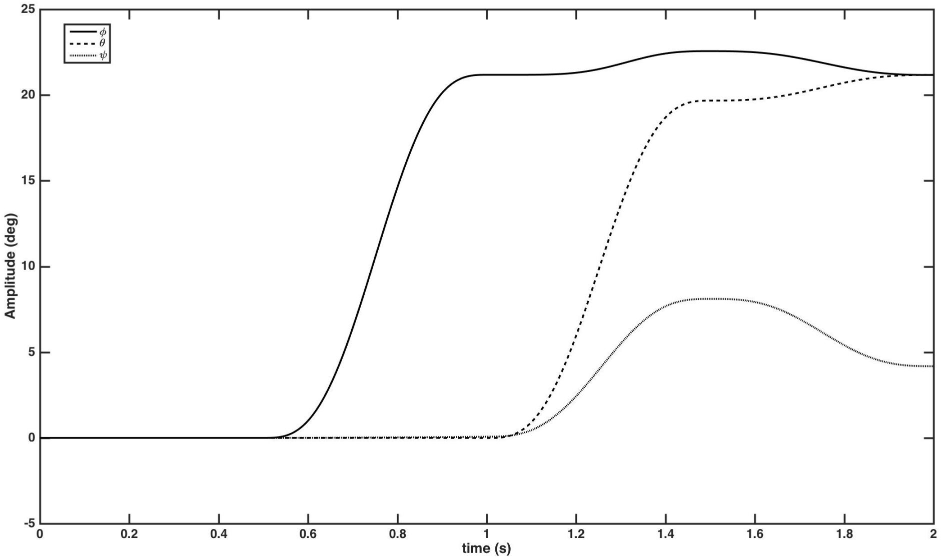

Figure 3.8: Quadrotor Attitude Output Obtained from User Simulation.

in order to simulate a more accurate depiction of the event. The individual quadrotor gains are tuned as to provide a satisfactory rise time and for the system to behave approximately critically damped as shown in figure (3.9). For the first stage of flight, the controllers are set to be identical. However another iteration process to tune the controller gains was done for the coupled system, as the altered dynamics will change the open loop poles of the plant. The gains for the coupled systems are set dependent on the application and anticipated maneuvers. In the case shown here, the system gain values are set to provide another critically damped response, considering that zero altitude corresponds to ground level and any overshoot in this direction would likely prove to be catastrophic. This third controller assumes control over the entire system at the time of coupling, and from this point on desired locations and velocities are set as those of the upper quadrotor. In the simulation results shown in figure (3.9), the coupled system is set to return the lower quadrotor safely to the ground as to emulate an air rescue of a vehicle with low battery. Considering that

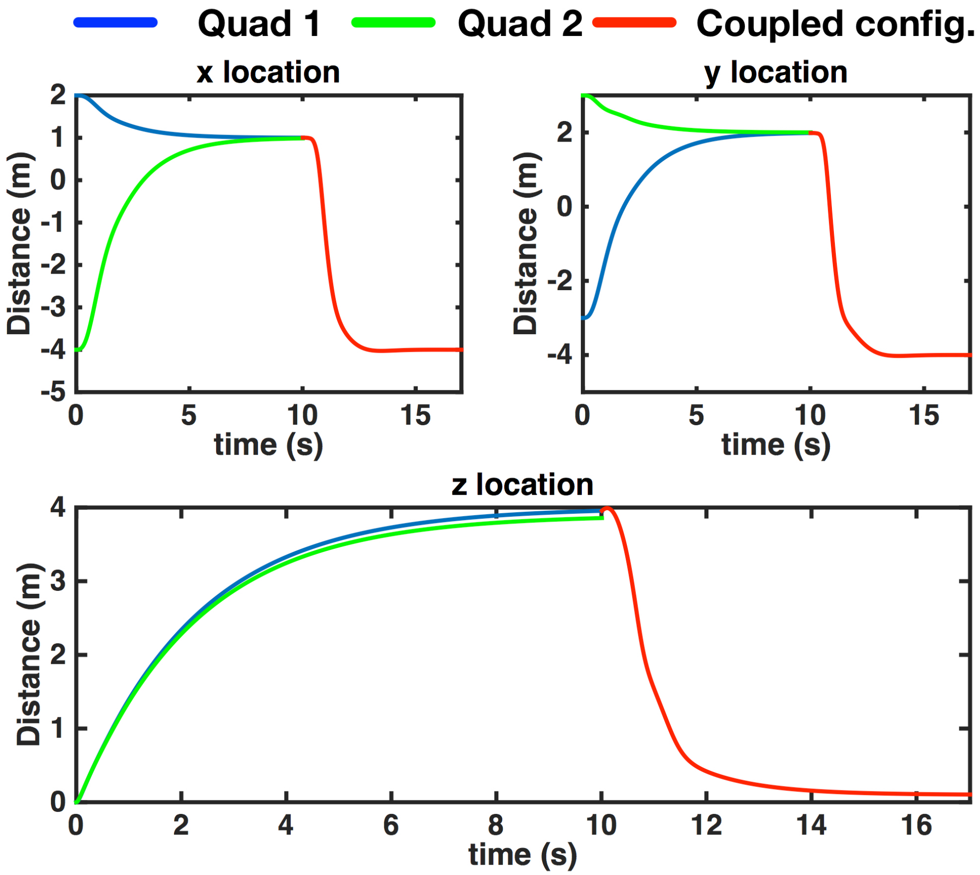

Figure 3.9: Simulation Results Showing Inertial x,y, and z Position for the Static Coupling Scenario.

it is the position of the upper quadrotor that we control, we must set the altitude of the final system to account for the linkage offset in order to ensure a safe landing for the lower vehicle. It is for this reason that the coupled system results in figure (3.9) never reaches an altitude of zero.

3.3.2 Controller Methodology & Results - Interception Coupling

The controller for the interception stage of flight differs somewhat from that for static coupling due to the increased complexity of the problem. In this stage of simulation, one vehicle is commanded to fly at a given velocity that may change at any point in flight, while the second is then to pursue and couple to the original

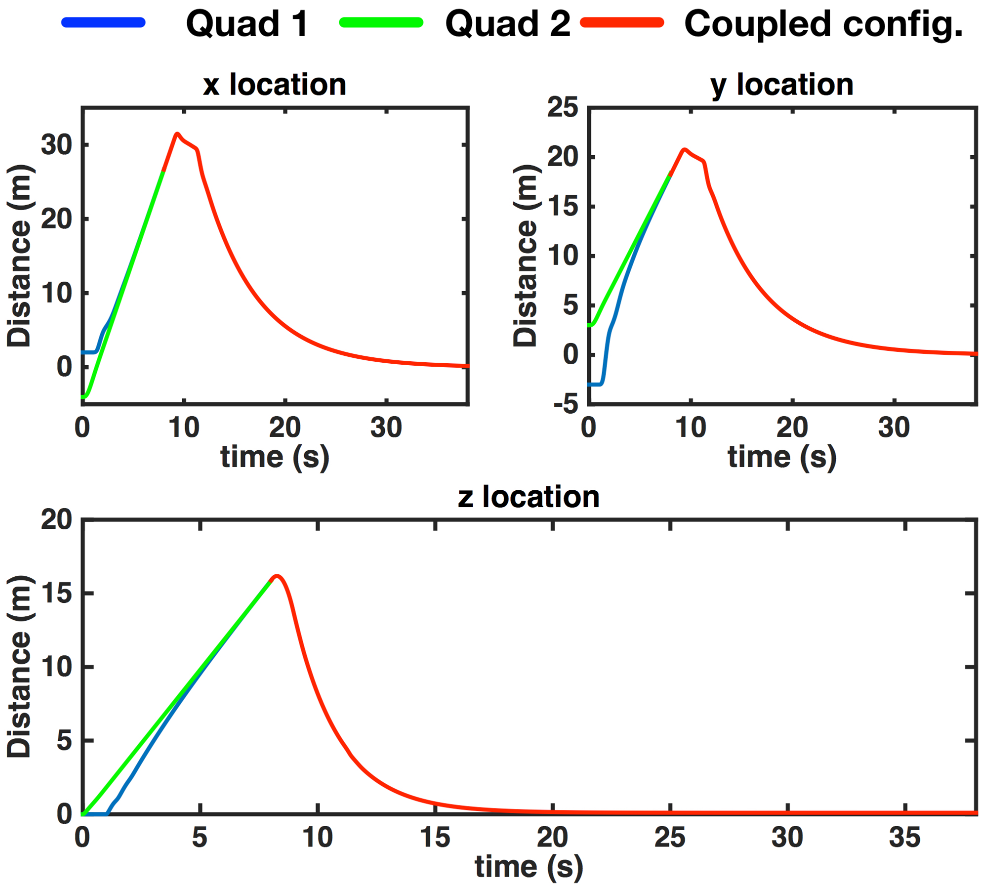

Figure 3.10: Simulation Results Showing Inertial x,y, and z Position for the Non-Static, No Target Maneuvering Coupling Scenario.

quadrotor. This pursuit mode is completed by modifying the error in the controller from that of absolute to relative. In other words, the controller on board the chasing vehicle is modified so that the error it eliminates is that of relative position and velocity, i.e.:

$$\ddot{\mathbf{r}}_{desired} = \mathbf{K}_P^{int}\boldsymbol{\Delta}_{pos} + \mathbf{K}_D^{int}\boldsymbol{\Delta}_{vel} \tag{3.25}$$

Here, $\boldsymbol{\Delta}_{pos}$ and $\boldsymbol{\Delta}_{vel}$ are the *relative* position and velocity errors between the chase and enemy quadrotor and $\mathbf{K}_P^{int}$, $\mathbf{K}_D^{int}$ are the gains of the chase vehicle's controller.

In addition, the desired yaw, roll and pitch angles of the first quadrotor are set to match those of the second ensuring a secure and feasible connection. The above controller functions in cases where the enemy vehicle maneuvers as well as those situations where it does not.

Simulation results are shown in figures (3.10) and (3.11) but for the non-static coupling scenario simulations. The results presented in figure (3.10) are for the case of quadrotor pursuit with no enemy maneuvering. In both non-static coupling scenarios, the second quadrotor's controller is nearly identical to that presented in equations (3.3) - (3.10), as discussed above. The chase vehicle has a controller onboard that works to minimize the relative error between the two objects, with the mathematical expressions given by (3.25) and (3.4) - (3.6). In other words, the desired translational acceleration controller is modified to calculate desired yaw, pitch and roll angles based on the relative position and velocity error between the two quadrotors. The desired roll, pitch and yaw angles are then fed in as inputs into the second control loop that remains in large part the same as previously discussed. In the simulation cases presented in figures (3.10) and (3.11), the first quadrotor was given an approximate one second lead from its original position in inertial space. This poses an additional consideration when conducting gain tuning as high error magnitudes are likely to result at the launch of the second (chase) vehicle, making system stability more difficult to maintain. The chase vehicle will constantly adjust its desired position and velocity based on feedback from the location of the second quadrotor as well as appropriately compensate for the linkage offset location to ensure an accurate connection. Once the coupling occurs, the system will again switch to a single joint controller to guide the remainder of the flight.

3.3.3 Adaptive Gain Tuning

It was noted that, if relatively aggressive maneuvers were to be desired immediately after the coupling time instance, large errors combined with relatively high gains and altered dynamics may cause an underdamped system. Therefore, for the coupling events shown in figures (3.10) and (3.11), the onboard controller has variable gains that change throughout the course of flight. More specifically, the gains of the combined controller are set to be more moderate in the time instances immediately following the coupling event and then changed to more aggressive values once the system has been stabilized.

This gain scheduling, mentioned above, is crucial for the successful coupling of the two vehicles. Additionally, the gains on the pursuit quadrotor were also dynamically changed throughout the flight scenarios. When the chase vehicle launches, its controller will attempt to minimize the relative error between it and the enemy quadrotor as given by equation (3.25). However, the gain tuning for a stable system can be a relatively difficult task to accomplish considering that large values of $\mathbf{\Delta}$ may occur at the time of initiation. Therefore, the initial flight stage gains are set relatively low, in sacrifice of a slower response time. This methodology is also followed for the cases when the enemy vehicle rapidly changes flight directions, in which the chase quadrotor must make appropriate adjustments to its attitude in order to achieve the newly command acceleration and direction while maintaining stability. Once the flight conditions stabilize into a set direction, the controller gains of the pursuing vehicle ($\mathbf{K}_P^{int}$, $\mathbf{K}_D^{int}$) are increased in order to facilitate a rapid capture. Once the capture occurs, the joint controller assumes responsibility for the position and velocity of the configuration, and a new desired position and velocity are set. Note that the joint controller, as given by equation (3.25), also allows for the system to change yaw angle during flight. Similarly, at the time of capture, the gains of the combined controller are set

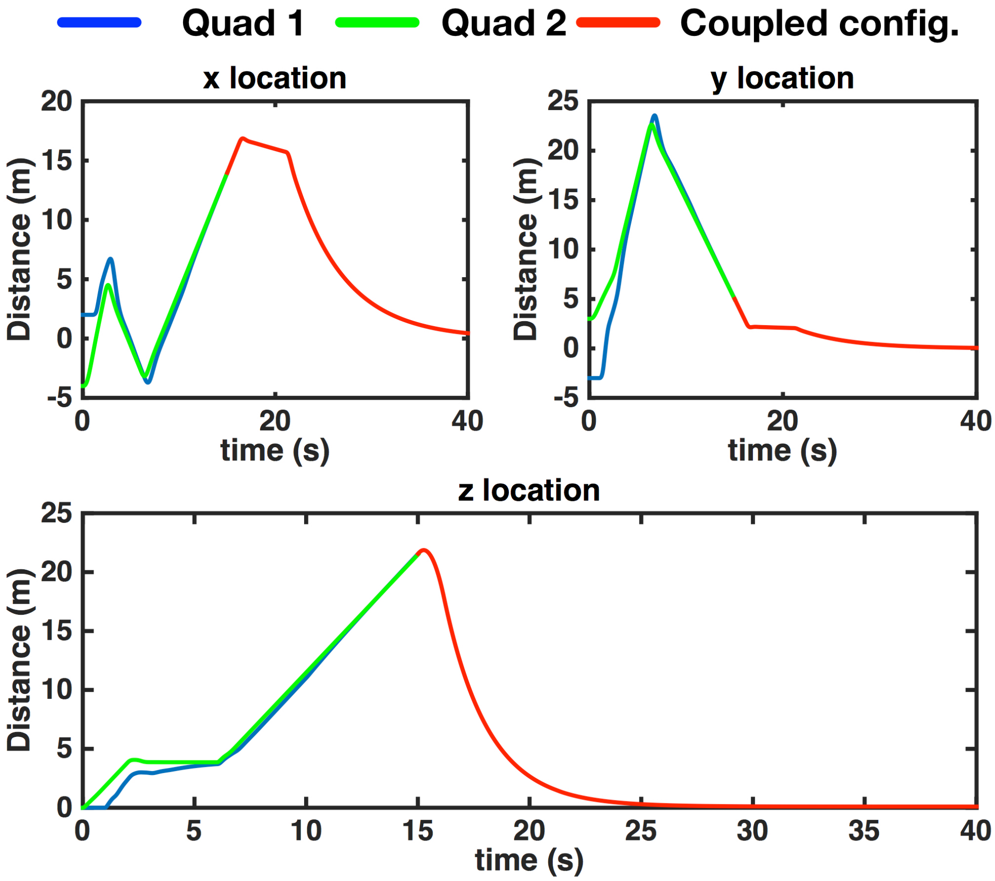

Figure 3.11: Simulation Results Showing Inertial x,y, and z Position for the Non-Static, Target Maneuvering Coupling Scenario.

conservatively in order to stably actuate motion towards the set position. As such, derivative control values on position are set to be high as compared to proportional gain values in order to provide adequate dampening to the system. Once motion is initiated towards the intended target, the proportional and derivative gains are increased to provide a faster response time. The increase in gains event can be clearly seen in figure (3.10) and figure (3.11) and was most notably observed in the non-static coupling situations. This is observed by examining the inertial x coordinate response characteristics in figures (3.10) and (3.11). Here, the initial coupling event occurs followed by a short period of relatively small change. However, when the gains on the joint controller are increased, an immediate change is noted in the rise time of the response. In the case shown in figure (3.11), this change in gains occurs around the twenty second mark.

There was additional consideration in interpreting the simulation results as to consider controller output saturation. It was also noted that the values of the joint controller gains are highly dependent on the orientation of the vehicles at the time of coupling relative to the desired attitude and position set immediately after the event. For the results shown here, the gain values have been specifically tuned to provide adequate system response characteristics for each flight situation.

Chapter 4

EXPERIMENTAL DESIGN

4.1 Quadrotor Hardware and Specifications

The quadrotor used for experimentation is shown in figure (4.1). There are numerous commercial quadrotors available, such as those made by Ascending technologies, Parrot, or 3DR, some of these options are to be discussed, however, for the current application, a custom build was done. This build consists of the components listed in the table (4.1) below, along with associated cost.

Many of the components listed in table (4.1) are relatively generic and can be fitted to virtually any quadrotor of similar size. The sizing of the main quadrotor frame is based on flying area constrains and the desire to fly multiple quadrotors in the same operating space. Given this constraint, the FPV 250 frame, sold and distributed by Hobbyking, offers an acceptable tradeoff between quadrotor size and cost. The motors, propellers and batteries were selected based on lift requirements and data sheet recommendations provided by the frame manufacture. Power is supplied to the autopilot by the APM power module, capable of delivering 3 A of current and transforming the 11.1 V battery potential to 5.0 V for usage of the onboard electronics. An alternative, weight-saving, method of supplying the autopilot with the required power is through the battery elimination circuit (BEC), which is built into the electronic speed controllers. If the BEC is utilized, the autopilot is supplied with 5 V and up to 2 A of current.

Table 4.1: Quadrotor Bill of Materials

Component	**Function**	**Qty**	**Cost**
FPV 250 Frame	Main quadrotor structure	1	\$9.99
Digi International Xbee Series 2	Wireless Telemetry	1	\$22.95
Digi International Xbee Explorer	Wireless Telemetry	1	\$11.89
RCTimer Electric Motors	Propulsion	4	\$11.89
QAV 250 Propellers	Propulsion	4	\$1.85
SimonK 12A ESC	Electronic Speed Control	4	\$9.99
APM 2.6+ Autopilot	Inertial Measurement Unit	1	\$42.99
APM 2.6 Power Module	Voltage/Current Regulation	1	\$9.99
3 Cell, 11.1V 25C Lipo Battery	Power Supply	1	\$16.99
Miscellaneous	Vibration Dampening/Wiring	1	\$5.99
Total			\$170.00

The motors are RCTimer 1806 - 2300 KV electric motors, capable of delivering approximately 400 grams of thrust each while consuming around 96 Watts of power. The motors are connected to the electronic speed controllers (ESC), which are provided power directly from the attached battery.   Apart from the components discussed above, two of the components listed in table (4.1) serve a vital role in the quadrotors operation, namely the autopilot and the wireless telemetry radios. In this case, the autopilot is the APM 2.6+, sold and distributed by 3DR robotics. The APM 2.6+ is a versatile, open source, arduino/C++ compatible autopilot that can be easily interfaced with a number of software programs. It features an Atmel ATMEGA 2560 8 bit, 16 MHz processor with an Atmel ATMEGA 32U-2 for additional processing and universal serial bus (USB) control. In addition to the two processors onboard, the APM module feature an Invensence MPU-6000, 6 degree of freedom (DOF) accelerometer and gyroscope (Inertial Measurement Unit) with integrated digital motion processor (DMP) for IMU sensor fusion and processing. The hardware also includes a separate, barometric pressure sensor for altitude determination. It should be noted that the onboard IMU is specified as 6 degree of freedom, due to to omission of an onboard 3 DOF magnetometer. The dearth of a magnetometer onboard poses no immediate concern, and is done to optimize the hardware for quadrotors, considering their substantial magnetic disturbance field due to high motor currents. Therefore, the creators of the APM 2.6+ autopilot designed the hardware to easily adapt an external global positioning system (GPS)/Compass peripheral unit via an I2C port. Out of the box, the APM 2.6+ is programmed with the available open-source firmware through one of many ground stations. In this case, the ground station utilized is Mission Planner, however other options exist such as qgroundcontrol or APM planner 2. The programing of the standard firmware was only done to verify the functionality of system components as custom routines were later developed so as to avoid any programming limitation placed on the system. One such limitation, that is to be discussed in a later section, is the use of the MavLink communication protocol. Ad-

Table 4.2: Main Quadrotor Components

Number	Component	Function
1	IR Reflectors	Tracking for Motion Capture System
2	Xbee Radio Module	Wireless Communication
3	APM 2.6+	Two-stage LL control & IMU
4	Power Module	Voltage Regulation from 11.1 V to 5.0 V

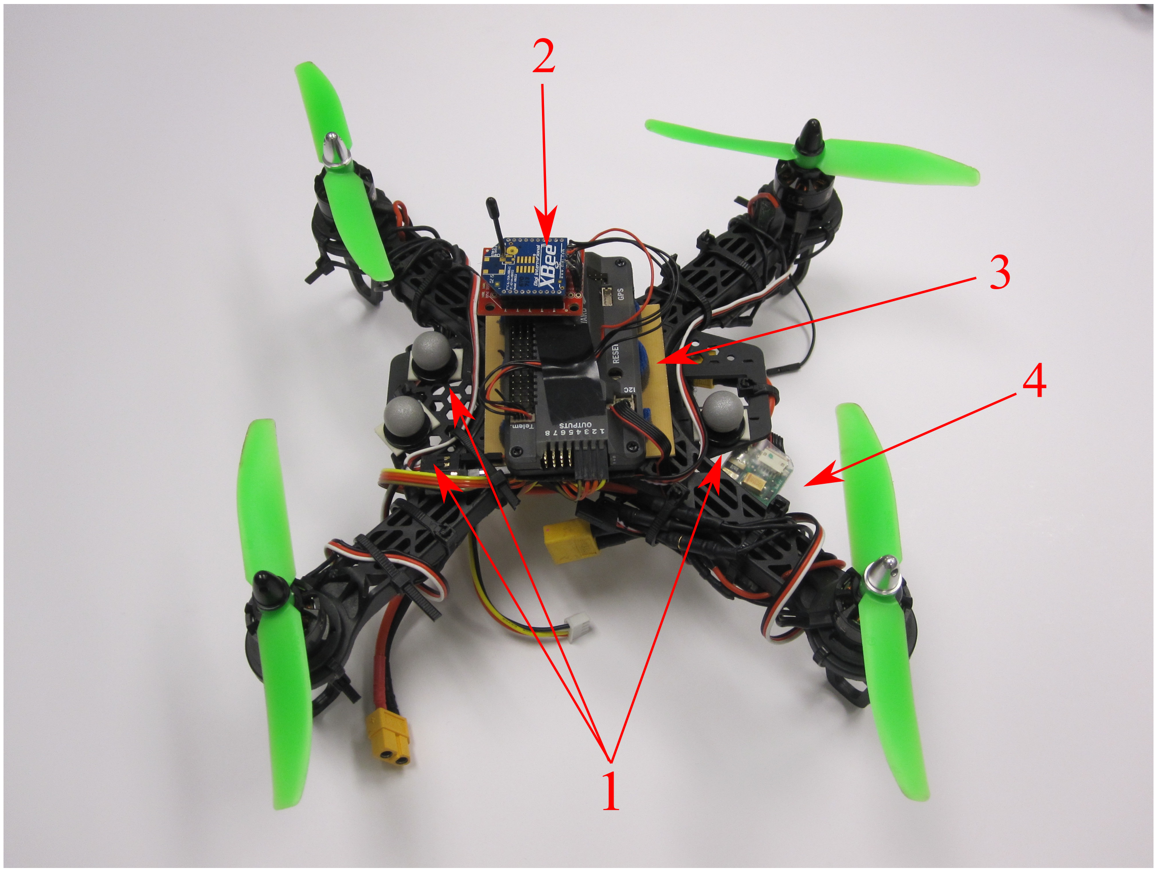

Figure 4.1: Quadrotor Hardware.

ditionally, the autopilot's arduino/C++ compatibility allows for an open workspace were specialized programs can be developed and tested. In addition to the autopilot are the Digi International Xbee series 2 radio modules. The Xbee series 2 comes from a product line consisting of very similar, yet vastly diverse series of radio transmitters and receivers. In this case, the Xbee series 2 was selected because of their ability to be set to send and receive serial data in a variety of configurations. Specifically, these modules operate on a frequency of 2.4 GHz and utilize the Zigbee communication protocol, which is derived from the IEEE 802.15.4 standard. The series two modules used in this application consume approximately 2 mW of power on transmission and have an operating range of around 400 feet. Unique to the series 2 is the ability to construct distributed networks and two main operational modes: node to node or broadcast. As will be discussed, the current setup has called for the series 2 in node to node configuration, however this can be potentially altered depending on the intended application. The location of these components can be seen in figure (4.1), with the labeling described in table (4.2).

Table 4.3: Quadrotor Specifications vs. Commercially Available Options.

	HORC Quad.	**Firefly**	**Hummingbird**
mass (g)	520	1000	510
dimensions (m)	0.26 x 0.26 x 0.11	0.6 x 0.66 x 0.16	0.54 x 0.54 x 0.08
max payload (g)	350 (est.)	600	200
endurance (min)	7 (est.)	12	20

4.2 Control Architecture and Flow Chart

In the sections that follow, the complete control architecture used to accomplish autonomous control of a, or multiple, quadrotors is described. The system can be divided up into two major subcomponents. These subcomponents are referred to as the High-Level (HL) controller, which runs on the ground station computer, and the Low-Level (LL) controller that is executed onboard the autonomous vehicle. A high level flow chart of the entire system is shown in figure (4.2). In this figure, the ground station and LL controllers are separated by the wireless telemetry link, which is to be discussed in more detail in a later section.

4.2.1 The High Level (HL) Controller & The Vicon Motion Capture System

In the present case, the HL controller is implemented with the use of MATLAB computational tool. The role of this controller is to (1) Connect to the Vicon motion

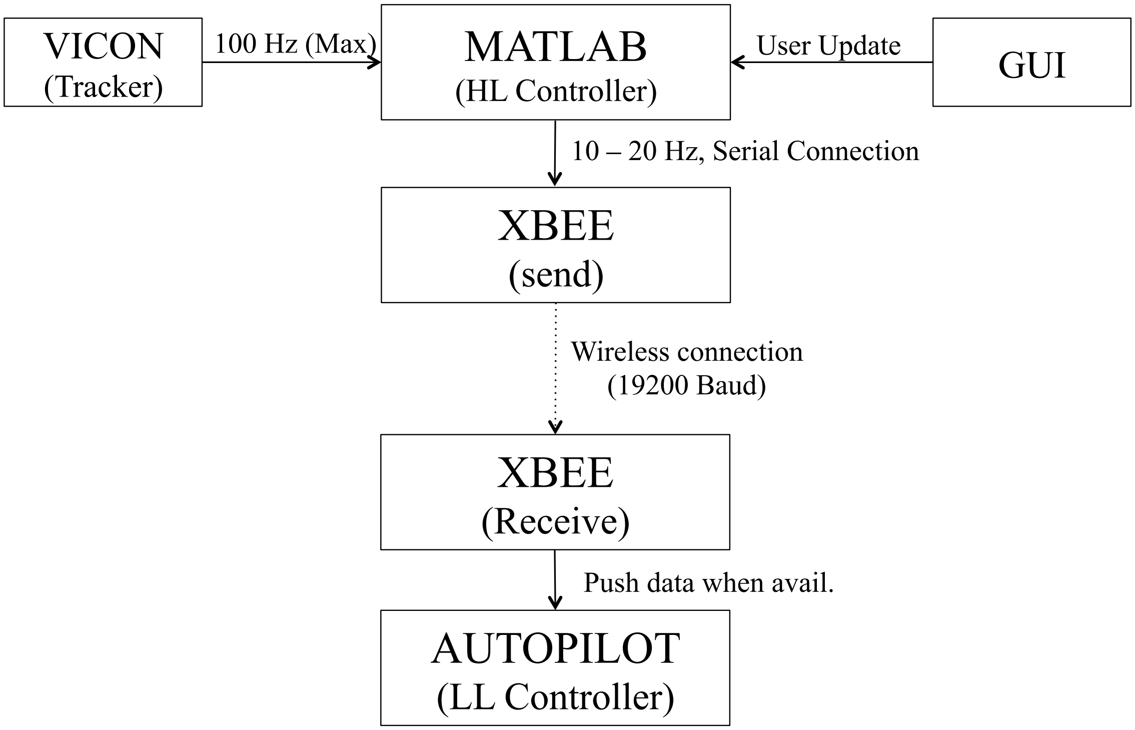

Figure 4.2: High Level Flow Chart of System Architecture.

capture system to obtain real-time position and velocity data, (2) run the developed graphical user interface (GUI) to obtain user-desired positions and yaw, (3) calculate position and velocity errors, and based on this information, compute desired roll, pitch and yaw angles as well as desired thrust, and (4) wirelessly transmit desired set points to the autopilot onboard the vehicle. An overall flow chart is shown in figure 4.3.

In order to obtain consistent and reliable position and velocity data, the flying area is mapped by a total of four Vicon Bonita 3 NIR cameras. These cameras are, in turn, connected with ethernet cables to a remote computer running the Vicon Tracker software.

Prior to experimentation, these cameras are calibrated and a user-defined coordinate system is created. The resulting coordinate frame, used by the Tracker software, serves as the inertial frame for the dynamic system.

The developed MATLAB ground station program retrieves position and velocity data from the Tracker software by utilizing part of Vicon's software development kit (SDK). This program toolkit includes a variety of MATLAB code designed to obtain Vicon data at a rate up to 100 Hz. In this case, parts of this toolkit where used to develop new functions that could then be implemented in a real-time loop and deliver the user requested data. Here, this information consists of the quadrotor position and current yaw angle. MATLAB connects to this software by defining a client connection through the wireless network and known IP address of the remote connection.

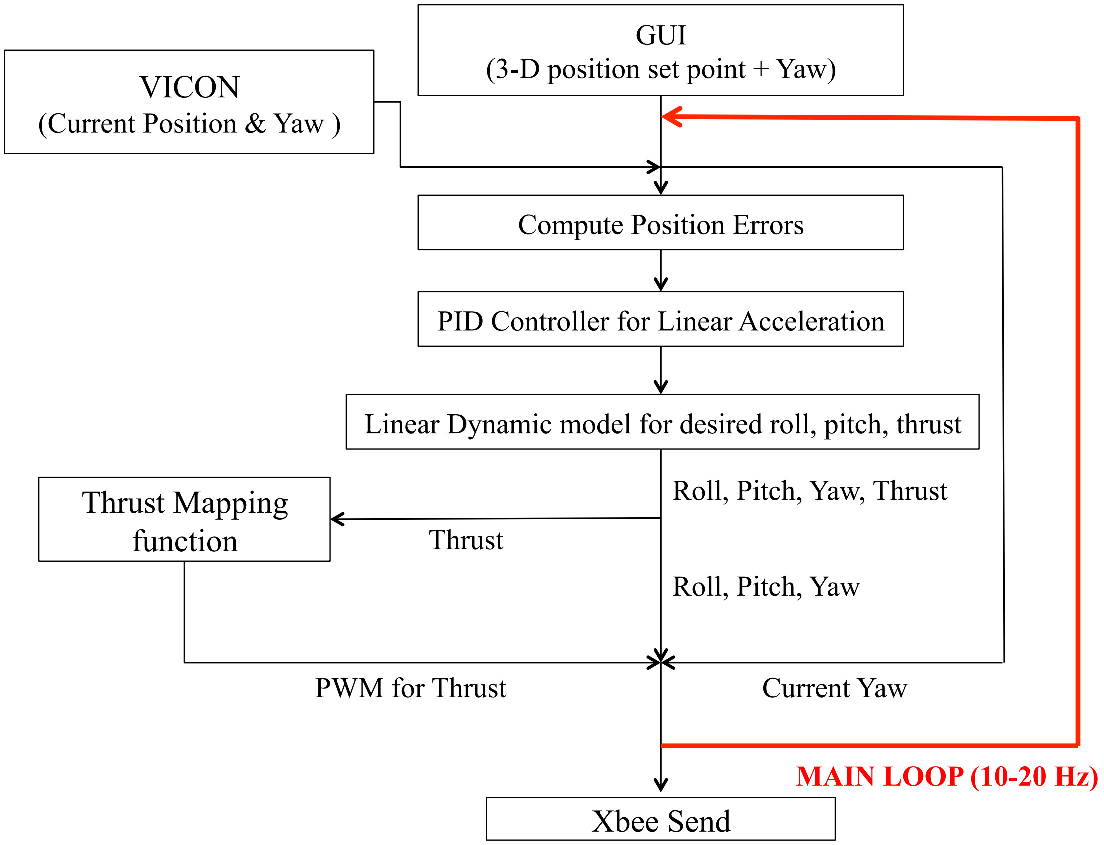

Figure 4.3: Detailed Flow Chart of the High Level Controller.

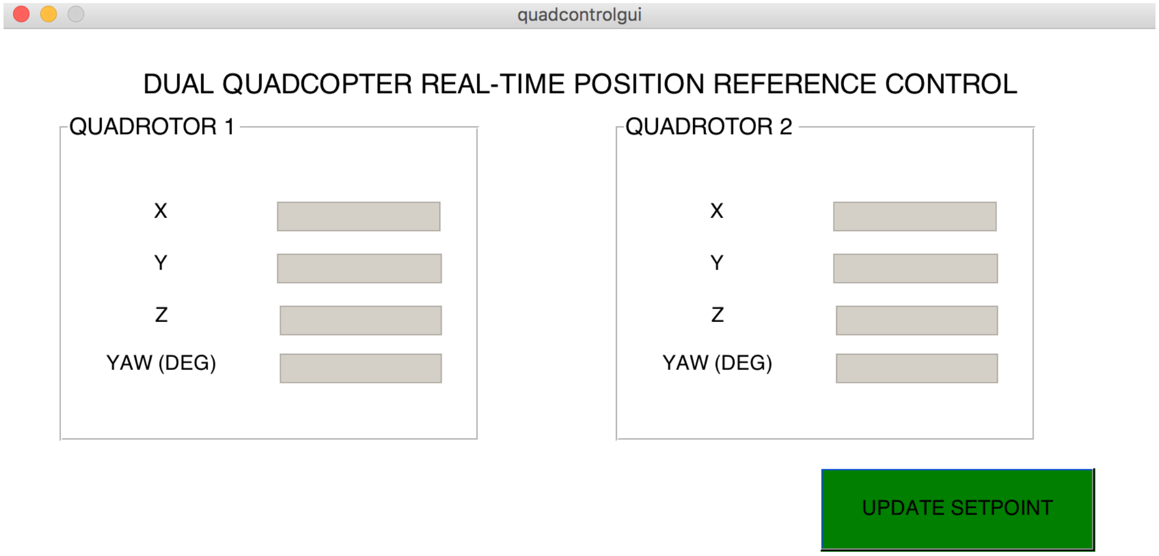

Figure 4.4: Graphical User Interface for Position Set Points.

In addition to obtaining real-time position data from the motion capture system, the developed ground station features a graphical user interface (GUI) which allows users to set desired position set points for one or multiple vehicles. This GUI, shown for the two quadrotor case, is displayed in figure (4.4). The GUI depicted in figure (4.4) is a general input method, and can be replaced by another source (for example, joysticks, etc.). The position reference values are updated once the user input is finished and "update set points" is selected.

Once required feedback information is retrieved from the motion capture system, the position and velocity errors are calculated and used in the high-level proportional derivative (PD) controller. The output of this first PD controller is that of desired translational acceleration in the inertial space. This is shown in equation (4.1).

$$\ddot{\mathbf{r}}_{\mathbf{des}} = K_p \mathbf{e}_{loc} + K_d \mathbf{e}_{vel} \tag{4.1}$$

In the above equation, K_p and K_d are diagonal matrices containing the values of proportional and derivative gains in x,y,z, respectively, $\ddot{\mathbf{r}}_{\mathbf{des}}$ are the desired accelerations in x,y,z, and $\mathbf{e}_{loc}$, $\mathbf{e}_{vel}$ are 3x1 vectors containing the errors in inertial position and velocity of the vehicle. The desired translational acceleration set by the PD controller in equation (4.1) are then used in a linear dynamic model derived from equation (2.2). This linear model assumes that the roll and pitch angles remain relatively small and that for each computation the yaw angles is assumed to be constant. As such, the current yaw angle of the vehicle must be known, and is found through the use of the motion capture system discussed in a previous section. Upon linearization, the resulting desired values for roll, pitch, yaw and thrust can be calculated through the following relations Mellinger (2012).

$$\theta_{des} = \frac{1}{g}(\ddot{r}_{x,des}\cos\psi + \ddot{r}_{y,des}\sin\psi) \tag{4.2}$$

$$\phi_{des} = \frac{1}{g}(\ddot{r}_{x,des}\cos\psi - \ddot{r}_{y,des}\sin\psi) \tag{4.3}$$

$$T_{des} = m(g + \ddot{r}_{z,des}) \tag{4.4}$$

In the above equations, θ_{des}, ϕ_{des} and T_{des} are the desired pitch and roll angles as well as the desired thrust, $\ddot{r}_{i,des}$ is the desired inertial acceleration in direction i, as provided by the controller in equation (4.1), ψ is the current yaw angle, and m, g represent the quadrotor mass and gravitational acceleration, respectively. The resulting pitch and roll angles are subject to user specified constraints as to limit the extremes of these angles prior to sending the set points to the onboard autopilot. Additionally, these angles are passed through the rotation described in the following paragraph before they can be relayed to the air vehicle. Also, as shown in the block diagram representation of the high level controller provided in figure (4.3), the resulting value of thrust from equation (4.4) is sent to a thrust mapping function prior to being "packaged" along with roll, pitch and yaw angles to be sent to the quadrotor via the serial protocol. In this case, it was determined that, instead of sending thrust values in units of Newtons, a pulse width would be sent instead. This gives rise to the necessity of this mapping function, as its role is to take thrust values (in Newtons) given by equation (4.4) and convert to a range suitable pulse widths between approximately 1100 and 2000 micro seconds in length. However, including this mapping function adds some level of complexity to the controller considering that there are now two parameters influencing the vertical controller dynamics of the system. These two parameters are (1) the input-output mapping range of the thrust function and (2) the controller gains in equation (4.1). As such, the thrust mapping function was adjusted with unit gains in equation (4.1) to the hover condition. That is to say, with unit gains in the z direction of the high level controller, the thrust mapping function was tuned so that the quadrotor would hover at fixed altitude. Following this process ensures that a reasonable input-output mapping range is established, and that the system response dynamics can then be fine tuned by employing non-unity gains in equation (4.1).

It is important to note that the onboard inertial measurement unit (IMU) does not utilize the same coordinate frame as the body reference provided in figure (2.1). Therefore, a coordinate transformation must be performed prior to sending the desired set points to the quadrotor in order to ensure system convergence and stability. Both the body and IMU coordinate systems are so-called “right-handed” frames, however the direction of the second and third axis differ. The inertial measurement unit with its associated coordinate frame is illustrated in figure (4.5).

While subtle, these differences poses an additional consideration for practical implementation. Accounting for this dissimilarity was done by finding the rotation matrix between the two systems to express quantities calculated in the body frame in the IMU frame. This rotation matrix is shown in equation (4.5).

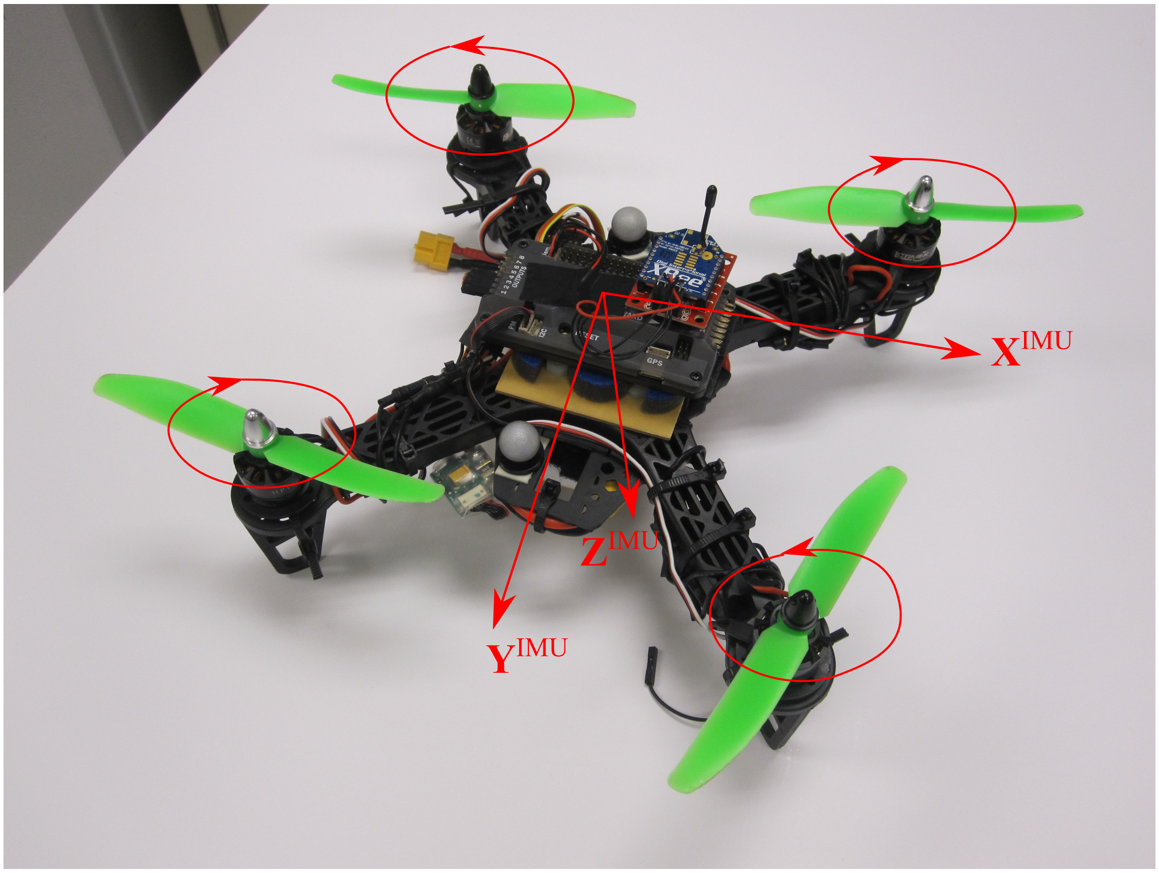

Figure 4.5: Coordinate System Utilized by Inertial Measurement Unit.

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}^{IMU} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}^{b} \tag{4.5}$$

It is important to note that equation (4.5) is used solely to account for the differences in the second and third axes of the two coordinate systems (body and IMU) - the body frame roll, pitch and yaw angles, shown on the right side of equation (4.5), are found through (reference equation here), which arrises from a linearization of the three translational equations of motion. These three equations, in turn, contain a rotation from the body coordinate system to the inertial frame through equation (2.1), and accounts for the current yaw angle for computational purposes.

4.2.2 Wireless Communication Protocol

In order to relay the desired roll, pitch, yaw and thrust angles to the flight vehicle, the utilization of a wireless telemetry link is necessary. The proposed telemetry link has a number of existing solutions, most of which are developed by ETH Zurich and the Pixhawk flight controller development team. The most common of such wireless communication links utilize the MavLink communication protocol in conjunction with radio frequency transmitters and receivers. The MavLink library is very diverse and includes built-in functions to control up to 255 vehicles and stream off-board set points in a number of commonly used forms. These forms include streaming desired roll, pitch, and yaw angles or desired roll, pitch, yaw rates as well as including select safety features. While MavLink offers a plethora of built-in functions and operations, it is overly complex for the current implementation and is not modifiable if more or other information were to be desired to be transmitted/received by the quadrotor. Additionally, if the MavLink protocol were to be used, a compatible ground station is needed to relay the MavLink messages from the off-board computing location to the air vehicle(s). This adds an additional level of complexity and software integration/validation required prior to successful flight. While it is possible to package and send serial MavLink messages via MATLAB, it was determined that a serial protocol for this application would be developed.

This wireless link has a number of vitally important parameters, with the main considerations given to (1) the protocol baud rate, (2) means of transmission and (3) protocol type. Independent of the system frequency is the serial protocol baud rate, representing the bits per second of data that is transmitted and received, or rather how quickly data is sent and received by the two radio modules. The baud rate for wireless communication must be adjusted so that satisfactory transmit and receive frequencies are achieved while packet loss is eliminated. For this reason, the process of selecting and implementing a given baud rate is coupled with the loop frequency of the the high level controller, which has been previously discussed. It was found that a baud rate of 19200 bits per second resulted in satisfactory serial communication robustness for high level controller loop frequencies in the range of 10 - 20 Hz. It should also be noted that the baud rate and robustness of the serial protocol was found to be heavily dependent on the amount of information, or bits, of data to be transmitted and received. Some of these complications will be discussed later in this section.

The second main consideration in the development of the wireless telemetry link is the means of transmission. Currently, as shown in figure (4.1) and table (4.2), a pair of Digi International Xbee series two radio modules are used for node to node transmission and receiving of serial data. These modules are, in turn, connected to an Xbee adapter to allow the pinout of the Xbee to be converted into another format. For connection to the autopilot, an Xbee to Future Technology Devices International (FTDI) adapter was used, while connection to the MATLAB high level controller was done with a Xbee to Universal Serial Bus (USB) converter. These radio modules can then be configured in X-CTU, a software program developed by Digi International. The Xbee series two modules are known to be diverse and feature a number of operating modes, including the ability to set up mesh networks. These radio transmitters/receivers can be configured in two main modes: broadcast and node to node. In the present configuration, the modules are set to transmit in node to node as opposed to broadcast to all Xbee modules with the same network identification number. This was found to work more efficiently by experimentation, as when operating in broadcast mode, the Digi International firmware layer introduces a considerable latency in data transmission at high baud rates and frequencies. This latency was measured to be variable, but generally on the order of approximately 1 second and would cause considerable stability issues due to the controller bandwidth required to stabilize the quadrotor. It was found that operating the modules in the node to node configuration eliminates this latency, but constrains a pair of Xbee's to communicate only with its node partner. In addition the above stated features, the radio modules can also be configured to include flow control. In serial communication, flow control can be added which allows two end points to communicate not only the serial data stream, but to also include flags that inform the system when to send the information so as to avoid packet loss and thereby allowing an additional level of logic. Flow control was not utilized in the current application as it was deemed to add an additional level of complexity as well as be of no benefit as experimental tests show that bandwidth and/or packet loss is not a current system constraint.

The final parameter that was considered is the type of data that is transmitted via the wireless link. While it is true that all data be ultimately binary, there exists two main methods of data packaging that were explored. These methods were (1) sending ASCII characters and (2) develop a method to send numbers directly as binary. The first method provides a plethora of additional characters that make the process of developing a start and end sequence identifier of a package that is relatively straight forward to identify in the read routine onboard the vehicle. However, it has the major drawback that each number and/or sign sent would require an entire byte of data to be transmitted and received, meaning that the serial communication must operate at a higher baud rate to achieve higher speeds, increasing the risk of packet loss. Additionally, sending ASCII characters requires a more advanced conversion to take place after the read routine has been executed. This is because all standard ASCII characters are represented by decimal numbers 0 to 127, with ASCII numbers 0 through 9 represented by decimal 48 through 57. This means that all numbers received by the autopilot are biased by 48, requiring an additional conversion to be done before the information can be used as the desired set points. In an effort to increase the system bandwidth, a second, revised, protocol was developed to send data directly as binary numbers. While this makes it possible to send more data with less bytes, the process of finding a unique end or start sequence is more complicated. As such, the set points of roll and pitch were defined to be 8-bit signed integers as they are limited to the domain -20 to +20 degrees, while thrust and yaw are defined as 16-bit unsigned and signed integers, respectively. This significantly reduces the amount of data to be sent, considering that a string of values for roll, pitch, yaw and thrust can be sent in six bytes. As in the case if ASCII information were to be sent, a read routine was necessary on the receiving side. However, this routine was greatly simplified when sending binary data and consisted mostly of shifting most and least significant bits (MSB/LSB). In addition to sending the desired roll, pitch, yaw and thrust angles, the actual yaw was also relayed through the quad wirelessly, which was done for reasons to be discussed in the following section. Finally, in order to identify the end of the packet and to provide a failsafe, an eight bit end sequence of hexadecimal 0xFF (255 decimal) is sent. This allows the autopilot to check the completeness of the received information, and discard corrupted or incomplete data.

4.2.3 The Low-Level Controller

Once the serial data is sent via the wireless telemetry link from MATLAB, it is received by the autopilot for use in the two-stage low level controller. This controller has two main functions that are critical to the operation of the quadrotor, they are: (1) Stabilize the quadrotor about a given set point(s) and (2) provide adequate dynamic transitional performance when setting new roll, pitch and yaw angles. This process is achieved through the use of two PID-type controllers that consider both

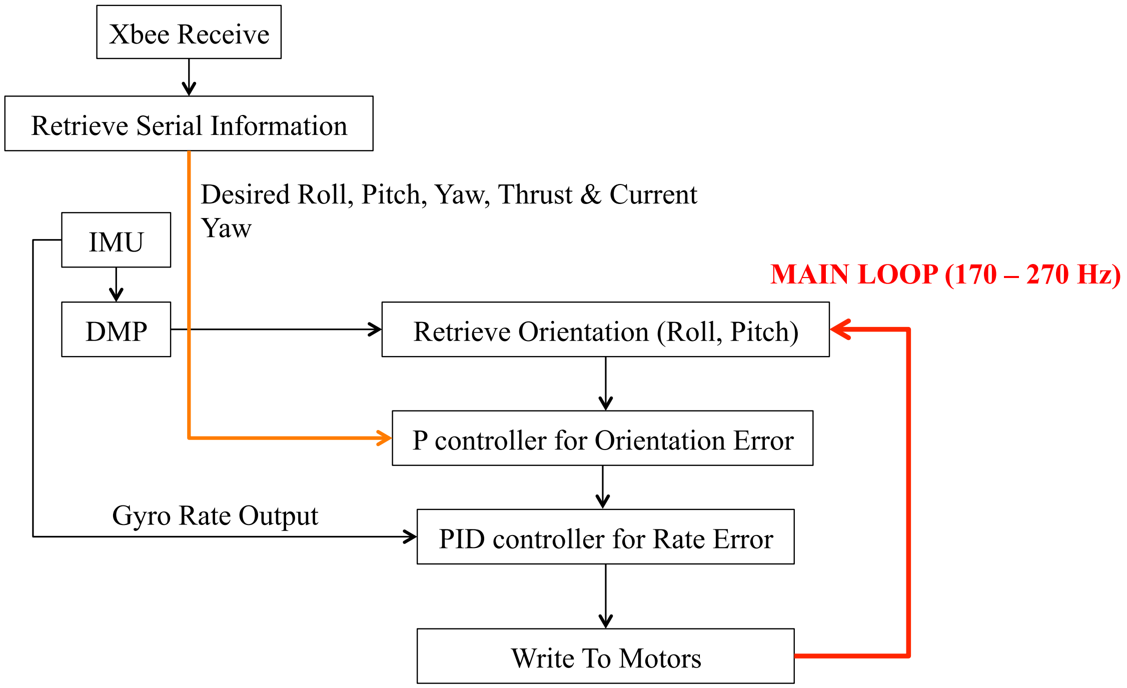
Xbee Receive
Retrieve Serial Information
Desired Roll, Pitch, Yaw, Thrust & Current Yaw
IMU
DMP
MAIN LOOP (170 – 270 Hz)
Retrieve Orientation (Roll, Pitch)
P controller for Orientation Error
Gyro Rate Output
PID controller for Rate Error
Write To Motors

Figure 4.6: Quadrotor Low-Level/Autopilot Controller Flow Chart.

the angular and angular rate errors. A detailed flow chart of the low level controller operation is provided in figure (4.6). The first level of control consists of a proportional controller which computes the desired roll, pitch and yaw rates based on angular errors. The reference values for this controller are provided by the serial telemetry stream every 10 - 20 Hz, with new reference conditions set as they are received where as the current roll and pitch angles are retrieved from the IMU and associated digital motion processor (DMP). In this case, and as discussed in a previous section, the digital motion processor is an additional component apart from the IMU which performs the necessary sensor fusion for delivery of current roll, pitch and yaw angles to the main autopilot. This has the advantage of reducing the computational burden required by the main board, enabling the system to perform the stabilization routines at higher bandwidths. The outputs of this first level P-controller are constrained so as to ensure reasonable angular rates are commanded even when this controller output saturates.

The desired angular rates for roll, pitch and yaw that are computed by the first P-controller are then fed into a secondary PI controller. This controller receives desired angular rates and samples the onboard rate gyro to obtain current rate information. The outputs of the PI controller are desired PWM values to be sent to each motor. It should be stated that all gain values were tuned through experimental testing, utilizing manually operated flight modes to test the dynamic stability and response of the vehicle.

The two PID-type controllers discussed above function well for a quadrotor when there is no sensor drift or bias. However, similar to Mahony *et al.* (2012), the output of the inertial measurement unit output can be generically modeled as:

$$\theta_{IMU} = \theta_{act} + \eta_\theta + \Gamma_\theta \tag{4.6}$$

$$\phi_{IMU} = \phi_{act} + \eta_\phi + \Gamma_\phi \tag{4.7}$$

$$\psi_{IMU} = \psi_{act} + \eta_\psi + \Gamma_\psi \tag{4.8}$$

In the above equations, θ_{IMU}, ϕ_{IMU}, ψ_{IMU} represent the IMU output for the pitch, roll and yaw angles, respectively, θ_{act}, ϕ_{act}, ψ_{act} are the true values of roll, pitch and yaw, η_i represents the bias, or drift, in variable i and Γ_i are the additive, random, noise terms in each measurement. Considering that the noise term is random and unknown, we rely on the digital motion processor to filter and fuse sensor data and to attenuate the effect of Γ_i. However, while the digital motion processor is capable of the required filtering functions, it does not account for potential offsets, or drifts, in the gyro or accelerometer. Various static tests were performed on verified flat surfaces, printing digital motion processor outputs to a serial read screen in order to determine the magnitude of potential offsets. From these experiments, it could be seen that a bias of ±0.5 to ±5 degrees was relatively consistent and constant. Although any potential bias was noted to be constant, it was also observed that the offset would not stabilize until approximately 30 - 40 seconds after the inertial measurement unit is initiated in the startup routine.

An inertial measurement unit bias of ±0.5 would be acceptable to operate the system normally, however, a much larger offset is very likely to be detrimental to the stabilization of quadrotor. Therefore, in order to mitigate the effect of any offset, and to make the startup process as general as possible, a subroutine was created to sample the IMU output and obtain roll and pitch angles some time after initial startup. Currently, the IMU is sampled for these offsets approximately 40 seconds after the initialization. This was offset was found through experimental testing, and is assumed to be constant throughout the flight, similar to the process discussed in Mahony *et al.* (2012). For this reason, the quadrotor is placed on a known flat surface until a LED indicator light is switched off, indicating that the sampling of the offsets is completed and the vehicle is ready for flight.

Finally, it should be noted that, due to having a 6-DOF inertial measurement unit without an onboard magnetometer, the determination of the yaw onboard the flight vehicle is made considerably more difficult. This is because the determination of roll, pitch and yaw angles is done through sensor fusion - a process of combining two or more sensory source information and creating one, less noisy high fidelity measurement. Roll and pitch measurements are found through fusion of the onboard rate gyro and accelerometer with the DMP. The same process is used to achieve the current yaw angle, however, for most quadrotor attitudes, the gravitational vector is acting perpendicular to the yaw plane, meaning that the yaw angle measurement must come almost solely from integration of the rate gyro. This has the inherent drawback of measurement drift, and, in this case, is relatively severe. Therefore, in order for the quadrotor to autonomously maintain a non-spinning condition, a reliable yaw angle measurement must be provided by some source. Additionally, it should be noted that the current yaw angle is required by the high level controller, as shown in equations (4.2) - (4.3). Therefore, this complication can be resolved in one of two ways: (1) purchase a external 3-DOF magnetometer and integrate via an I2C port or (2) obtain the current yaw angle from the Vicon motion capture system. Both methods were explored, however it was ultimately decided that using the Vicon system to determine the yaw angle was the more suitable choice. This was done mainly with the consideration that the high level and low level controllers operate at different frequencies, and that obtaining the current yaw angle from IMU measurements onboard the vehicle would require the serial stream to operate in two directions (i.e the quadrotor radio module must both transmit *and* receive). Therefore, to simplify the overall control loop, the current yaw angle is sampled at the high level controller frequency (10 - 20 Hz) and sent to the quadrotor in the serial stream. The quadrotor, obtaining new roll, pitch and yaw set points at regular intervals will stabilize the vehicle around the last requested points at a rate of 170 - 270 Hz.

Figure 4.7: HORC Quadrotor During System Robustness Testing.

4.3 Quadrotor Response Data and Characteristics

Considering the system described in the previous section, it is now desired to evaluate the response characteristics of the autonomous system. In order to accomplish this, a number experiments were performed. Specifically, tests were conducted with one, as well as two, flying vehicles, in flight scenarios ranging from single, point-to-point, position control to position control with external perturbations in order to test the robustness of the system. To accomplish this task, the high level controller is modified as to record both the position data obtained from the Vicon motion capture system as well as store the desired set points that are provided by the user. Results from a test using a single quadrotor utilizing point-to-point motion control is shown

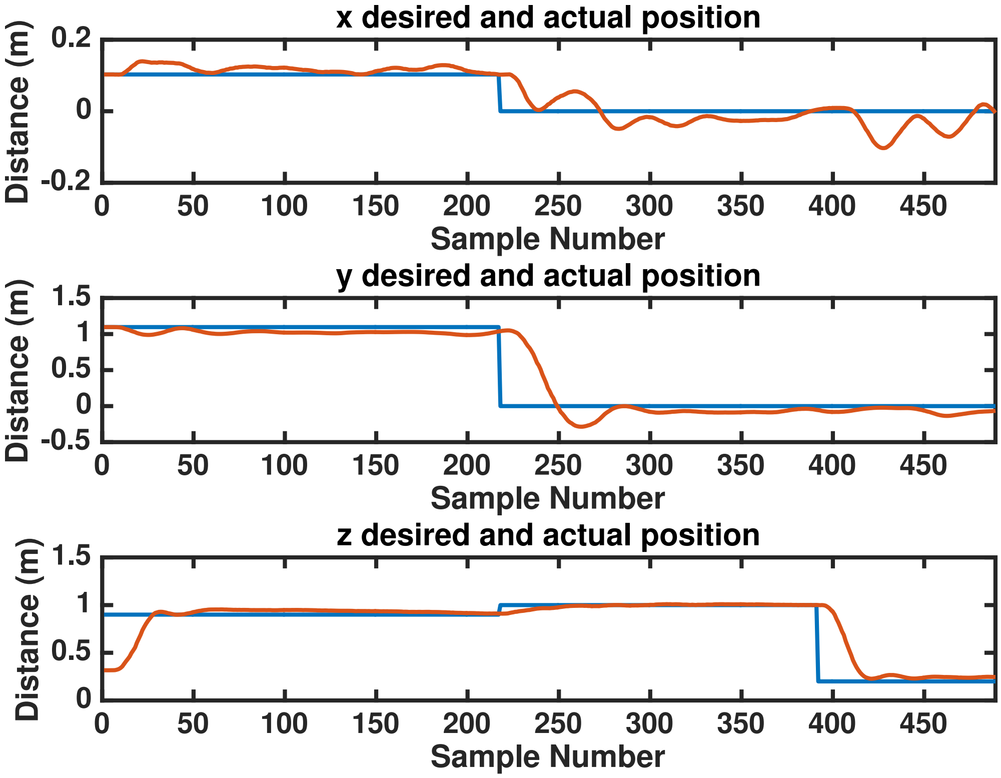

Figure 4.8: Single Quadrotor Point-to-point Motion Control Response.

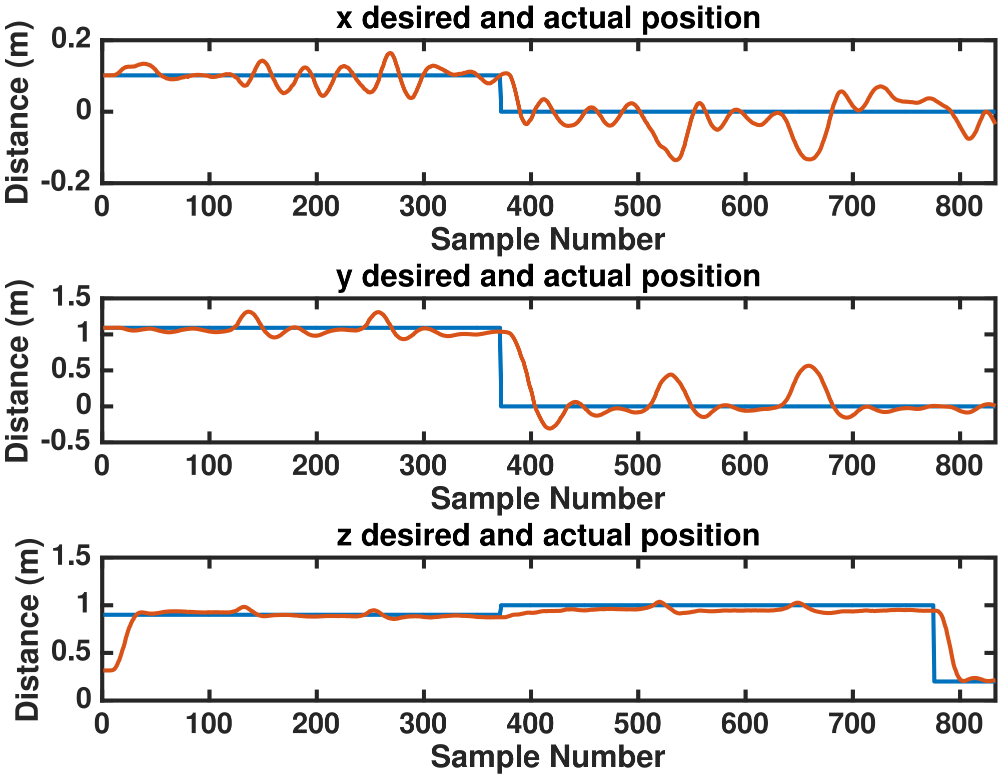

Figure 4.9: Single Quadrotor Point-to-point Motion Control with Disturbance Response.

in figure (4.8).

Results shown in figure (4.8) show the position tracking ability of the closed-loop system. It is noted that the system in slightly underdamped in the inertial x and y directions considering the observed small-scale steady state oscillations and overshoot when commanding a new desired inertial position. It is believed that some aspects of this non-favorable response be attenuated by additional gain tuning. While additional gain tuning may be performed in order to reduce the system overshoot to step inputs and dampen steady state oscillations, it is also vital that a suitable stability margin be maintained as to guarantee closed-loop system response convergence. In addition to the experimentation discussed above, perturbance testing was done in order to assess

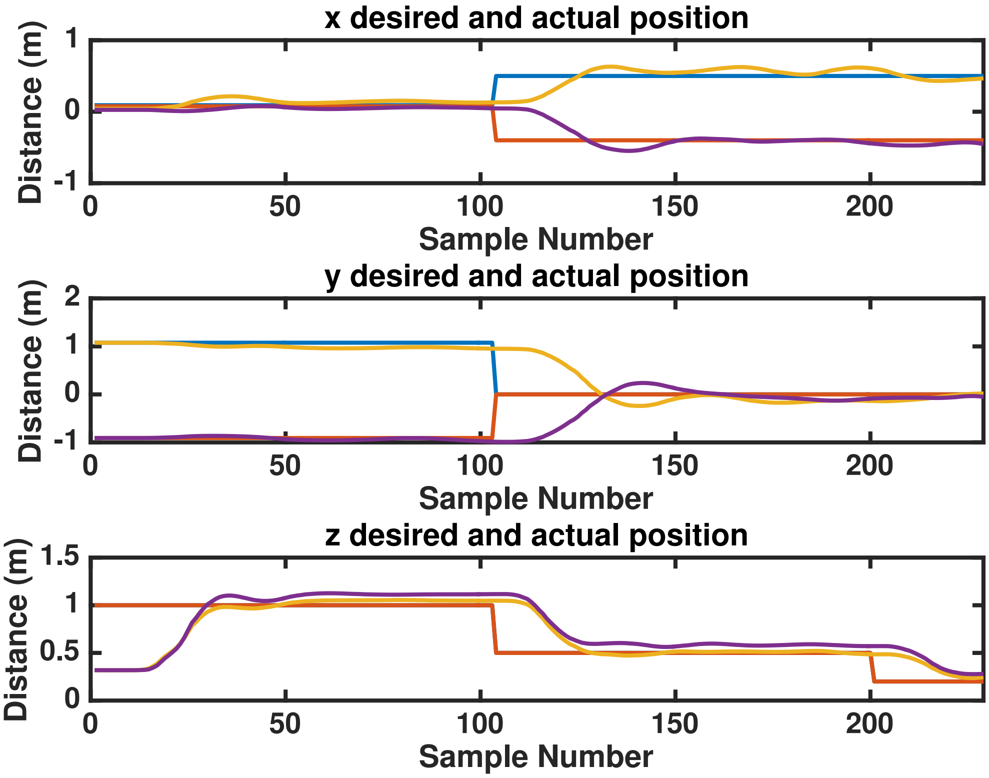

Figure 4.10: Dual Quadrotor Point-to-point Motion Control Response.

system stability, robustness and response characteristics to external disturbances. This is especially pertinent to the current system considering the proposed coupling mechanism and method will generate an external disturbance to the system. In this phase of system testing, a single quadrotor was commanded to hover in a desired, user-provided, position in inertial space, then being "pushed" in the inertial y direction. Recorded data is presented in figure (4.9).

In figure (4.9), the application of the external disturbance is clearly seen in the collected inertial y position data. Namely, the presence of the two peaks in the data set show the time instances in which the quadrotor was involuntarily moved from its current, desired position. It should be noted that the disturbance is not solely applied in the inertial y direction, but also contains a component in the inertial x

and z directions, as can be seen in the results presented. Once two disturbances are applied to the vehicle, it is autonomously moved to another position where the process is repeated.

Considering the robustness of the system shown in the results presented above, it is now desired to consider the case of addition an additional quadrotor to the system. In the new configuration, a slight, but noticeable latency is observed requiring retuning of the closed-loop gains considering discrete time system dependency on the sampling frequency. Results from an experiment of point-to-point autonomous motion control of two flight vehicles is shown in figure (4.10). The data shown in figure (4.10) show the individual position tracking ability and response characteristics for the system with two agents.

Chapter 5

FUTURE WORK & CONCLUSIONS

The analysis and simulation of a coupled quadrotor system opens new frontiers and the possibility of quadrotor rescue or enemy quadrotor chase and capture. This thesis presents the modeling and dynamics of a coupled quadrotor system as well as a simulation displaying various flight situations. Simulation results show the feasibility of this concept and the applicability of PID control with adaptive gain tuning to be used in a variety of flight situations. Gain tuning was performed to account for system configuration and orientations to ensure stability and satisfactory response characteristics. Additionally, an experimental setup was introduced, and a discussion regarding possible difficulties that arise when working with real-world systems.

Future extensions of this work include a multitude of improvements to the current model. One such advancement would be the development of a coupling system capable of automatic detachment. Such mechanism would directly improve the multi-agent cooperation task as it would enable vehicles to dynamically couple and decouple to perform a variety of tasks. However, the draw back to such configuration is that the coupling mechanism itself would not be static, but rather contain a number of sensors and actuators which introduces additional complexities to the system. Furthermore, such mechanical device will likely be power consuming, which is detrimental to the flight vehicles endurance time. A feasible solution will likely encompass the use of a ball and socket joint or implementation of electromagnets. Another potential solution for a "coupling" mechanism suitable for enemy pursuit and capture is that of a free-swinging net, which can be modeled as a three dimensional pendulum, as can be found in Chaturdevdi *et al.* (2011).

In terms of control system logic, it is necessary to add collision-avoidance algorithms to ensure that the enemy capture scenarios be entirely physically possible. Additionally, exploration of optimum control algorithms in order to perform routine, capture or pursuit tasks would likely benefit the system subject to the cost function selected. Other considerations include implementation of new non-linear control laws in order to avoid adaptive gain tuning/scheduling and to develop and obtain a single controller for the entire flight envelope. Multiple such algorithms can be implemented and tested, particularly for cases when the quadrotor must aggressively maneuver, such as enemy capture. Work is currently being done to develop an onboard sensing platform, utilizing a laser scanner similar to that used by Sa and Corke (2011), to track other quadrotors. The outcome of this research will contribute to a feedback source where the control algorithm will be able to identify, track, intercept and remove potentially malicious quadrotors. Other algorithms may be developed in order for a quadrotor or ground vehicle to serve as a mobile charging station as part of a heterogeneous robot swarm.

In all, it has been found through simulation that the concept of coupled quadrotors for both cooperative tasks and enemy pursuit and capture is feasible subject to a set of system constrains. The dynamic modeling of a single quadrotor as well as the coupled configuration has been developed and presented. Additional dynamic considerations regarding the coupling mechanism, disturbance modeling, and potential aerodynamic effects have been addressed. A simulation has been built to study and animate the aforementioned flight conditions and situations, utilizing the dynamics presented in chapters two and three. A presentation regarding the practical implementation and other considerations, such as gyro bias and serial protocols, is also shown and confirm the feasibility of the system design.

REFERENCES

Achtelik, M., T. Zhang, K. Kuhnlenz and M. Buss, “Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors”, in “IEEE International Conference on Mechatronics and Automation”, (2009).

Akoun, G. and J.-P. Yonnet, “3D analytical calculation of the forces exerted between two cuboidal magnets”, IEEE Transactions on Magnetics pp. 1962 – 1964 (1984).

Ali, A., N. Ballou, B. McDougall and J. L. V. Ramos, “Decision-support tool for designing small package delivery aerial vehicles (dst-spdav)”, in “2015 IEEE Systems and Information Engineering Design Symposium”, (2015).

Blosch, M., S. Weiss, D. Scaramuzza and R. Seigwart, “Vision based mav navigation in unknown and unstructured envrironments”, (2010).

Bloss, R., “Unmanned vechiles while becoming smaller and smarter are addressing new applications in medical, agriculture, in addition to military and security”, Industrial Robot: An International Journal **41**, 82 – 86 (2014).

Brandt, J. B. and M. S. Selig, “Propeller performace data at low reynolds numbers”, in “AIAA Aerospace Sciences meeting”, (American Institute of Aeronautics and Astronautics, inc., 2011).

Chaturdevdi, N. A., T. Lee, M. Leok and H. McClamroch, “Nonlinear dynamics of the 3d pendulum”, (2011).

Coleman, C. P., “A survey of theoretical and experimental coaxial rotor aerodynamic research”, Tech. rep., Ames Research Center (1997).

Etkin, B. and L. D. Reid, *Dynamics of Flight stability and control* (John Wiley & Sons, 1996), third edn.

Furlani, E. P., “A formula for the leviation force between magnetic disks”, IEEE Transactions on Magnetics **29**, 6, 4165 – 4169 (1993).

Hoffmann, G. M., H. Huang, S. L. Waslander and C. J. Tomlin, “Quadrotor helicopter flight dynamics and control: Theory and experiment”, in “AIAA Guidance, Navigations and Control Conference and Exhibit”, (2007).

Huang, H., G. M. Hoffmann, S. L. Waslander and C. J. Tomlin, “Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering”, in “IEEE International Conference on Robotics and Automation”, (2009).

Leishman, J. G., *Principles of Helicopter Aerodynamics* (Cambridge University Press, 2006), second edn.

Luukkonen, T., “Modelling and control of quadcopter”, (2011).

Mahony, R., V. Kumar and P. Corke, “Multirotor aerial vehicles”, IEEE Robotics & Automation Magazine (2012).

Mellinger, D., N. Michael and V. Kumar, “Trajectory generation and control for precise agressive maneuvers with quadrotors”, The International Journal of Robotics Research (2012).

Mellinger, D. W., *Trajectory Generation and Control for Quadrotors*, Ph.D. thesis, University of Pennsylvania (2012).

Michael, N., S. Shen, K. Mohta, Y. Mugaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi and S. Tadokoro, “Collaborative mapping of an earthquake-damaged building via ground and aerial robots”, Field Robotics **29**, 832–841 (2012).

Nemati, A. and M. Kumar, “Modeling and control of a single axis tilting quadcopter”, in “American Control Conference”, (2014).

Ostojic, G., S. Stankovski, B. Tejic, N. Dukic and S. Tegeltija, “Design, control and application of quadcopter”, (2015).

Rudol, P., M. Wzorek, G. Conte and P. Doherty, “Mico unmanned aerial vehicle visual servoing for cooperative indoor exploration”, (2008).

Sa, I. and P. Corke, “Estimation and control for an open-source quadcopter”, in “Australasian Conference on Robotics and Automation”, (2011).

Sa, I. and P. Corke, “System indetification, estimation and control for a cost effective open-source quadcopter”, in “IEEE International Conference on Robotics and Automation”, (2012).

Schmid, K., T. Tomic, F. Ruess, H. Hirschmuller and M. Suppa, “Stereo vision based indoor/outdoor navigation for flying robots”, International Conference on Intelligent Robots and Systems pp. 3955–3962 (2013).

Schmidt, M. D., *Simulation and Control of a Quadrotor Unmanned Aerial Vehicle*, Master’s thesis, University of Kentucky (2011).

Simoes, C. M., “Optimizing a coaxial propulsion system to a quadcopter”, (N/A).

Thrun, S., W. Burgard and D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping”, International Conference on Robotics and Automation pp. 321–328 (2000).

Young, H. D. and R. A. Freedman, *University Physics*, vol. 2 (Pearson Addison-Wesley, 2007a), 12th edition edn.

Young, H. D. and R. A. Freedman, *University Physics*, vol. 1 (Pearson Addison-Wesley, 2007b), 12th edition edn.