STARS

Electronic Theses and Dissertations, 2004-2019

2019

A Smart UAV Platform for Railroad Inspection

Ryan Debevec
University of Central Florida

STARS Citation

Showcase of Text, Archives, Research & Scholarship

A SMART UAV PLATFORM FOR RAILROAD INSPECTION

by

RYAN DEBEVEC
B.S. University of Central Florida, 2019

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Mechanical and Aerospace Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2019

Major Professor: Tarek Elgohary

ABSTRACT

Using quadcopters for analysis of an environment has been an intriguing subject of study recently. The purpose of this work is to develop a fully autonomous UAV platform for Railroad inspection The dynamics of the quadrotor is derived using Euler's and Newton's laws and then linearized around the hover position. A PID controller is designed to control the states of the quadrotor in a manner to effectively follow a vision-based path, using the down facing camera on a Parrot Mambo quadrotor. Using computer vision the distance from the position of the quadrotor to the position of the center of the path was found. Using the yaw controller to minimize this distance was found to be an adequate method of vision-based path following, by keeping the area of interest in the field of view of the camera. The downfacing camera is also simultaneously observing the path to detect defects using machine learning. This technique was able to detect simulated defects on the path with around 90% accuracy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: BACKGROUND AND INTRODUCTION

A major Florida infrastructure project ongoing right now is the railroad connecting Orlando to Miami. With such a critical piece of infrastructure comes the need for cost and time efficient inspection, that will keep the system operating safely and effectively.

Currently there are several methods being implemented to inspect railroad tracks. Some the most recent methods include vision-based algorithms, with several showing success. These methods are intrusive of the system, as it requires a camera being mounted on a train cart or having a separate cart to use for inspection. The former being a costly addition to the train and the latter can cause costly and critical track closures.

Figure 1 Various platforms for railroad inspection

Research has been limited on the use of guidance, navigation, and control (GNC) of UAVs in conjunction with the vision inspection techniques. Vision-based inspection was explored using a cart as the platform, [1] [2]. Although, vision based GNC of UAVs is very popular, with several contributions being found, [3] [4].
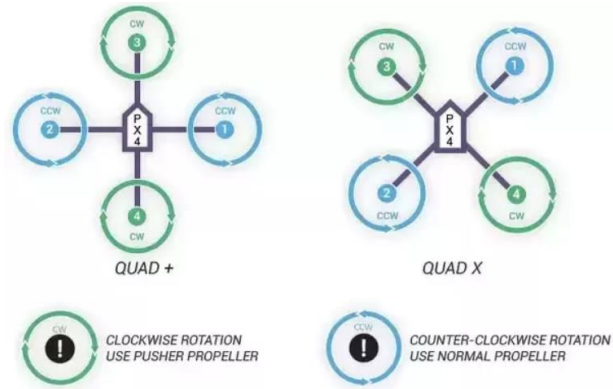
Figure 2 (+) configuration and (x) configuration of a quadcopter [5].

The quadcopter has grown to become a very versatile and highly popular system to use for many applications. Small unmanned aerial vehicles (UAV) have been made possible by the advances in sensors, microcomputing, control and aerodynamic theory. The small size of the sensors needed in the quadcopter bring forth the problem of the sensors creating more noise. The Micro-Electro-Mechanical Systems (MEMS) sensors that are needed at this size are susceptible to temperature and vibrations as they are placed in a compact shell and it is hard to isolate them from the vibrational effects of the quadcopter system. Another problem that comes about with small aircrafts, generally less than a meter, is stability and robustness to an unpredictable environment. A strong focus must be applied to solving these challenges and a solution is pretty well documented at this point but is still being constantly perfected, [6] [7] [8] [9] [10] [11] [12] [13].

Currently there are several nondestructive methods for railroad inspection, [14]. There have been several successful implementations of vision-based methods to conduct these inspections as well, [1] [15] [16]. Some of these methods utilize attaching a camera to a specialized cart or placing it on an existing train cart, [1] [2]. This is usually a costly implementation, with some of the methods needing rail closures or slower than normal speeds of the trains. This can have an adverse effect on the railway system itself as it limits the effectiveness of the system. There has been limited

research on using UAVs to conduct the railroad inspections, with a lot of the focus being on computer vision, [4]. There are none that couples this with guidance, navigation and control algorithms. There are however plenty of work relating to the use of the guidance, navigation, and control algorithms used on UAVs, [3] [17]. In this work a fully autonomous UAV platform that can conduct inspections on simulated railway defects in various lighting and in a GPS denied environment is developed.

Objectives

The objectives of this thesis are to develop, implement, and analyze:

- Development of a platform using a UAV to autonomously navigate railways.
- A method for detecting defects on the railroad with a vision-based method for aerial detection

Outline

Chapter 2 is dedicated to showing the literature review that was conducted to obtain knowledge of current and past work related to the challenges at hand. Chapter 3 consists of the mathematical modeling of the quadcopter. All theory, equations and implementation of the modeling will be represented here.

Chapter 4 deals with all the control techniques used in this project. It will show the deriving and implementation of PID and LQR controllers. Path following will conclude this chapter as the method uses image processing and a controller to minimize the error of the tracked path and quadrotor position.

Chapter 5 details how defect detection will be used and the analysis of the performance. Machine learning and neural networks will also be discussed in this section as it will be needed for the detection method.

Chapter 6 will be the results and discussion portion of the paper. It will analyze each component of the project and represent the performance of the system. There will also be a discussion on the results of the analysis.

Finally, Chapter 7 will state the conclusion of the findings from the project as well as the future work that will be pursued.

CHAPTER TWO: LITERATURE REVIEW

Dynamics and System Modeling

There was a great deal of work found describing the derivation of a quadcopter's dynamics and system identification with 6-degrees of freedom (DOF). It is a very well documented procedure and some of the contributions focused on system identification [18], [19], while others used Newton's and Euler's laws, [20], [9], [11], [21].

Using Newton's laws and Euler's laws, the force and moment equations can be derived, considering a quadcopter in the plus configuration. A two-frame system, involving the inertial frame and the body frame, and the transformation matrices was used to relate the different forces with some being convenient to describe in the different reference frames. Some of the assumptions made in the development of the quadcopter's dynamics were to treat the quadcopter as a rigid body, the center of gravity aligns with the center of the quadcopter, and to neglect the aerodynamics effects. The state space model can then be organized using the derived equations. It can then be used to generate a linear model of the quadcopter.   A simplified model, by small oscillations, was used to do the linearization. The model was linearized around the hover position and the input value needed to counter the weight of the quadcopter. The linear model was used to obtain the A, B, C, and D matrices, [11] [21].

Control Law

Control methods for quadrotors has been a popular subject of research, with well documented work. Some of the most common works utilizing backstepping [7], [22], sliding mode [23], [24], Linear Quadratic Regulator (LQR) [10], [25], [11], and Proportional-Intergral-Derivative (PID) [26], [8], [6], [21], [27] control techniques.

The focus of the control technique used in this work will be using the PID controller. It has been concluded that a PID controller, used on a quadrotor, has shown smooth flight conditions, good disturbance rejection, and reliable control. PID controller is commonly used in the industry. It was stated that a PID controller cannot give robust performance, as it depends on the error of the Euler angles alone. This type of controller neglects the system transient response due to the angular rate, which leaves the classic PID controller limited. A cascade PID controller utilizes multiple PID controllers in series and the feedback from the outer loop becomes the setpoint for the inner loop controller (secondary). The angle and the angular rate are controlled, and it was shown that it gives better disturbance rejection. The robustness of the controller is then greatly increased compared to the classic PID control law. The cascading PID drives the Euler angle to converge to the desired angle and it limits the angular rate during this convergence, which results in a smoother transient response controller. Wang shows numerous simulation results of the pitch, roll, and yaw responses to show the improvement that comes with using cascading PID control, [21].

The basic knowledge of Laplace transforms and transfer functions is enough to develop an effective method. A block diagram is a great method to use in these methods, as it gives an overview perspective of the system and the block can be defined as transfer functions. Amethod was found on how to design PI controllers for first order systems and PI and PID controllers for second order systems and also describes the structure of stabilizing controllers. The characteristic equation was obtained and then it was described in how to use it to find the controller parameters. Insight is also given on selecting relative damping and how when designing a cruise control system, a relative damping of one, critically damped, is usually used to make the response smooth.

The undamped natural frequency in the characteristic equation gives response speed and relative damping gives the shape, [27].

Path Following/Tracking

Various methods for path following/tracking has been proposed, such as, characterizing the path using color identification [2], [5], [28], [29], vanishing points analysis [30], [31], funnel lane navigation [32], [33], and edge detection [34], [35], [36], [37] These have all shown to be capable methods of path following.

The method using color characterization to identify the path was used in this work. It was found that a vehicle in a GPS denied environment can follow a path with vision and that this can help minimize the reliance on obstacle avoidance. The path the vehicle is following would have to be assumed obstruction free. A yaw tracking controller was used to follow the target as to make the field of view of the camera efficient. This makes it so the offset of the image is zero. This is achieved through a proportional gain on the yaw rate. Only a proportional gain is required as the low-level controller sets the heading from the given yaw rate. The Zielgler-Nichols method was utilized to tune the controller in an indoor environment. A higher gain can be found through further fine tuning, that gives a faster response to disturbances. It was also noted that there is a decrease in the position and height controller performance if the response time of the yaw controller was any further improved, [38].

Defect detection

Detecting surface defects in railways via computer vision was discussed in [39]. The necessary principles are machine vision, image acquisition device with auxiliary lighting and shading box, and portability. Some of the results are that the method had 4.65 ms maximum positioning and

position failure rate of 5 percent. The real-time detection speed is 2 m/s which is capable of real-time detection for hand walking. The time to process each image is 245.61 ms proving the ability for real-time detection. It is stated that the rail, sleeper, and fastener are the most important components to detect defects on within a railroad track. Some of the main defects needing detection are deformed rails and broken sleepers. These defects can be inspected by the eye, which in turn makes it possible to detect with machine vision. The research done in this article is mainly based on cracks, peeling, scratches, folds, broken rails, scars, and some other forms. The inspection of the rails will have a strict detection time constraint, which increases the importance of being able to quickly detect the railway itself. Hue mutation is to quickly pick out the targeted areas. Some of the standards used in this type of application are the use of hue, saturation, and lightness, as well as colors that the human eye can perceive. The hue characteristic is mentioned to be stable in different lighting condition. It is easy to influence the characteristics in the images so image filtering is said to be a must. In this article the main filter talked about is the median filter. The variation between pixels is analyzes to make a comparison. Color and temperature are less sensitive than contours. So they used the canny algorithm to convert the image to binary to get the image contour. Chain code is then used to track the contours. There is not one parameter to describe to contour defects, so a table is shown to show the different parameters [39].

Defects that relate to rail gauge or the space between the rails were discussed in [1], where two different methods that were used. One was two pairs of unaligned cameras and the other is using a pair of range finders to develop a depth map. Other things that were detected were damaged or missing rail fasteners, tie clips, and bolts. To make the detection real-time a GPU based library was used for parallel computation. It is stated in his dissertation that rail gauge is the most important measurement as it is used as an indicator of the other defects. A 3D panorama is made

and then some filtering and smoothing is applied. It is demonstrated how to apply computer vision methods to look at railway components that are fixed in looks. He uses a template the has the gray scale features to match up against the detected ones. Correlation filters are used, but it is noted that a processing difficulty is present, and it is managed by parallel computing. It is declared that a 0.98 rate for fastener detection was achieved, [1].

In [40], a double-layer data-driven frameworks for automated rail surface inspection was discussed. First the method uses the images to detect location of cracks and next automatically get the crack boundaries from a feature-based linear iterative crack aggregation. It is presented with six benchmarking methods to prove the advantages of the approach proposed. These six are the Otsu's method, mean shift, the visual detection system, the geometrical approach, fully convolutional networks, and the U-net [40].

Using UAVs for railway detection is a method that has not been implemented. The UAV is a platform that should be able to advance the railway inspection method with less intrusion into the system and become a more effective and efficient inspection platform. It is being proposed as not a total replacement of current platforms, but it should also be able to thrive as a supplemental platform to the others and fill the gaps of inspection that exist.

CHAPTER THREE: MATHEMATICAL MODELING OF A QUADROTOR
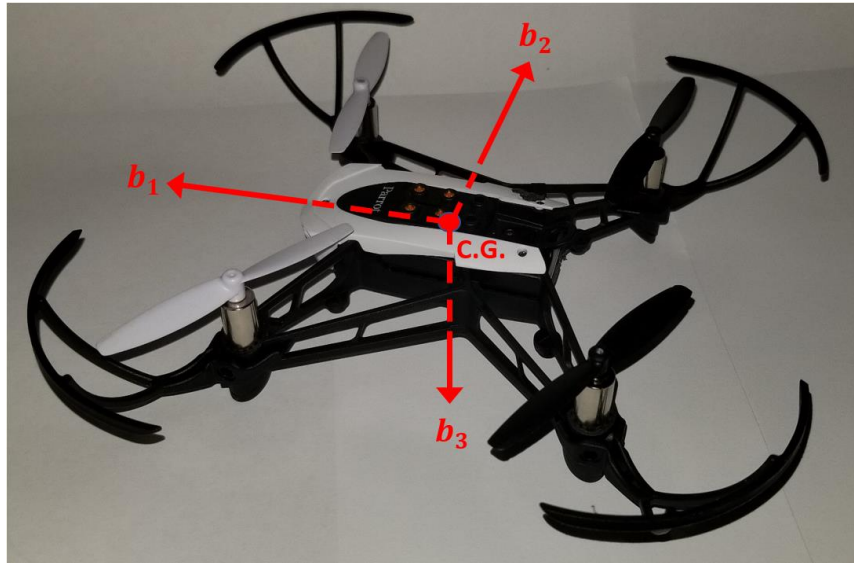
Figure 3 Body coordinate system of the quadrotor

The quadcopter that was chosen for use in this project was the Parrot Mambo because of its ability to be coded, relatively cheap price for testing purposes if it breaks, and the small frame to maximize the flight capabilities in a laboratory environment. As shown in Figure 3, the X-configuration was chosen mostly for the purpose of keeping the proper orientation of the camera. This is especially important in this project, as the cameras images will be used for line tracking. The b_1-axis is positive out of the front of the quadcopter, the b_2-axis is positive out of the right side, and due to the righthand rule the b_3-axis is positive out of the bottom.
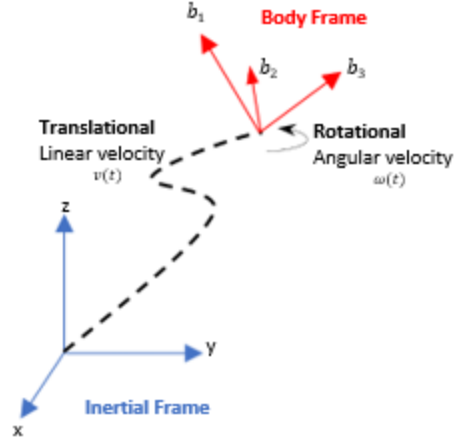
Figure 4 From Inertial to body frame, translational and rotational changes

A quadrotor has 6 degrees of freedom (DOF) with 3-DOF for translation and 3-DOF for rotation.

To describe the motion of the quadcopter, the following two vectors for linear velocity in the body-frame (v^b) and angular velocity (ω^b) in the rotating body-fixed reference frame will be defined.

$$v^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} longitudinal\ velocty \\ lateral\ velocity \\ normal\ velocity \end{bmatrix} \tag{1}$$

$$\omega^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} roll\ rate\ (x-axis) \\ pitch\ rate\ (y-axis) \\ yaw\ rate\ (z-axis) \end{bmatrix} \tag{2}$$

Next, the forces that influence the quadrotors motion are defined. These will again be separated into translational and rotational vectors

$$Force = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} longitudinal\ force \\ lateral\ force \\ normal\ force \end{bmatrix} \tag{3}$$

$$Moment = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} Moment\ about\ \hat{b}_1 - axis \\ Moment\ about\ \hat{b}_2 - axis \\ Moment\ about\ \hat{b}_3 - axis \end{bmatrix} \tag{4}$$

Where M_x, M_y, and M_z are rotations about the x, y, and z axis in the body frame. The naming of these axes in the body frame use the nomenclature $\hat{b}_1$, $\hat{b}_2$, and $\hat{b}_3$ for x, y, and z respectively to reduce confusion with the x, y, and z axis in the inertial frame.

The "influences" can either be in the inertial frame (gravity) or the body frame (propulsion). There will be some conversions back and forth between these two frames.. The conversions need to be done in a specific sequence and in this work the 3-2-1 sequence is used, shown in Figure 7. Therefore, the three Euler angles (below) will need to be defined and tracked as well. Φ is the rotation about the x-axis, θ is the rotation about the y-axis, and ψ is the rotation about the z-axis.

$$\Phi = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} bank\ angle \\ pitch\ angle \\ heading \end{bmatrix} \tag{5}$$

To make sure the right-hand rule is followed for the dynamics equations, the resulting inertial reference frame ends up being North, East and Down. For navigation purposes, it is convenient to define height in the positive direction. This leads to the relation:

$$\begin{bmatrix} x^E \\ y^E \\ h^E \end{bmatrix}_{Navigation} = \begin{bmatrix} x^n \\ y^n \\ -z^n \end{bmatrix}_{Inertial} \tag{6}$$

The reference frame for which each of these variables is in reference to is expressed and needs to be considered. The translational and the rotational motion will be with respect to the inertial frame. A rigid body is assumed, so there will be no motion with respect to the body frame. The sensors that detect the changes of these states are attached to the body of the quadrotor, so the equations of motion (EOM) that are derived in the Inertial frame will be represented in the body frame.

To do this the chain rule needs to be used. This means that both the change due to the time derivative of the vector within the coordinate frame needs to be accounted for, as well as the time derivative of the coordinate frame's rotation.

Newton's second law is used to derive the quadrotors dynamics. Newton's law can is defined as:

$$Force = \frac{d\bar{p}}{dt} \tag{7}$$

Where, p, is the momentum. For systems with constant mass, this law can be written to give the linear form of the law:

$$Force = m\frac{d\bar{v}}{dt} = m\bar{a} \tag{8}$$

From the linear form of the law, the rotational form can be defined to give the Moments by the crossing both sides of the equation with a position vector:

$$\bar{r} \times Force = \bar{r} \times \frac{d\bar{v}}{dt} \tag{9}$$

This can be simplified to:

$$Moment = \frac{dH}{dt} \tag{10}$$

Where, H, is the angular momentum. If the system has a constant distribution of mass, then it can be reduced to:

$$Moment = I\Omega \tag{11}$$

Where, I is the moment of inertia and Ω is the angular acceleration.

Focusing on the linear form of the law the external forces, which come from the thrust of the four

propellers ($F_1, F_2, F_3, and\ F_4$), can be represented by the thrust vector:

$$Propulsive\ Force = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_{body} = \begin{bmatrix} 0 \\ 0 \\ -F_1 - F_2 - F_3 - F_4 \end{bmatrix}_{body} \tag{12}$$

The linear forces are in the z-direction and this is logical, since the force from the propellers is

always in the z-direction. It is also noted that the forces are also negative, since the $\hat{b}_3$ is positive

in the down direction.

The gravity force also needs to be considered. The gravity force points towards the center of the

Earth and is expressed in the Inertial frame as

$$Gravity\ Force = F^n_{gravity} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}_{Inertial} \tag{13}$$

It now needs to be converted to the body frame and to do this the Euler angle transformation matrix

of a 3-2-1 sequence will be used as defined below

$$C^b_n = \begin{bmatrix} C_\theta C_\psi & C_\theta S_\psi & -S_\theta \\ -C_\theta S_\psi + S_\phi S_\theta C_\psi & C_\phi C_\psi + S_\phi S_\theta S_\psi & S_\phi C_\theta \\ S_\phi S_\psi + C_\phi S_\theta C_\psi & -S_\phi C_\psi + C_\phi S_\theta S_\psi & C_\psi C_\theta \end{bmatrix} \tag{14}$$

Where S and C are placeholders for Sine and Cosine respectively. Now gravity can be expressed

in the body frame as

$$F^b_{gravity} = C^b_n F^n_{gravity} \tag{15}$$

$$= \begin{bmatrix} C_\theta C_\psi & C_\theta S_\psi & -S_\theta \\ -C_\theta S_\psi + S_\phi S_\theta C_\psi & C_\phi C_\psi + S_\phi S_\theta S_\psi & S_\phi C_\theta \\ S_\phi S_\psi + C_\phi S_\theta C_\psi & -S_\phi C_\psi + C_\phi S_\theta S_\psi & C_\psi C_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$

$$F_{gravity}^n = \begin{bmatrix} -mgsin(\theta) \\ mgsin(\phi)cos(\theta) \\ mgcos(\phi)cos(\theta) \end{bmatrix} \tag{16}$$

Now that the forces are defined, the moments can be defined as

$$Moments = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} Moment\ about\ \hat{b}_1 - axis \\ Moment\ about\ \hat{b}_2 - axis \\ Moment\ about\ \hat{b}_3 - axis \end{bmatrix} \tag{17}$$
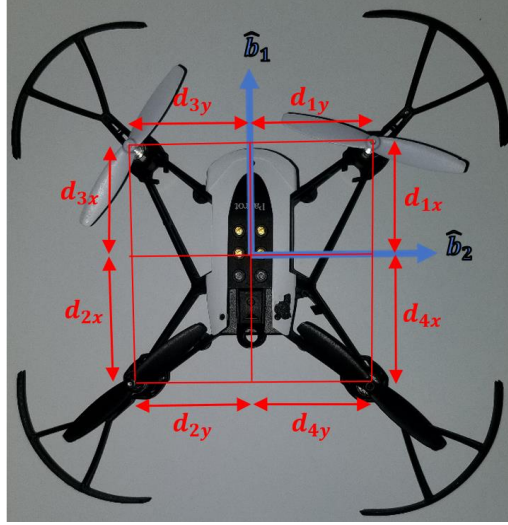
Figure 5 Distances from the axes to the motors

Above, in Figure 5, the distance from the x and y axes to the motors is defined. The moments are then expressed as

$$L_{(roll)} = F_1 d_{1y} - F_2 d_{2y} - F_3 d_{3y} + F_4 d_{4y} \tag{18}$$

$$M_{(pitch)} = -F_1 d_{1x} + F_2 d_{2x} + F_3 d_{3x} - F_4 d_{4x} \tag{19}$$
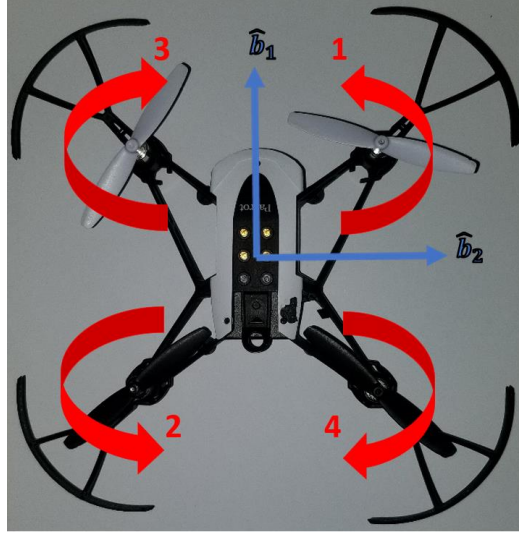
Figure 6 Rotation direction of the propellers (1 through 4)

The yaw moment (N) can be derived from the torque produced by the spinning propellers. The direction of the spinning propellers is chosen as #1 and #2 spinning in the counterclockwise direction and propellers #3 and #4 spinning in the clockwise direction (Figure 6). This gives the resulting equation for yaw:

$$N_{(yaw)} = -\tau_1 - \tau_2 + \tau_3 + \tau_4 \tag{20}$$

Where τ, is the torque of the corresponding motor as a function of its thrust, propeller radius, and the distance from the center of gravity.

The moment of inertia is defined in a 3x3 matrix, since it depends on the axis of rotation. To find the values for the matrix the mass is integrated over the body of the quadcopter.

$$I = \int r^2 dm = \int [r \times r \times dm] = \int \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} dm \tag{21}$$

$$I = \begin{bmatrix} \int (y^2 + z^2)dm & -\int (xy)dm & -\int (xz)dm \\ -\int (xy)dm & \int (x^2 + z^2)dm & -\int (yz)dm \\ -\int (xz)dm & -\int (yz)dm & \int (x^2 + y^2)dm \end{bmatrix} \tag{22}$$

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{23}$$

Due to the assumption of the quadrotor being symmetric about the x-axis and y-axis, this leads to only the diagonal components of the matrix remaining

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{24}$$

The linear and angular acceleration needs to be defined now and to do this the Coriolis Theorem will be used to describe the inertial velocity derivative in the rotating body frame. The chain rule must be used here to get the equation

$$\dot{v}_{inertial} = \dot{v}^b + \omega^b \times v^b = \dot{v}^b + \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} v \tag{25}$$

After using the previously defined v^b and ω^b, this equation becomes

$$\dot{v}_{inertial} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \tag{26}$$

Newton's Second Law for translation motion can be defined as

$$F_{propulsive} + F_{gravity} = m\dot{v}^b_{inertial} \tag{27}$$

Combining the force, quadcopter dynamics in the body force is given by

17

$$\begin{bmatrix} -mgsin(\theta) \\ mgsin(\phi)(\theta) \\ -F_1 - F_2 - F_3 - F_4 + mgcos(\phi)\cos(\theta) \end{bmatrix} = m\begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (28)$$

For angular acceleration and rotational motion, the Coriolis Theorem is used in a similar way to

get

$$M = I^b \dot{\omega}_n^b + \omega_n^b \times I^b \omega_n^b \quad (29)$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (30)$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} I_{zz}qr - I_{yy}qr \\ I_{xx}pr - I_{zz}pr \\ I_{yy}pq - I_{xx}pq \end{bmatrix} \quad (31)$$

$$\dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}}qr + \frac{L}{I_{xx}} \quad (32)$$
$$\dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}}pr + \frac{M}{I_{yy}}$$
$$\dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}}pq + \frac{N}{I_{zz}}$$

The rotational kinematics are computed by the rate of change of the Euler angles $[\phi\ \theta\ \psi]^T$. The

angular velocity as a function of the rate of change of the Euler angles can then be expressed by

representing them all in the body frame of reference. The sensors are all on-board the quadrotor

and reference the body frame. To do this the transformations have to be done in a specific series

of sequences. It is a sequence that first will be done about the z-axis, y-axis, and then the x-axis.

The representation is $R(\psi), R(\theta),$ and $R(\phi)$ respectively. This transformation is known as the

3-2-1 transformation. The intermediate frames of this transformation are represented as b'' and b'

(Figure 7).
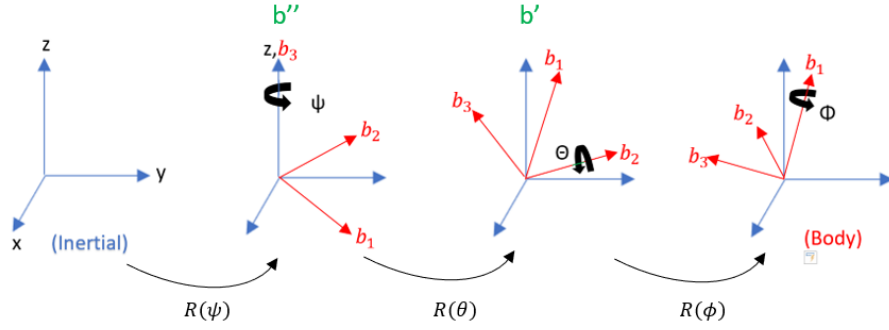
Figure 7 Intermediate steps of transforming from Inertial Frame to Body frame

Now using the coordinate transformations,

$$R(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & S_\phi \\ 0 & -S_\phi & C_\phi \end{bmatrix} \tag{33}$$

$$R(\theta) = \begin{bmatrix} C_\theta & 0 & -S_\theta \\ 0 & 1 & 0 \\ S_\theta & 0 & C_\theta \end{bmatrix}$$

$$R(\psi) = \begin{bmatrix} C_\psi & S_\psi & 0 \\ -S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

the angular velocities $[p \; q \; r]^T$ can be expressed as a function of the Euler angle rates $[\phi \; \theta \; \psi]^T$ as

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix}^b = R(\phi)R(\theta)\begin{bmatrix} 0 \\ 0 \\ \dot\psi \end{bmatrix}^{b''} + R(\phi)\begin{bmatrix} 0 \\ \dot\theta \\ 0 \end{bmatrix}^{b'} + \begin{bmatrix} \dot\phi \\ 0 \\ 0 \end{bmatrix} \tag{34}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} -\dot\psi S_\theta \\ \dot\psi C_\theta S_\phi \\ \dot\psi C_\phi C_\theta \end{bmatrix} + \begin{bmatrix} 0 \\ \dot\theta C_\phi \\ -\dot\theta S_\phi \end{bmatrix} + \begin{bmatrix} \dot\phi \\ 0 \\ 0 \end{bmatrix} \tag{35}$$

This results in the equations

$$p = \dot\phi - \dot\psi\sin(\theta) \tag{36}$$

19

$$q = \dot{\theta}\cos(\phi) + \dot{\psi}\cos(\theta)\sin(\phi)$$

$$r = \dot{\psi}\cos(\phi)\cos(\theta) - \dot{\theta}\sin(\phi)$$

Which can be rewritten to give the Euler angle rates

$$\dot{\phi} = p + (q\sin(\phi) + r\cos(\phi))tan(\theta) \tag{37}$$

$$\dot{\theta} = q\cos(\phi) - r\sin(\phi)$$

$$\dot{\psi} = \big(q\sin(\phi) + r\sin(\phi)\big)\sec(\theta)$$

To get the last three states, which are the navigation coordinates, the body frame velocities can be

transformed using the Euler angle transformation matrix C_b^n

$$\begin{bmatrix} \dot{x}^E \\ \dot{y}^E \\ -\dot{h}^E \end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & -C_\theta S_\psi + S_\phi S_\theta C_\psi & S_\phi S_\psi + C_\phi S_\theta C_\psi \\ C_\theta S_\psi & C_\phi C_\psi + S_\phi S_\theta S_\psi & -S_\phi C_\psi + C_\phi S_\theta S_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b \tag{38}$$

$$\dot{x}^E = C_\theta C_\psi u + \big(-C_\theta S_\psi + S_\phi S_\theta C_\psi\big)v + \big(S_\phi S_\psi + C_\phi S_\theta C_\psi\big)w \tag{39}$$

$$\dot{y}^E = C_\theta S_\psi u + (C_\phi C_\psi + S_\phi S_\theta S_\psi)v + \big(-S_\phi C_\psi + C_\phi S_\theta S_\psi\big)w$$

$$h^E = S_\theta u - S_\phi C_\theta v - C_\phi C_\theta w$$

Using equations (38), (39, and (41), the full nonlinear 12th order system describing the quadrotor's

translational and rotational dynamics is then given by

$$
\begin{cases}
\dot{\phi} = p + (q\sin(\phi) + r\cos(\phi))\tan(\theta) & \text{(40)} \\
\dot{\theta} = q\cos(\phi) - r\sin(\phi) \\
\dot{\psi} = \left(q\sin(\phi) + r\sin(\phi)\right)\sec(\theta) \\
\dot{p} = \dfrac{I_{yy} - I_{zz}}{I_{xx}} qr + \dfrac{L}{I_{xx}} \\
\dot{q} = \dfrac{I_{zz} - I_{xx}}{I_{yy}} pr + \dfrac{M}{I_{yy}} \\
\dot{r} = \dfrac{I_{xx} - I_{yy}}{I_{zz}} pq + \dfrac{N}{I_{zz}} \\
\dot{u} = rv - qw - g\sin(\theta) \\
\dot{v} = pw - ru + g\sin(\phi)\cos(\theta) \\
\dot{w} = qu - pv + g\cos(\phi)\cos(\theta) + \dfrac{F_{propulsive}}{m} \\
\dot{x} = C_\theta C_\psi u + \left(-C_\theta S_\psi + S_\phi S_\theta C_\psi\right)v + \left(S_\phi S_\psi + C_\phi S_\theta C_\psi\right)w \\
\dot{y} = C_\theta S_\psi u + (C_\phi C_\psi + S_\phi S_\theta S_\psi)v + \left(-S_\phi C_\psi + C_\phi S_\theta S_\psi\right)w \\
\dot{z} = S_\theta u - S_\phi C_\theta v - C_\phi C_\theta w
\end{cases}
$$

Linearization of quadrotor's dynamics

The nonlinear dynamics of the quadrotor needs to be applied as to simplify the system, so a linear

controller can be applied. The linearization method used here is the small perturbations concept

and linearizing around an equilibrium point. This will give the linear dynamics that the linear

controllers can be applied to and the those controllers can be utilized on the actual plant

(quadrotor).

To start the linearization, small oscillations are assumed. This gives $\sin(\angle)=\angle$, $\cos(\angle)=1$, and

$\tan(\angle)=\angle$. When this method is applied to our state vector the resulting vector becomes:

$$\begin{cases} \dot{\phi} \approx p + r\theta + q\phi\theta \\ \dot{\theta} \approx q - r\phi \\ \dot{\psi} \approx r + q\phi \\ \dot{p} \approx \dfrac{I_{yy} - I_{zz}}{I_{xx}} rq + \dfrac{L}{I_{xx}} \\ \dot{q} \approx \dfrac{I_{zz} - I_{xx}}{I_{yy}} pr + \dfrac{M}{I_{yy}} \\ \dot{r} \approx \dfrac{I_{xx} - I_{yy}}{I_{zz}} pq + \dfrac{N}{I_{zz}} \\ \dot{u} \approx rv - qw - g\theta \\ \quad \dot{v} \approx pw - ru + g\phi \\ \dot{w} \approx qu - pv + g + \dfrac{-F_{propulsive}}{m} \\ \dot{x} \approx w(\phi\psi + \theta) - v(\psi - \phi\theta) + u \\ \dot{y} \approx v(1 + \phi\psi\theta) - w(\phi - \psi\theta) + u\psi \\ \dot{z} \approx w - u\theta + v\phi \end{cases} \tag{41}$$

The next step in linearizing this system is to find an equilibrium point that the model can be

linearized around. For this quadcopter problem the hover position is chosen.

All of the states will be zeros except for the position. When in the hover position there ideally is

no changes in the angles or in the position, which leads to the equilibrium state being

$$\bar{x} = [\,0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \bar{x}\ \ \bar{y}\ \ \bar{z}\,]^{T} \tag{42}$$

Where $\bar{x}$, $\bar{y}$, and $\bar{z}$ are arbitrary position components. The needed input vector can then be

resolved by setting the input thay relates to altitude equal to the force in the z_direction. This leads

to the input vector being

$$\bar{u} = [\,mg\ 0\ 0\ 0\,]^{T} \tag{43}$$

Using the small perturbation method to linearize the system, the higher order terms are neglected.

For an example in linearizing $\dot{\phi}$, each variable that shows up in the equation are given a initial

value, plus a small change (perturbation). Hence the expansion of $\dot{\phi}$ is given by

$$\phi = \phi_0 + \delta\phi \tag{44}$$

$$\dot{\phi} = \delta\dot{\phi}$$

$$\dot{\phi} = p \tag{45}$$

So for simplicity we will adopt equation 46. When this method is used for all the state equations,

we get the linearized state vector:

$$
\begin{cases}
\dot{\phi} = p \\
\dot{\theta} = q \\
\dot{\psi} = r \\
\dot{p} = \dfrac{L}{I_{xx}} \\
\dot{q} = \dfrac{M}{I_{yy}} \\
\dot{r} = \dfrac{N}{I_{zz}} \\
\dot{u} = -g\theta \\
\dot{v} = g\phi \\
\dot{w} = \dfrac{-F_{propulsive}}{m} \\
\dot{x} = u \\
\dot{y} = v \\
\dot{z} = w
\end{cases}
\tag{46}
$$

With the associated matrices:

$$
A =
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
\tag{47}
$$

$$\mathbf{B}=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \dfrac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \dfrac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \dfrac{1}{I_{zz}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \dfrac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (48)$$

When wind disturbance $d = \begin{bmatrix} F_{wx} & F_{wy} & F_{wz} & \tau_{wx} & \tau_{wy} & \tau_{wz} \end{bmatrix}^{T}$ is considered, it gives the D matrix:

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 & & \dfrac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & 0 & & \dfrac{1}{I_{zz}} \\ \dfrac{1}{m} & 0 & 0 & 0 & 0 & \\ & \dfrac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & & \dfrac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (49)$$

Where the linear model of the system is:

$$\dot{\mathrm{x}} = \mathrm{A} \cdot x + \mathrm{B} \cdot u + \mathrm{D} \cdot d \qquad (50)$$

From the system above, using the Matrices A, B, C, and D the transfer functions of the states can

be derived using the flowing process. Also using the values for the variables shown in Table 1

Table 1 Moment of Inertia values, aslong with mass and gravity values

VARIABLE	VALUE
I_{xx}	5.8286e-05 kg/m^2
I_{yy}	7.1691e-05 kg/m^2
I_{zz}	1.0000e-04 kg/m^2
m	0.0630 kg
g	9.81 m/s^2

Starting with the state space matrix equations

$$\dot{x} = Ax + Bu \tag{51}$$

$$y = Cx + Du$$

Then taking the Laplace transfer of both to get

$$X(s) = (sI - A)^{-1}BU(s) \tag{52}$$

$$Y(s) = CX(s) + DU(s)$$

Substituting X(s) into the Y(s) (output) equation and dividing by U(s) to get the transfer function

H(s)

$$H(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \tag{53}$$

This gives the open loop transfer functions for the states and they are separated by input

Input 1:

25

$$H(s)_w = \frac{15.87}{s} \tag{54}$$

$$H(s)_z = \frac{15.87}{s^2}$$

Input 2:

$$H(s)_\phi = \frac{1.716e04}{s^2} \tag{55}$$

$$H(s)_p = \frac{1.716e04}{s}$$

$$H(s)_v = \frac{1.683e05}{s^3}$$

$$H(s)_y = \frac{1.683e05}{s^4}$$

Input 3:

$$H(s)_\theta = \frac{1.395e04}{s^2} \tag{56}$$

$$H(s)_q = \frac{1.395e04}{s}$$

$$H(s)_u = \frac{-1.368e05}{s^3}$$

$$H(s)_x = \frac{-1.368e05}{s^4}$$

Input 4:

$$H(s)_\psi = \frac{10000}{s^2} \tag{57}$$

$$H(s)_r = \frac{10000}{s}$$

The plant transfer function in equations (57 -59 will be the basis for the control design of the quadcopter as discussed in the next chapter.

CHAPTER FOUR: CONTROL TECHNIQUES

This section expands on two types of control techniques. The first controller discussed is the PID controller, followed by the discussion on an LQR controller. Both control techniques use the linearized version of our system obtained in the previous section. The controller gets tuned to this linearized system and then used on the nonlinear system of the quadcopter. The tuning should be close to the necessary values to stabilize the quadrotor and can be further fine-tuned. Simulations were done for each control technique as well to graphically show the response of the system when controlled by these techniques. Figure 8 shows the response of the system when it is not controlled and the system quickly diverges. This shows the necessity for a controller to be implemented.
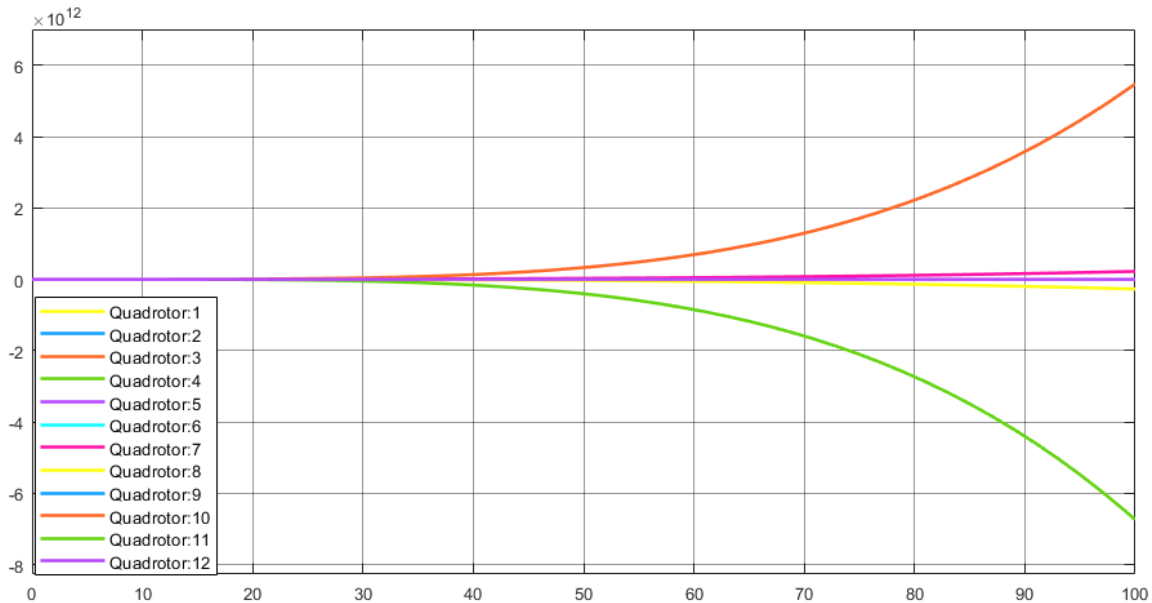

Figure 8 The states of the system diverging in response to a step input.

PID Controller

In this section, a PID controller is defined and it is showed how it was used for controlling the chosen states. A PID controller is used in a control feedback loop to drive a state to a selected value. It is broken down into three operations to carry this out. The three operations are the

proportional, integral and derivative controllers. Each one has a different function for driving the state to the desired value. Some systems may only need the proportional controller or a combination of the proportional and the others. Sometimes when just the proportional controller is used it will cause and oscillation and the system may never converge or may converge too slowly. To combat this, the derivative controller is coupled with it which helps drive the state to the desired value. Then if a faster convergence is needed the integral controller is added as well. This helps drive the state to the desired value more quickly. When the integral controller is used a phenomenon that must be watched called integral windup, due to an accumulated error in the process.

In the figure below, the general layout of a PID controller is shown. It starts with the error, which is the difference between the measured/estimated state and the reference input for that state. This error gets fed into the controller and goes through the three branches Proportional (P), Integral (I) and the Derivative (D). These three branches essentially access data from the present, past, and future. This helps by reacting to the present and past data and it helps predict what will happen shortly which can help with oscillation and overshooting the target value. For example, if there was no information about the future, then the system would not know to slow down ahead of time and pass the value and would have to correct this after passing the desired value. After the signal passes through the three branches, it then enters the plant as the input.
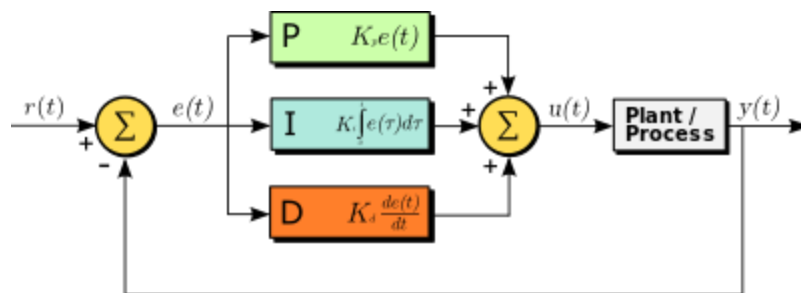

Figure 9 Simple block diagram of a PID controller with feedback

From the transfer functions derived in Chaper two and the transfer function of the controller, the parameters for the PID parameters can be equated. A PID controller transfer function is defined as

$$C(s)_{PID} = K_p + K_d s + \frac{K_i}{s} \tag{58}$$

This is used in the equation

$$\frac{H(s)_{plant} C(s)_{controller}}{1 + H(s)_{plant} C(s)_{controller}} \tag{59}$$

to compare the denominator to the characteristic equation

$$s^2 + 2\xi \omega_n^2 s + \omega_n^2 \tag{60}$$

With ξ being relative damping and ω_n being the undamped natural frequency. To show how this works, this method will be used to find the parameters of the PD roll controller, with the corresponding transfer function found below

$$\frac{\frac{b}{s^s}(K_p + K_d s)}{1 + \frac{b}{s^s}(K_p + K_d s)} \tag{61}$$

With b, being the numerator from the roll transfer function. When it is simplified it gives the denominator

$$s^2 + bK_d s + bK_p \tag{62}$$

to compare with the characteristic equation. This gives the equations

$$K_d = \frac{2\xi \omega_n^2}{b} \tag{63}$$
$$K_i = \frac{\omega_n^2}{b}$$

Now that the parameters K_p and K_d are defined, they can be implemented in the controller and use the Ziegler-Nichols method to fine tune them. The Ziegler-Nichols method is used by increasing K_p until there are small oscillations and then back it down. After this start increasing K_d until there are small oscillations again and then back it down until the oscillations are gone.

In the figures below, it can be how some of the different combinations of the 3 separate branches of a P-I-D controller is used to control different states. A simulation showing the response of each state correlating to a step input is shown along with the controller setup.

Figure 10 3D Simulation of Parrot Mambo mini drone flight using MATLAB/Simulink.
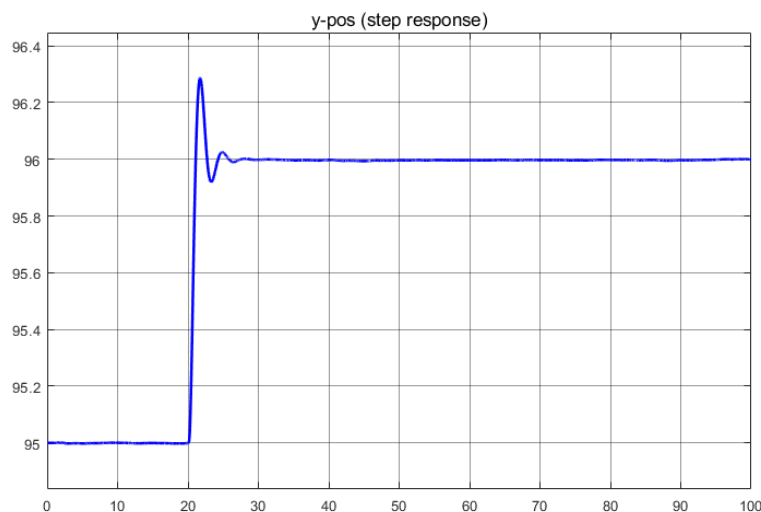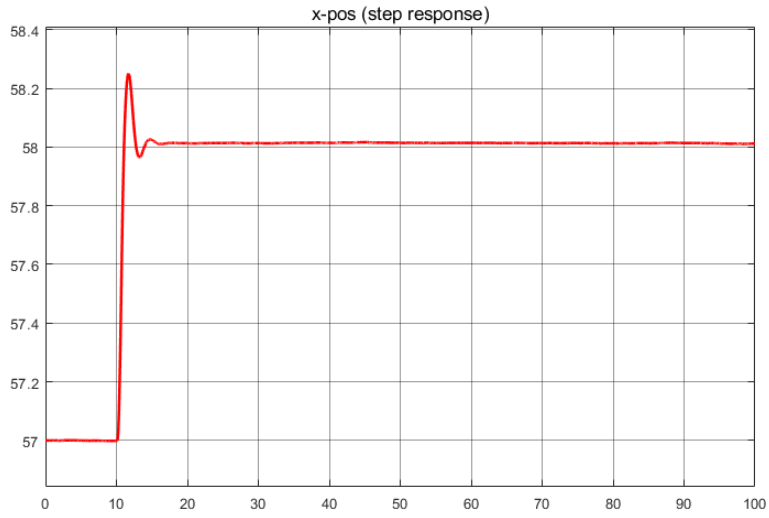

Figure 11 Step response of the y-position at 20 seconds

Figure 12 Step response of the x-position at 10 seconds
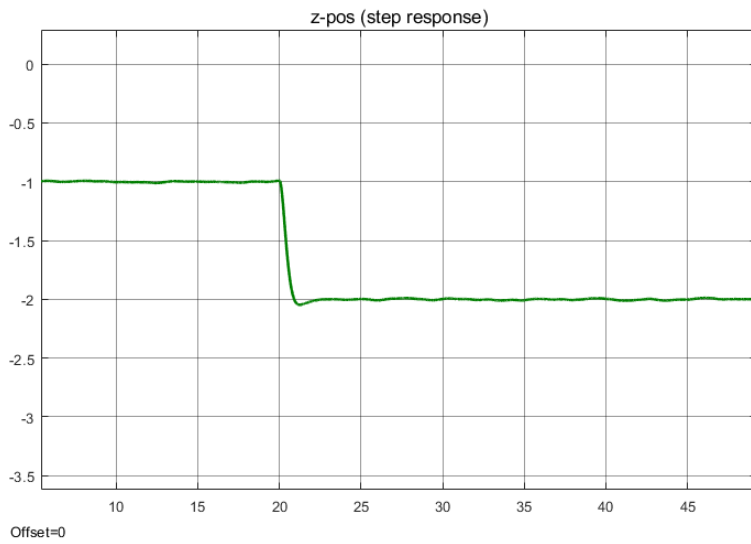
Figure 13 Step response of the z-position at 20 seconds
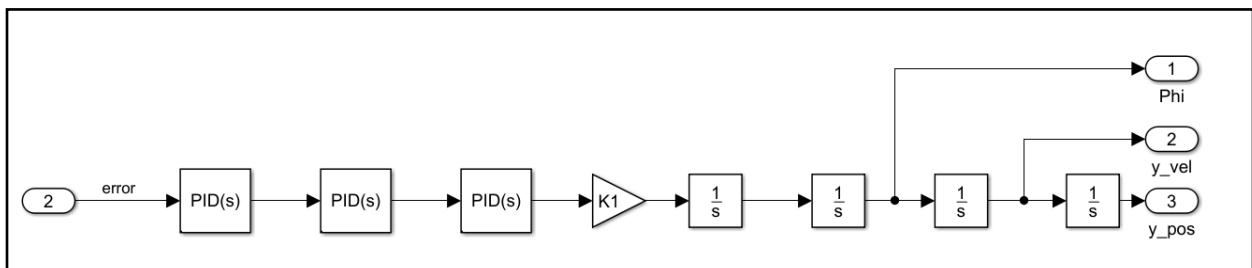
Figure 14 Simplified block diagram of y-position

31
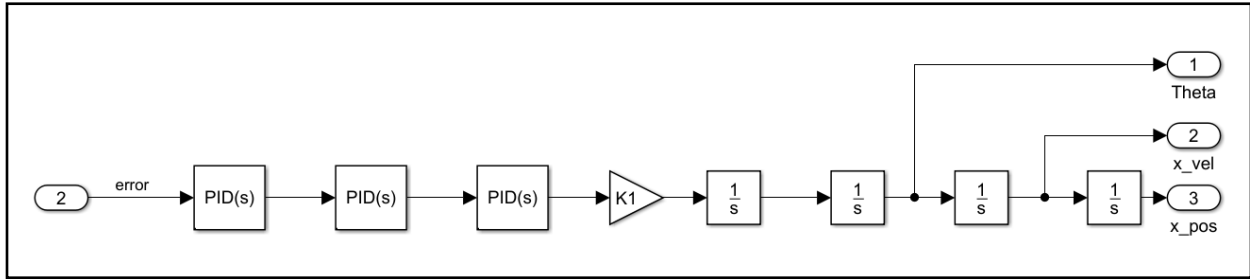
Figure 15 Simplified block diagram of x-position
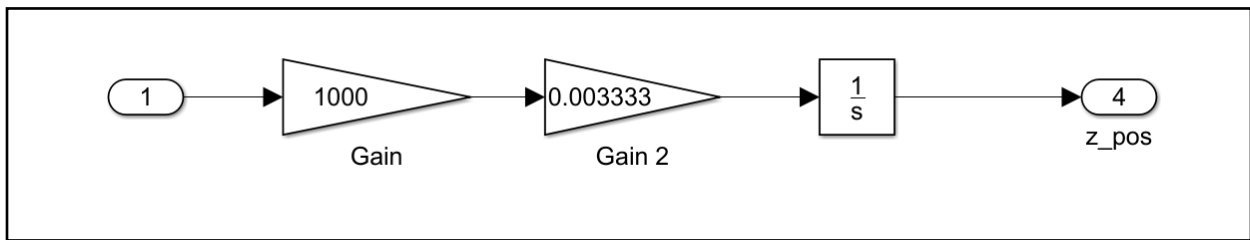
Figure 16 Simplified block diagram of z-position
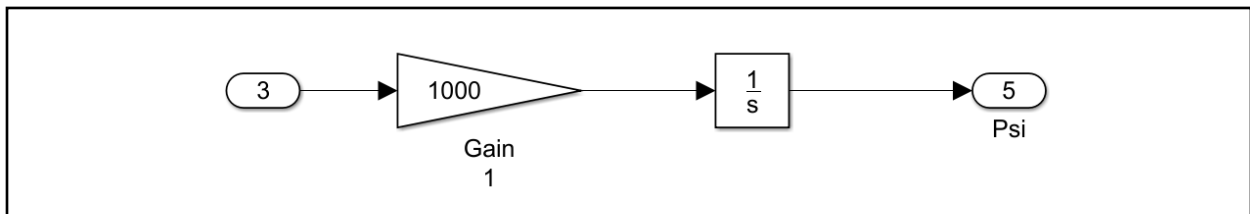
Figure 17 Simplified block diagram of yaw state

The controller was implemented in MATLAB and Simulink. The A,B,C, and D matrices and the values for different variables like mass, moment of inertia, and gravity are stored in the MATLAB workspace and referenced throughout the use of Simulink. The input to the system is the rpm of the propellers coinciding with the thrust force. As mentioned in the derivation of the dynamics of the system in Chapter 3, a combination of the thrust from the different propellers gives the input

to the different states being controlled. The outputs are the 6 states, which are the positions for x, y, and z as well as the orientation of the Euler angles.

With the combination of these states and the controllers shown, it is possible to stabilize the quad rotor. Below is a graph showing the results of a quadcopter being controlled by a PID controller for a hover position. A step input of four meters is being implemented with respect to the y-position, so it can be seen how quickly the quadcopter will converge to a value of four for the y-position.
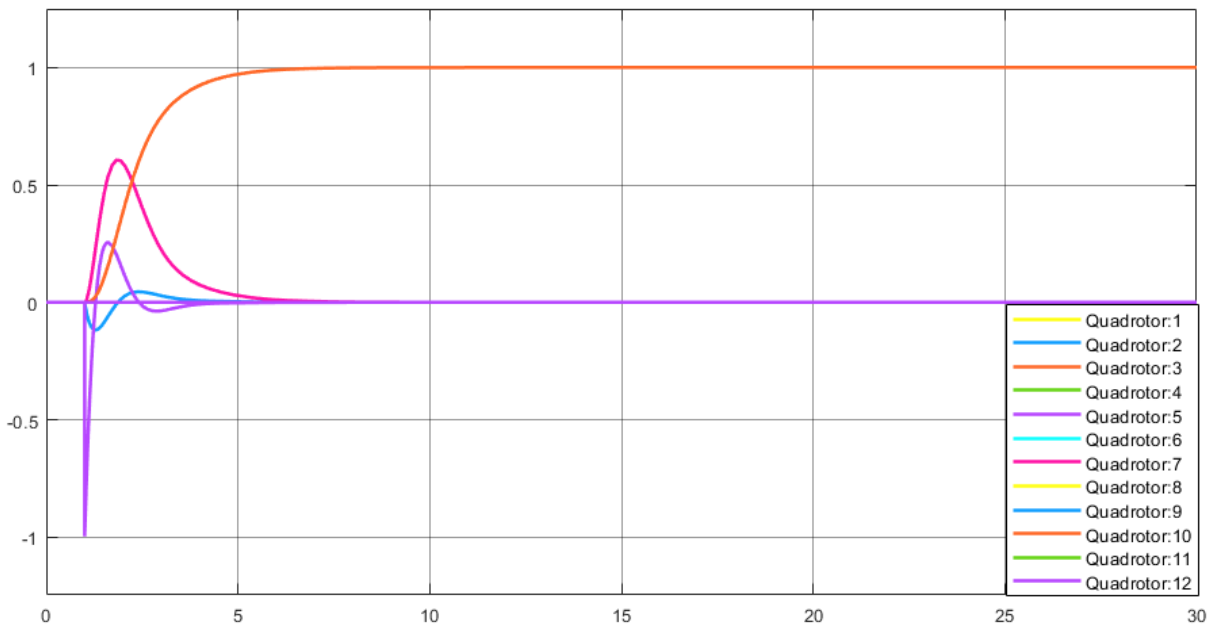
Figure 18 Simplified representation of the quadrotor's system, with a PID controller, responding to a step response.

Line Following Technique
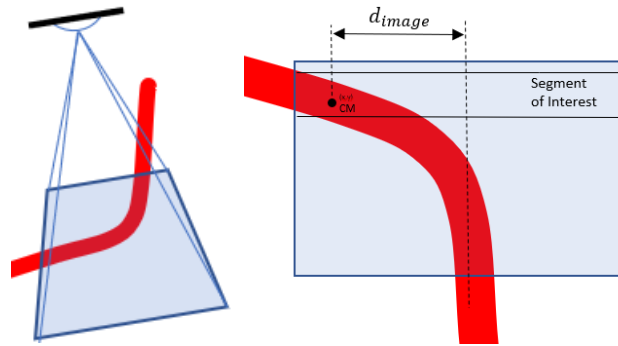
Figure 19 Illustrations showing the downfacing camera of the quadcopter and defining the distance 'd_image'

To simulate a railroad in a laboratory environment, it was decided that a line of red tape would be used for the quadcopter to follow.   The downfacing camera on the quadcopter was used to detect and track the line. Using Simulink, the Parrot Image Conversion block (see Figure 20) was used to take the image data captured by the camera and return the R-G-B values. The Green (G) and Blue (B) values were divided in half and subtracted by the Red (R) value. Next, the value obtained was compared to the value of 100 and anything above that number would illustrate that the camera was seeing the red tape.
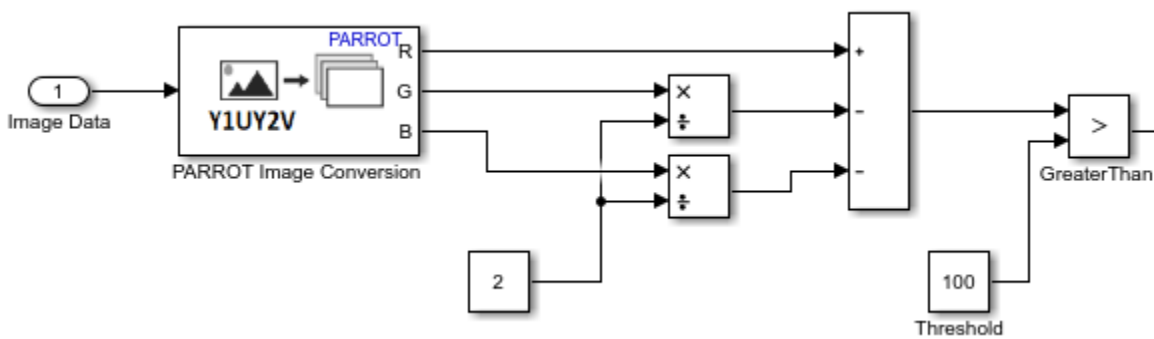
Figure 20 Block diagram of the system capturing the image and analyzing the R-G-B values
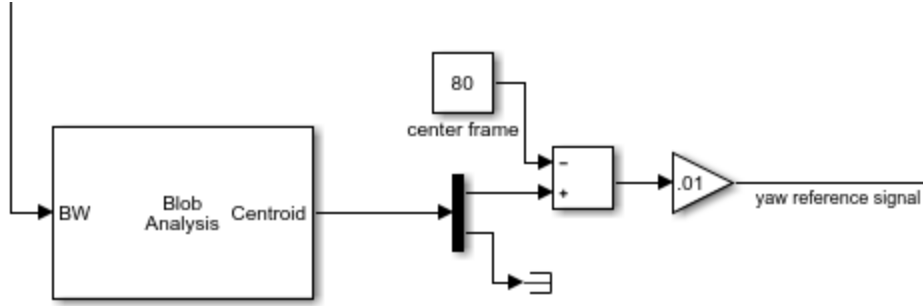
Figure 21 Blob analysis block and blocks used to determine yaw reference input

The output is a binary image that gets passed through a function block that takes a few rows from the top of the image. This segment acts as the steering for the quadrotor, as it is set to become the data that will be used to build the yaw reference values. As the red pixels shift from the middle to the right or left of the segment, it will indicate a desired change in the heading angle to follow the tape. The distance between red pixels and the center of the image will be weighted and be represented in the inertial frame. This will then be set as the error used in the yaw PID controller. Mathematically the yaw reference (ψ_{des}) is

$$d_{image} = d_{cm_x} - d_{ci_x} \tag{64}$$

$$\psi_{des} = d_{image} C_{45} + \psi\_est \tag{65}$$

$$\psi_{err} = \psi_{des} - \psi_{est} = d_{image} C_{45} \tag{66}$$

where C_{45} is a constant to put a weight on the input correlating $d_{image} = 40 \leftrightarrow \psi_{des} = 45°$ and d_{cm_x}, d_{ci_x} are the center of mass of the red pixels and center of the image respectively.

Now that the reference input for yaw is established, the quadrotor needs to move along the red tape in the desired direction. To keep the orientation of the camera compared to the line, it is desired that the quadcopter moves forward in the x-direction in the body frame. The reference input is

given in the inertial frame, so the desired x-direction in the body frame had to be changed to the inertial frame. This led to the x and y position reference input becoming

$$
\begin{aligned}
x_{ref} &= x_{est}^n + p * \cos(\psi) \\
y_{ref} &= y_{est}^n + p * \sin(\psi)
\end{aligned}
\tag{67}
$$

where p is a constant that relates to the desired velocity in the x-direction in the body frame.

Defect detection using machine learning

The quadrotor needs to be able to detect defects in the railroad and save a reference to the position of the defect as it finds it for later analysis. In the simulated laboratory environment machine learning was used to execute this. Machine learning gives the ability to train the neural network to look for and distinguish a feature in a picture. In the laboratory environment the letter X was chosen as the "feature" to search for.

The neural network used was the alexnet neural network. This network has good accuracy, as well as a good processing time. A balance of these two traits was viewed as important because it needs to be accurate, but it also needs to be ran on the test quadrotor which doesn't have a very high processing capacity. Other neural networks are available with a little higher accuracy, but it also increases the processing time. These will be further looked at as the project progresses and the code is built onto a quadrotor with better processing.

The neural network was trained with one hundred images with an X in it and one hundred images without the X. Through testing of the neural network, it was found to have around 85% accuracy. Accuracy as high as 90% was also seen during the testing.

The code runs the vision portion as a feed of pictures at a high rate. Every still image is analyzed and determined whether it has an X (defect) or if it does not. If it does detect a "defect" it will save the time the image was taken and the image itself. The quadrotor keeps a log of its position

with time, so the position of the "defect" is determined by referencing the saved time with the logged position data. After flight a file is then extracted that holds the recorded data of all the "defects" found for that flight.

Lab experimental setup description

Right now, in our research laboratory, numerous different quadcopters are being used. They range in size from small to large and from simple to complex. All the quadcopters are mostly using the same sensors, sensors also can be added and removed. Right now, they are using an Inertial Measurement Unit (IMU) consisting of two accelerometers and two gyroscopes for position and attitude measurements. They also use a sonar sensor for altitude determination. Cameras are also used on-board to reduce drifting, using optical flow techniques, and for imaging. Some of the quadcopters have multiple cameras for different views and functions. Below, in Figure 22, is a picture of the quadrotor that was used for the experiments.

Figure 22 Parrot mambo mini drone

In addition to having access to the various quadcopters, within our research laboratory there is access to numerous useful tools and space. There is a grid on the floor that gives visual indicators

and references of where the quadcopters should be during experiments. There are also motion detection cameras (OptiTrak) that gives what can be described as an indoor GPS. From using reference points on the quadrotor and an established coordinate system, the translational and rotational states can be established for a more accurate understanding of what is happening, and the coordinates can also be used as if the quadrotor was receiving outdoor GPS coordinates for feedback loops. There is also additional access to machine shop, 3-D printers and other various tools and references on campus to help make the end goal possible and give a more efficient process.

A 3-D simulation of the quadcopter is also available to analyze how the quadcopter will react with different inputs and controllers. It uses a 3-D model of the specific quadrotor that is being used and places it in a 3-D environment, so the user can visually see the response in a close to real world environment. This gives the power to try different controllers and setups, so they can be experimented with without the real-world risk of breaking components, risk of injury, and it also saves time in the fact that new things can be tried in quick succession without having to build the code to the quadrotor repetitively.

Quadcopters have been used in many applications in the area of inspections. They are great for integrating cameras and other sensors into the system. The fact they are easily hovered, and great maneuverability makes them a great base for these applications. The technology used in these quadcopters have increased greatly over the years also making them very affordable. It has been found that they have been used in this area for inspecting wind turbines, roads, buildings and many more inspections applications. It has been found that there is a need for the use of these systems in inspecting railroads.

CHAPTER FIVE: RESULTS AND DISCUSSION

The results that have been obtained from MATLAB /Simulink will be discussed and presented in this chapter. The step response of the states displays how the systems responds to a step input of 1. Next, the error of the quadcopter's position to the position of the line following path will be discussed. To end the chapter, the accuracy of the defect detection technique is analyzed.

State Response

The Quadcopter Project from Simulink provided a student development kit (sdk) for the Parrot Mambo mini drone that was used in our initial lab experiments. The 3D simulation license was needed to utilize 3D simulation to display a video of the simulated flight. It uses a model of the Parrot Mambo mini drone quadcopter that we also used for lab experiments. Figure 23 displays the response to a step input for the x, y, z, ψ with a combination of PID controllers applied to the system. The combinations and the gains applied are shown in

Table 2 Values for PID gains and shows combination of PID controllers per state

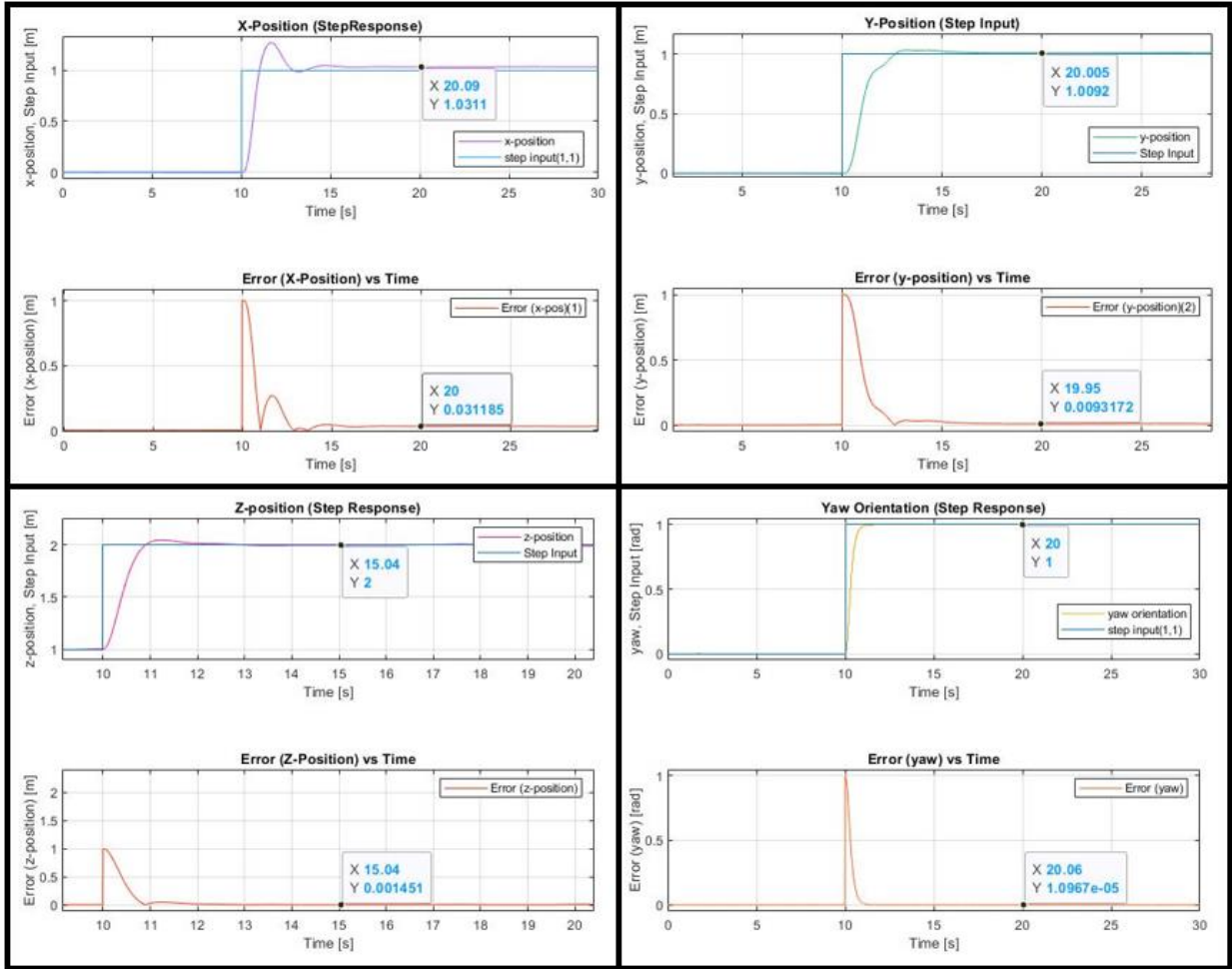| PID controller gains	k_p	k_i	k_d
x	-0.23	0	0.1
y	0.15	0	0.3
Altitude	0.85	0	0.35
Yaw	0.004	0	0.011
Pitch	0.013	0.01	0.003
Roll	0.03	0.01	0.004

Figure 23 Displays the step response and a graph of error vs time for the x,y,z,ψ states.

Figure 23 shows the response of the system to a step unit step input. These figures help to characterize the response of the system. The rise time, settling time, maximum overshoot, and the steady state error can be obtained from these plots, as shown in

Table 3 shows the rise time, settling time, maximum overshoot, and the steady state error for the x, y, and z states.

	Rise time [s]	**Settling Time [s]**	**Maximum Overshoot [m]**	**Steady State Error [m]**
x	1.2	2.6	1.38	0.031
y	2.7	4.3	0.02	0.0092
z	0.7	0.87	0.06	0.001
Yaw	0.7	0.85	0.098e-5	0.0967e-5

The controllers for this project didn't need to be extremely responsive, as only very smalls inputs were needed to follow the railroad tracks. The quadcopter will also be moving slow, as to increase the chances of correctly categorizing the images as having a defect or not. To have the most consistent images of the track a higher emphasis was placed on the steady state error. Graphs of the steady state error of the states can be found in *Figure 12* as well. The error in the step response of the yaw (ψ) state was extremely small and essentially zero in this simulation. The steady state errors were 0.031m and .009m for the x, y states respectively. These errors proved to be adequate for following a piece of tape on the ground used as the reference path. In the future other and more complex controllers will be experimented with tweak the response of the system to have a more ideal behavior for path following using vision.

Path Following

The quadrotor was able to follow the red piece of tape with a small enough error to keep the downfacing camera above the red tape to analyze it for defects. The maximum error that was

found using the vision path following technique used was 0.05 meters. Below is a graph of the error of the position of the quadcopter relative to the path.

Through the experiments it was seen that the error in the yaw relative to the angle needed to "steer" down the path was near zero, but there was still an error in the line following. In the future a line following controller will be placed to try to decrease the remaining error. Other techniques for line following using computer vision will be experimented with as well.
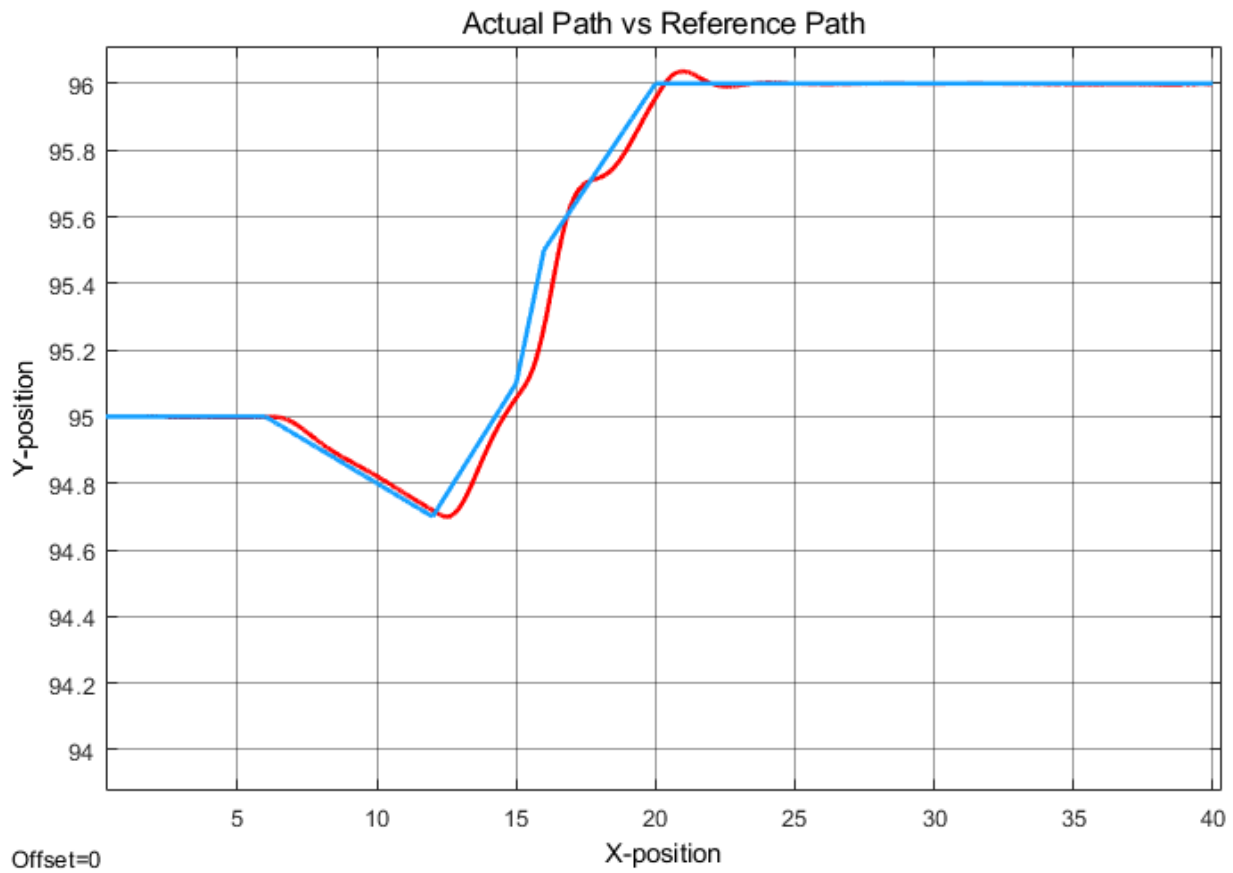
Figure 24 Actual path overlaid on top of the reference path
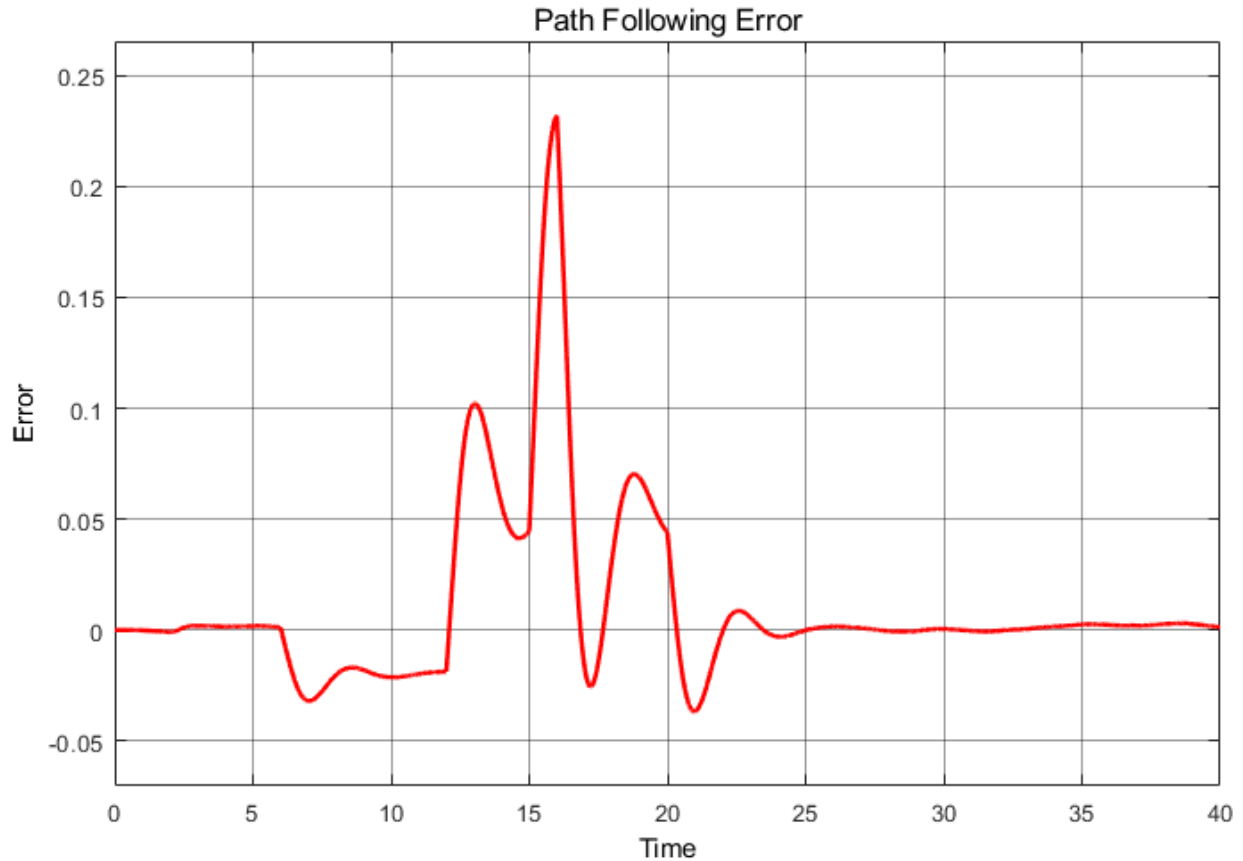
Figure 25 Error in path following

It can be seen from Figure 25 that the max error is 0.23 feet. This was shown to be a small enough error to keep the path in camera view to continue following the path and analyze the path for defects as it progresses along it.

Defect Detection Using Machine Learning

The alexnet neural network was trained to categorize the stream of images from the quadcopter as 'Not Defective' and 'Defective' if it detects an X in the image representing a defect. The network was tested with and a set of test images and both single images. In Figure 26 below, the statistics from the set of images test runs can be found and the returned accuracy was 90% (.90).

```
>> cnn_scratch
Training on single GPU.
Initializing image normalization.
|========================================================================|
|   Epoch   |  Iteration  | Time Elapsed | Mini-batch | Mini-batch | Base Learning|
|           |             |  (seconds)   |    Loss    |  Accuracy  |     Rate     |
|========================================================================|
|       1 |           1 |         2.97 |     0.6932 |     50.00% |       0.0010 |
|      10 |          20 |         4.30 |     0.6843 |     71.00% |       0.0001 |
|========================================================================|

ii =

    17


ans =

   0.9000
```

Figure 26 Statistics and accuracy of the trained neural network.

Figure 27, represents how the single test image runs looked and the categorization of a 'defective' image. The green 'x' above the picture denotes it as a defective image. If it was a non-defective image it would show up as a red 'O'. It shows the image along with the designation of 'X' or 'O'.
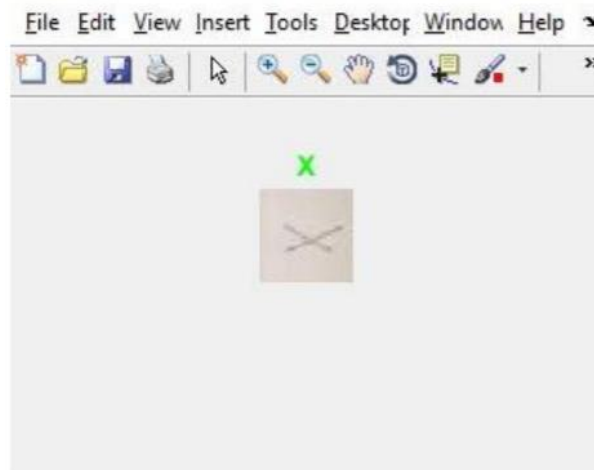

Figure 27 Single image test of the neural network accuracy (displaying defective categorization).

The following table, Table 4, represents 5 consecutive runs and the accuracy associated with each. The average accuracy of these five runs is 89%, backing up the displayed accuracy of the set images in Figure 26.

Table 4 Shows the accuracy in percentage of five consecutive test runs.

Test Run #	Accuracy [%]
1	0.85
2	0.87
3	0.93
4	0.89
5	0.91

This can be expanded to simulate a specific railroad defect, as it looks for a signature in the image to categorize it as necessary. For example, if the rail itself has a crack, the program is trained to recognize the characteristics of that crack. It will then categorize the image by matching it with the training images and flagging it as a defect. This can be applied to other defects along the railway and not only be able to characterize it as a defect, but it should be able to characterize it by the type of defect as well.

CONCLUSION

The dynamics of the quadrotor were derived using Newton's laws, as well as Euler's laws. After linearizing the system around the hover position the PID controllers for the states were able to be adequately designed to get a reasonable starting point for the PID controllers' gains. The Ziegler-Nichols technique accompanied with more trial and error was used to tune the gains to get a useable response from the system. These were later finely tuned more for better line following traits. Using the Alexnet neural network gave good initial results for a good starting point to detecting true defects on a railroad track. Future changes to this project will be explained to show the proposed progression of this project. Using new and more complex controllers to increase the response of the quadcopter for better line following results. A more in-depth architecture of the quadrotor's system and environment to account for conducting flights outdoors. Different and more complicated techniques for line following based on vision. The sensors used to detect defects will either need to change or an algorithm to detect actual defects in the railroad tracks and more accurate neural networks will be used to more accurately categorize the images.

Future Work

There are some things to be add in for future work to fully address the problem of autonomously detecting defects in a railroad track in a real worl environment. Further investigation and experiments into the different controllers that can be used, including sliding mode and H-infinity controllers. These may be able to enhance the response of the system. Different following algorithms will also be investigated further to attempt to drive the path following error closer to zero. A quadcopter that can utilize a processor with greater performance specs giving the capability for more complex algorithms and onboard processing. This leads to the ability to

changes in the defect detection strategy of this project. Other neural networks will be used in combination with a quadcopter that carries a higher processing capacity to try to increase the accuracy and still handle the processing in adequate time.

LIST OF REFERENCES

[1] P. Babenko, "Visual inspection of railroad tracks(Unpublished master's thesis).," Orlando, 2009.

[2] ENSCO, "ENSCO Track Inspection Technologies," [Online]. Available: https://www.ensco.com/rail/track-inspection-technologies. [Accessed 10 5 2019].

[3] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti and S. Longhi, "A vision-based guidance system for UAV navigation and safe landing using natural landmarks," *Journal of intelligent and robotic systems,* vol. 57, no. 1-4, p. 233, 2010.

[4] Y. Wu, Y. Qin, Z. Wang and L. Jia, "A UAV-Based Visual Inspection Method for," *Applied Sciences,* vol. 8, no. 7, p. 1028, 2018.

[5] C. A. Amadi, "Design and Implementation of Model Predictive Control on Pixhawk Flight Controller.," Stellenbosch University, 2018.

[6] R. A. García, F. R. Rubio and M. G. Ortega, "Robust PID Control of the Quadrotor Helicopter," *IFAC Conference on Advances in PID,* vol. 2, no. 5, pp. 229-334, 2012.

[7] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *International Conference on Robotics and Automation*, 2005.

[8] S. Bouabdallah, A. Noth and R. Siegwart, "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[9] R. Ji and J. Ma, "Mathematical Modeling and Analysis of a Quadrotor with Tilting Propellers," in *Chinese Control Conference (CCC)*, Wuhan, 2018.

[10] E. Kuantama, I. Tarca and R. Tarca, "Feedback Linearization LQR Control for Quadcopter Position," in *5th International Conference on Control, Decision*, Thessaloniki, 2018.

[11] F. Sabatino, "Quadrotor control: modeling,nonlinear control design, and simulation," Stockholm, 2015.

[12] V. Choudhary, MEMS : fundamental technology and applications, Boca Raton: CRC Press,Taylor & Francis Group, 2013.

[13] J. G. Korvink and O. Paul, MEMS : a practical guide to design, analysis, and applications, W. Andrew Pub.: Norwick, 2006.

[14] G. Thomas, "Overview of nondestructive evaluation technologies," *International Society for Optics,* vol. 2459, pp. 5-10, 1995.

[15] M. Shah, "Automated visual inspection/detection of Railroad track," 2010.

[16] Y. Li, H. Trinh, N. Haas, C. Otto and S. Pankanti, "Rail component detection, optimization, and assessment for automatic rail track inspection," *IEEE Transactions on Intelligent Transportation Systems,* vol. 15, no. 2, pp. 760-770, 2014.

[17] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *Journal of Field Robotics,* vol. 29, no. 2, pp. 315-378, 2012.

[18] M. K. Haider, I. Abdel-Qader and M. Alsafasfeh, "A Robust Sparse System Identification via Integrating an Individualized Time-Varying Stepsize Adjustments," in *IEEE Jordan International Joint Conference on Electrical Engineering* , 2019.

[19] X. Yang, X. Liu and S. Yin, "Robust Identification of Nonlinear Systems With Missing Observations: The Case of State-Space Model Structure," *IEEE Transactions on Industrial Informatics,* vol. 15, no. 5, pp. 2763-2774, 2019.

[20] S. Khatoon, M. Shahid, Ibraheem and H. Chaudhary, "Dynamic modeling and stabilization of quadrotor using PID controller," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, New Delhi, 2014.

[21] P. Wang, Z. Man, Z. Cao and J. Zheng, "Dynamics Modelling and Linear Control," in *2016 International Conference on Advanced Mechatronic Systems*, Melbourne, Australia, 2016.

[22] T. Madani and A. Benallegue, "Backstepping Control for a Quadrotor Helicopter. In Intelligent Robots and Systems," in *IEEE/RSJ International*, 2006.

[23] R. Xu and U. Ozguner, " Sliding mode control of a quadrotor helicopter. In," in *45th IEEE Conference*, 2006.

[24] D. Lee, H. J. Kim and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of Control, Automation and Systems,* vol. 7, no. 3, pp. 419-428, 2009.

[25] S. Zhao, A. Honglei and Z. Daibing, "A New Feedback Linearization LQR Control for Attitude of Quadrotor," in *International Conference on Control, Automation, Robotics and Vision*, 2014.

[26] E. Paiva, J. Soto, J. Salinas and W. Ipanaque, "Modeling, Simulation and Implementation of a modified PID controller for stabilizing a Quadcopter," in *2016 IEEE International Conference on Automatica (ICA-ACCA) Automatica (ICA-ACCA), IEEE International Conference*, Shah Alam, 2016.

[27] K. J. Astrom and R. M. Murray, Feedback Systems: An Introduction for Scientists and Engineers, Princeton: Princeton University Press, 2009.

[28] O. Surinta and S. Khruahong, "Tracking People and Objects with an Autonomous Unmanned Aerial Vehicle using Face and Color Detection," in *Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)*, Nan, 2019.

[29] T. Susanto, R. Mardiyanto and D. Purwanto, "Development of Underwater Object Detection Method Base on Color Feature," in *International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, Surabaya, 2018.

[30] H. Su, "Vanishing Points in Road Recognition: A Review," in *International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, 2018.

[31] Y. Su, Y. Zhang, T. Lu, J. Yang and H. Kong, "Vanishing Point Constrained Lane Detection With a Stereo Camera," *IEEE Transactions on Intelligent Transportation Systems,* vol. 19, no. 8, pp. 2739-2744, 2018.

[32] T. Nguyen, G. K. Mann and R. Gosine, "Vision-based path-following control of quadrotor aerial vehicle.," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, 2014.

[33] M. M. KASSIR, M. PALHANG and M. R. AHMADZADEH, "Qualitative vision-based navigation based on sloped funnel lane concept," *Computer Science,* 2018.

[34] R. Al-Jarrah, M. Al-Jarrah and H. Roth, "A Novel Edge Detection Algorithm for Mobile Robot Path Planning," *Journal of Robotics,* vol. 2018, p. 12, 2018.

[35] Z. Wang, M. IEEE, G. Cheng, J. Zheng and S. M. IEEE, "Road Edge Detection in All Weather and Illumination via Driving Video Mining," *IEEE TRANSACTIONS ON INTELLIGENT VEHICLES,* vol. 4, no. 2, pp. 232-243, 2019.

[36] Z. Hu, C. Deng, Y. Shao and C. Wang, "Edge Detection Method based on Lifting B-Spline Dyadic Wavelet," *International Journal of Performability Engineering,* vol. 15, no. 5, pp. 1472-1481, 2019.

[37] M. S. N. Devi and S. Santhi, "Improved Edge Detection Methods in OCT Images using a Hybrid Framework based on CGWO Algorithm," in *International Conference on Communication and Signal Processing (ICCSP)*, Chennai, 2019.

[38] D. Mel, K. A. Stol, J. Mills and B. R. Eastwood, "Vision based object path following on a quadcopter for GPS-denied environments.," in *Internation Conference of Unmanned Aircraft Systems (ICUAS)*, Miami, 2017.

[39] H. Ma, Y. Min, C. Yin, T. Chengb, B. Xiao, B. Yue and X. Li, "A Real Time Detection Method of Track Fasteners Missing of Railway based on Machine Vision," in *International Journal of Performability Engineering*, 2018.

[40] L. Zhuang, L. Wang, Z. Zhang and K. Tsui, "Automated vision inspection of rail surface cracks: A double-layer data-driven framework.," *Transportation Research Part C: Emerging Technologies,* pp. 258-287, 2018.

[41] A. Brandao, F. Martins and H. Soneguetti, "A Vision-based Line Following Strategy for an Autonomous UAV.," in *International Conference of Informatics in Control*, Colmar & Alsace, 2015.

[42] D. Lee, H. J. Kim and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of Control, Automation and Systems,* vol. 7, no. 3, pp. 419-428, 2009.