EFFECTS OF ACTIVE LANDING GEAR ON THE ATTITUDE DYNAMICS OF A QUADROTOR

A MASTER'S THESIS

IN

MECHATRONICS ENGINEERING

ATILIM UNIVERSITY

BY

MEHMET YILDIZ

OCTOBER 2015

EFFECTS OF ACTIVE LANDING GEAR ON THE ATTITUDE DYNAMICS OF A QUADROTOR

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

MEHMET YILDIZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF MECHATRONICS ENGINEERING

OCTOBER 2015

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

_____

Prof. Dr. İbrahim AKMAN

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Prof. Dr. Abdulkadir ERDEN

Head of Department

This is to certify that we have read the thesis "Effects of Active Landing Gear on the Attitude Dynamics of a Quadrotor" submitted by Mehmet Yıldız and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____	_____
Asst. Prof. Dr. Bülent İRFANOĞLU	Asst. Prof. Dr. Kutluk Bilge ARIKAN
Co-Supervisor	Supervisor

Examining Committee Members

Asst. Prof. Dr. Hakan Tora (ATU, E.E.E. Dept.) _____

Asst. Prof. Dr. Bülent İrfanoğlu (ATU, MECE. Dept.) _____

Asst. Prof. Dr. Kutluk Bilge Arıkan (ATU, MECE. Dept.) _____

Asst. Prof. Dr. Can Ulaş Doğruer (HU, M.E. Dept.) _____

Asst. Prof. Dr. Ali Emre Turgut (METU, M.E. Dept.) _____

Date: (21-10-2015)

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Mehmet YILDIZ

Signature:

ABSTRACT

EFFECTS OF ACTIVE LANDING GEAR ON THE ATTITUDE DYNAMICS OF A QUADROTOR

Yıldız, Mehmet

M.S, Mechatronics Engineering Department

Supervisor: Asst. Prof. Dr. Kutluk Bilge Arıkan

Co-Supervisor: Asst. Prof. Dr. Bülent İrfanoğlu

October 2015, 114 pages

In this thesis, effects of the active landing gears which are used in most of the commercial multirotor systems, on attitude dynamics are investigated. Utilization for the purpose of stability assistance, steering and the agility enhancement is discussed. A Quadrotor test system is constructed by using a commercial quadrotor frame, actuators, drivers and an open source embedded flight controller. The Quadrotor system with active landing gears is modelled on Matlab/Simulink environment. PID controller which is used in the embedded flight controller is designed to control attitude dynamics. PID controller is tuned on the test bench and tuned parameters are applied to the identified model of the Quadrotor. This model is used to optimize the controller parameters. For the active landing gear system, a simple servo motor actuated mechanism is used and this mechanism is applied on the Quadrotor roll axis. The positions of the gears are controlled by embedded flight controller as well. Experiments show that active landing gears can be used to improve the stability, to steer platform in horizontal plane and to improve the agility by applying an additional moment about the roll axis.

Keywords: Quadrotor, Flying Robotics, Multirotor, Retractable Landing Gear, Multiwii, CleanFlight, PID Controller, Manipulator

ÖZ

AKTİF İNİŞ TAKIMININ DÖRT PERVANELİ HAVA ARACI DİNAMİĞİ ÜZERİNDEKİ ETKİLERİ

Yıldız, Mehmet

Yüksek Lisans, Mekatronik Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Kutluk Bilge Arıkan

Ortak Tez Yöneticisi: Yrd. Doç. Dr. Bülent İrfanoğlu

Ekim 2015, 114 sayfa

Bu tez kapsamında, birçok ticari multirotor sisteminde kullanılmakta olan açılır-kapanır aktif iniş takımlarının multirotor sistemlerinin yönelim dinamikleri üzerindeki etkisini incelenmektedir. Sistemin stabilizasyonuna yardımı, yönlendirme ve çevikliğini geliştirme amacı ile kullanımı tartışılmıştır. Tez kapsamında, ticari bir Quadrotor şasesi, eyleyicileri, sürücüleri ve açık kod kaynaklı bir gömülü uçuş kontrolcü kartı kullanılarak bir Quadrotor test sistemi oluşturulmuştur. Quadrotor test sistemi aktif iniş takımı ile birlikte Matlab/Simulink ortamında modellenmekte olup, yönelim dinamikleri kontrolü için gömülü uçuş kontrolcüsündeki PID kontrolcüsü kullanılmaktadır. Bu kontrolcü test düzeneği üzerinde ayarlanmaktadır ve ayarlanan kontrolcü değerleri oluşturulan Quadrotor modeli üzerinde uygulanmaktadır. Bu model kontrolcü parametrelerini optimize etmek için kullanılmaktadır. Aktif iniş takımı için basit bir servo motor mekanizması kullanılmaktadır ve Quadrotor sisteminin yuvarlanma eksenine etki etmektedir. Servo motorların pozisyonlarının kontrolü gömülü uçuş kontrol kartı üzerinden yapılmaktadır. Yapılan testler, aktif iniş takımının, sistem stabilizasyonunu geliştirmede, yönelimde ve çevikliği geliştirmede kullanabileceğini göstermektedir.

Anahtar Kelimeler: Quadrotor, Uçan Robotlar, Multirotor, Açılır-Kapanır İniş Takımı, Multiwii, CleanFlight, PID Kontrolcü, Manipülatör

To My Mother, Sister, Aunt and Nephews

ACKNOWLEDGMENTS

I express sincere appreciation to my supervisor Asst. Prof. Dr. Kutluk Bilge Arıkan for his guidance and insight throughout the research. Thanks also go to my co-supervisor Asst. Prof. Dr. Bülent İrfanoğlu. I owe thanks to Mete Aydemir, Doğanç Küçük, Berk Yavru and Robonik Mekatronik Technologies Company Family for their help during the experiments with their endless help. Finally, I offer sincere to my Sister, Aunt and Duygu thanks for them continuous support and patience during this period.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ESC	-	Electronic Speed Controller
I/O	-	Input/Output
PC	-	Personal Computer
PID	-	Proportional Integral Derivative
PWM	-	Pulse Width Modulation
RTWT	-	Real Time Windows Target
ROS	-	Robotic Operating System
UAV	-	Unmanned Aerial Vehicle
VTOL	-	Vertical Take-off and Landing
ARCAS	-	Aerial Robotics Cooperative Assembly System
COG	-	Center of Gravity
LQR	-	Linear Quadratic Regulator
ADRC	-	Active Disturbance Rejection Control
DC	-	Direct Current
DAQ	-	Data Acquisition
MEMS	-	Microelectromechanical Systems
IMU	-	Inertial Measurement Unit
DOF	-	Degree of Freedom
RC	-	Radio Control
CAD	-	Computer Aided Drawing
GUI	-	Graphical User Interface
LI-PO	-	Lithium Polymer
GPL	-	General Public License

NOMENCLATURE

The following are the list of variables used in this thesis:

ϕ, θ, ψ - Euler Angles (rad)

$\dot{\eta}$- Euler Angular Rate Matrix

$\dot{\phi}, \dot{\theta}, \dot{\psi}$ - Euler Angular Rates (rad/s)

ω- Angular Body Rate Matrix

$\omega_{x,y,z}$- Angular Body Rate Matrix Components (rad/s)

R- Rotation Matrix

J- Inertia Matrix

$J_{xx,xy,xz,zx,zy,zz,yx,yy,yz}$- Inertia Matrix Elements ($kg.m^2$)

M_G- Gyroscopic Moments due to Rotor's Inertia

J_r- Rotor's Inertia ($kg.m^2$)

Ω_r- Rotor's Relative Speed (rad/s)

M_B- Moments Acting on the Quadrotor in the Body Frame Matrix

$M_{x,y,z}$- Moments Matrix Components (N.m)

$F_{1,2,3,4,z}$- Thrust Force (N)

$M_{1,2,3,4}$- Drag Force (N)

K_f- Thrust Factor

K_m- Drag Factor

$\Omega_{1,2,3,4}$- Angular Rotor Speed (rad/s)

ρ- Air Density (kg/m^3)

A- Blade Area (m^2)

C_T, C_D- Aerodynamic Coefficients

r_b- Radius of Blade (m)

g- Gravitational Acceleration (m/s^2)

h- Distance Between O and O_1 points (O and O_2 points) (m)

L_m- Length of Landing Gear Rods (m)

m- Mass of Gears (kg)

δ_1, δ_2- Landing Gears Angles (deg)

M_{xg}- Moment Generated by Landing Gears (Nm)

M_{xp}- Moment Generated by Propulsion Units (Nm)

r- The Distance between the Earth Frame and the Body Frame Matrix

F_B- Force Acting on the Quadrotor (non-gravitational Forces) Matrix

x,y,z- Linear Displacement of Body in x, y, z Directions (m)

$\dot{x}, \dot{y}, \dot{z}$- Linear Velocity of Body in x, y, z Directions (m/s)

$\ddot{x}, \ddot{y}, \ddot{z}$- Liner Acceleration of Body in x, y, z Directions (m/s^2)

v- The Effective Input Voltage (V)

v_{R_m}- The Resistor Voltage (V)

v_{L_m}- The Inductor Voltage (V)

v_{e_m}- The Motor Back-Emf (Electro Motor Force) Voltage (V/ (rad/s))

R_m- Motor's Resistance (Ω)

$\mathbb{L}_m$- Motor's Inductance (L)

i_a- Motor's Armature current (A)

K_{mot}- Motor Torque Constant (Nm/A)

Ω_i- i_{th} Motor Angular Velocity (rad/s)

J_m- The Rotor Shaft Inertia ($kg.m^2$)

T_m- The Rotor Torque

T_{load}- The Load Torque which is generated from the Propeller

U- Control Signal Vector

$U_{1,2,3,4}$- Control Signal Vector Components

M_B- Total Moment's Vector

Kp- PID Controller Proportional Gain

Ki- PID Controller Integral Gain

Kd- PID Controller Derivative Gain

Kp_Level- PID Controller Level Proportional Gain

J_o- Inertia on O point ($kg.m^2$)

M- Quadrotor Body Mass (kg)

b- Viscous Damping (N.m.s)

L- Length from Rotor to COG, C Point (m)

h_c- Length from O Point to COG, C Point (m)

M_x- Total Moment Produced by Propellers

CHAPTER 1
INTRODUCTION

In recent years, with the development of technologies such as electronics, microcontrollers, motors, drivers, sensors and software, autonomous flying robotic systems have gained huge commercial potential and became worldwide available; also this topic becomes more popular for researches day by day. Quadrotor system is the most popular system in flying robotics area (Figure 1). It has four rotating blades and by using those blades, it changes its attitude and altitude. Those systems have advantages as simple construction, small sized, flight flexibility, high maneuverability, capability of vertical take-off and landing, and hover stably compared to other flight robotic system such as fixed-wing and helicopter. Nowadays, quadrotor system is used in many areas. Usage of areas can be classified into two military and civil applications. Briefly, military operations are focused on security and intelligence and civil applications are focused on photography, area mapping and search-rescue. Today, the most popular and known quadrotor system is Phantom series which is produced by a Chinese company DJI [1] (Figure 1). This product has been a pioneer in spreading the quadrotor system in civil application is very quickly.

Figure 1: DJI Phantom Quadrotor [1]

These products are developed day by day in military and civil area. Generally, those systems bring out their duties successfully in applications that have mentioned

above. But still some problems such as flight time, pay load, obstacle avoidance, GPS denied environment navigation, interaction with environmental objects and advanced fail safe control algorithm to minimize human error are waiting to be solved by researchers and developers. Also researchers and developers started to focus on using flying robotic system such a flying manipulator system to apply some environmental interaction tasks for indeterminable and dangerous areas. By integrating manipulator systems such as robot arm, griping mechanism, and etc. on the flying robotics, it becomes a flying manipulator system. Some good study on this issue is demonstrated by GRASP laboratory of Pennsylvania University [2] and Flying Machine Area from Zürich University [3]. Other studies are mentioned in Literature Survey section.

The idea of using external manipulator mechanisms which are integrated on quadrotor system to assist steering and stability of quadrotor's attitude dynamics besides environmental interaction task is the starting point of this study. Actually, this idea may sound very different and thought to use it only in the laboratory conditions, but it is interesting to note that today a lot of commercial products use external manipulators which are called retractable landing gears. They are used for a wider viewing angle in photography applications. In Figure 2, a commercial product which called S900 [4] can be seen. In this thesis, effects of active landing gears on orientation dynamics on one axis of Quadrotor are analyzed and stability assistance of active landing gears is investigated. In this study, active landing gears are used such as manipulators. The active landing gears can be used in the right directions on the system's orientation actively and it can affect the system controller positively. If a positive contribution on controller is observed, that idea can be applied to aerial multirotor systems which have manipulator system. In the literature, stability assistance of manipulators on aerial multirotor platforms topics is huge lack of information. With this study, it is planned to remedy this lack of information.

Figure 2: S900 System [4]

To investigate the effect of active landing gears on quadrotor attitude dynamics, a quadrotor test setup is constructed by using a commercial quadrotor frame kit with an open source embedded flight controller card which is called NAZE32 [5] based on CleanFlight software projects [6]. Also a simple servo motor mechanism applied on Quadrotor one axis for manipulating. In physical section, all components of test system are given.

Test Quadrotor is modelled on Matlab/Simulink environment and PID controller which is used in embedded flight controller is designed to control attitude dynamics of Quadrotor. PID controller is tuned on test bench system and tuned parameters are applied on the model. Uncertain parameters are identified experimentally. Then, controller parameters are optimized using the identified model and applied on the real system.

For active landing gears, three modes are determined to study their effects on Quadrotor's attitude dynamics. In first mode, active landing gears create a counter moment to improve the stability. A regulatory feedback controller is designed to control the position of active landing gears. This mode is called assistance mode. In the second mode, by moving the gears position to the left and right, system can be steered in horizontal plane. Position of the gears can be controlled via Transmitter directly. This mode is called steering mode. The last one is the improved agility mode; it is the opposite of the assistance mode to improve the agility. Those three modes are examined by flight tests and results are presented in Chapter 6.

1.1 Aim and Scope of Thesis

Multicopters have been treated as flying cameras. They have been used for aerial photography, video streaming, mapping, etc. Whereas recent studies show that they can be utilized for many different purposes by equipping various sensors and actuators. Flying manipulator concept is on the recent channels where multicopters have much to do. Literature survey section presents flying platforms that include robot manipulators, arms, etc. Future trends of multicopters and recent literature make us carry out research on multicopters with actuated limbs. Coupled dynamics of robotic sub system with multicopter must be considered during design. This robotic sub assembly is designed to be used for performing a certain task. However, it may also be utilized to serve as the part of guidance system. This study is a component of the ongoing research at the *Flying Robotics and Robotic Vehicles Laboratory*. In this thesis, effect of active landing gears on attitude dynamics is investigated to utilize it for the purpose of steering and stability assistance. Active landing gear is available in most of the commercial systems. In addition it is one of the simplest robot arms that can be placed on a multicopter. The motivating research question is "Can we can utilize it for the steering and/or control purposes?"

Many alternatives exist as the controller hardware and software in the market. Some of them provide practical issues for the end users. They have built-in control algorithms and provide change of settings and tuning through user interfaces. Arducopter, KK-Multicopter, GAUI, Naze32 and Multiwii are typical and popular open source projects and products that are available for multicopters. In this thesis Naze32 board is utilized. Active landing gear is also controlled by using Naze32 controller board.

Mathematical models are developed for the quadcopter platform employed in the thesis. Utilization of active landing gear is also simulated using these models. Physical platform and its components are presented in the related sections. Besides simulations, real tests are performed on the physical quadcopter.

1.2 Outline of the Thesis

The literature survey is mentioned in Chapter 2. Mathematical modeling which is used to design a controller and simulate the system is given in Chapter 3. After mathematical model is obtained, control system design and simulations are performed and it is mentioned in Chapter 4. Physical system is described and the components are introduced in Chapter 5. Designed control system is implemented on real system and these experiments are in Chapter 6. In the end, discussion and conclusion of thesis are stated in Chapter 7.

CHAPTER 2

LITERATURE SURVEY

Nowadays, Unmanned Aerial Vehicles (UAV) which are designed for military applications (observation, reconnaissance, border security, etc.) and recovery operations (fire, earthquake, flood, etc.) become very popular research topics for researchers. In the field of unmanned aerial vehicles (UAV), long-range aircraft and helicopters which can remain suspended in the air are the most preferred platforms by researchers. Along with technological developments researchers start to focus on aerial multirotor systems which can take-off and land vertically (VTOL). Aerial multirotor systems have important benefits. These systems can be scaled down to small sizes and can operate indoor environment and with the smaller size comes increased agility and the ability to adapt to the environment [7], and also these systems don't need a landing runway, only a small area is enough for take-off, landing, and operations. These systems are named according to the number of engines they have. The most popular ones are Quadrotor, Hexarotor, and Octorotor (Figure 3). Operating principles are the same, the biggest difference between is payload capacity. Also these systems in research are divided into two groups such as Indoor [9], [10] and Outdoor [11], [12].

Figure 3: Most Common Aerial Multirotor Types [8]

Many research groups have started the research on these systems and constructing Quadrotor flying systems as robotics research tools [13], [14], and [15]. Many

research groups have made many researches on modeling and control of these systems [16], [17], [18], and [19]. As a result of these researches, successful results are obtained in the control and modeling of these systems nowadays [7]. Along with obtaining good results, these systems combined with manipulators and began to interact with the external environment [11], [12], [20], [21] and [26]. The reason of the interact with the external environment is that especially for surveillance applications where system's objectives are limited to only "look" and "search", there is no interaction between physical environment. In order to interact with the external environment, manipulator mechanisms are combined with aerial multirotor systems (Figure 4 and 5).

Figure 4: A Quadrotor as a Serial Chain Manipulator [21]

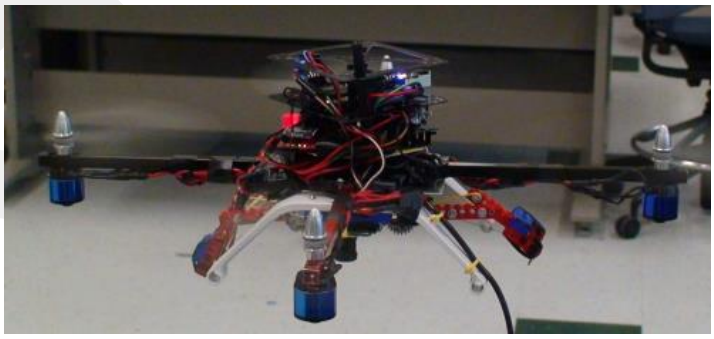

Figure 5: Quadrotor with a Griper Mechanism [26]

Actually, Aerial Multirotor systems first became popular, researchers work on such a systems with UAV helicopters [22], [23]. One of the UAV helicopters which have a

manipulation mechanism such as a robot arm, designed by Aerial Robotics Cooperative Assembly System (ARCAS) [24] is shown in Figure 6.

Figure 6: Aerial Robotics Cooperative Assembly System's Preliminary Design [24]

Combining the manipulator mechanisms with aerial multirotor systems brings new control problems. When an aerial multirotor system tries to grip an object in Figure 7, the dynamics of the system with external payloads will change, and this change affects the center of gravity (COG) of the system .Due to this change, system can be unstable and it is not a desired situation for an aerial multirotor systems.

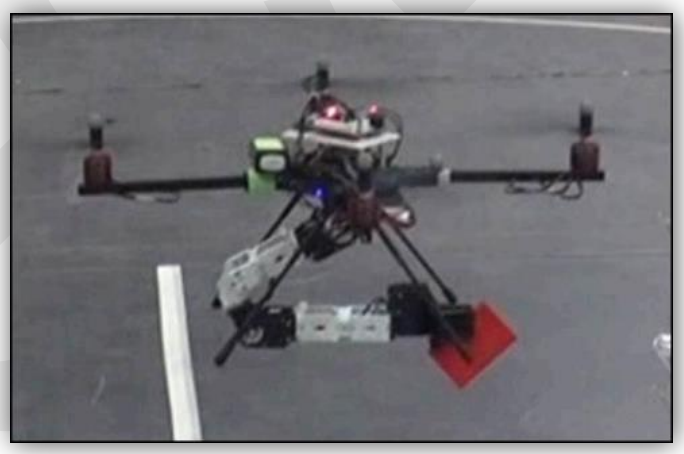

Figure 7: Quadrotor Gripping and Carrying Object [11]

In some studies, that problem is handled [20], [25], and [37]. In [25], [37] changes in the COG of the system are calculated by using a set of dynamic equations common in robotic manipulator modeling. In [20], the inertia of the payload is estimated and

adapts to the system dynamics to improve controller performance. Other studies assume the COG of the system fix in all situations and try to develop controller to compensate the changing the COG and the reaction forces that characterizing the forces and torques experienced by the system body during both flight and manipulation [11], [12], [26]. In these studies; that common result is emphasized; changes in the dynamic of the system and reaction forces that make the system less stable for a Quadrotor system with manipulator and payload. Contrary to general opinion, if manipulator system can use proper with payloads, positive effects on controller can be observed [27], [28].

In many cases, different control techniques are implemented to control a Quadrotor system. PID controller is the most widely used controller [20], [21], [22], [26], and [29]. As well as, some different control methods are used to control Quadrotor such as LQR [30], [31], Nonlinear Controllers [23], [32], Back Stepping Controller [30], [33] and Active Disturbance Rejection Control (ADRC) [34], [35]. In most of study, system mathematical model is obtained by using Newton-Euler Methods [26], [27] also Euler-Lagrange methods can be used to obtained system mathematical models [37].

In many studies, commercial systems are used to build and test Quadrotor [20], [21], [26], and [37]. The most used commercial systems are Mikrokopter [8], Ascending Technologies [47], GAUI [48] and Arducopter [50]. Also, custom build frames are used in some studies [11], [27]. Almost all systems are similar in structure to each other, and these systems are offered all in one solutions. In almost all systems, brushless DC motors, fast PWM electronic speed controller (ESC) and plastic propellers are used to propulsion systems. Today lots of motor, ESC and propeller is available for different manners, but some ESCs have special communication protocol such as Mikrokopter and Ascending Technologies. In most of studies, embedded controllers which can be directly programed on MATLAB/SIMULINK software program are used [20], [37]. The most popular controller embedded card is Mastermind from Ascending Technologies [47] and Gumstix [49] which is some researchers prefer to work with MATLAB/RTWT and MATLAB compatible DAQ card [9]. Other low cost embedded solutions can be used to control the system [11], [12], [38] and [39]. In many research, low cost MEMS sensor technologies are used

to measure attitude and altitude of the system [11], [12], and [38]. Especially built-in IMU are used such as Mastermind, Mikrokopter, GAUI, Arducopter, and Multiwii controller cards. On the other hand, commercial ready to use IMU units are used [9], [39]. Indoor applications, beside the IMU unit, some vision sensor units are used such as VICON [51], VZ400 [52], and OptiTrack V100:R2 [53] to use indoor localization. To meet the power needs of the system, Li-Po batteries which have high current capacities are used. Almost all researchers use MATLAB/SIMULINK computing environment to develop controller algorithm, simulation, and collecting data. Also, MATLAB/SIMULINK *ROS (Robot Operating* System) libraries [54] are very popular in research area. Occasional, Labview can be used in studies.

CHAPTER 3

MATHEMATICAL MODELLING

In this thesis, attitude dynamics of a quadrotor are considered. Therefore the kinematics and dynamics model of a quadrotor are derived by using Newton-Euler formalism with the following assumptions:

- The structure and landing gears are rigid and symmetrical.
- The center of gravity of the quadrotor coincides with the body fixed frame origin.
- The propellers and rotors are rigid.
- The center of gravity of active landing gears is at the end of the rods.
- Active landing gears create moment only one axis but in two directions.
- Thrust and drag are proportional to the square of rotor's speed.
- Landing gears axis and Quadrotor roll axis are coincident.
- Aerodynamic effects (Drag Forces and Drag Moments) are neglected.

After obtaining the kinematics and dynamics models of the quadrotor, landing gears and rotor dynamics are obtained to investigate the effects on the quadrotor body. At the end of this chapter, a state space model for the quadrotor system is given.

3.1 Kinematic Model

Before starting the quadrotor modeling, the coordinate frame should be introduced. There are two coordinate frames. First one is the Earth reference frame that are fixed on a specific place at ground level with N, E and D axes where the axes point to the North, East and Downwards. Second one is the body frame that is at the center of gravity the quadrotor body with x, y, z where x-axis pointing towards propeller 1, y-

axis pointing towards propeller 2 and z-axis pointing to the ground. Frames and axes are represented in Figure 8.

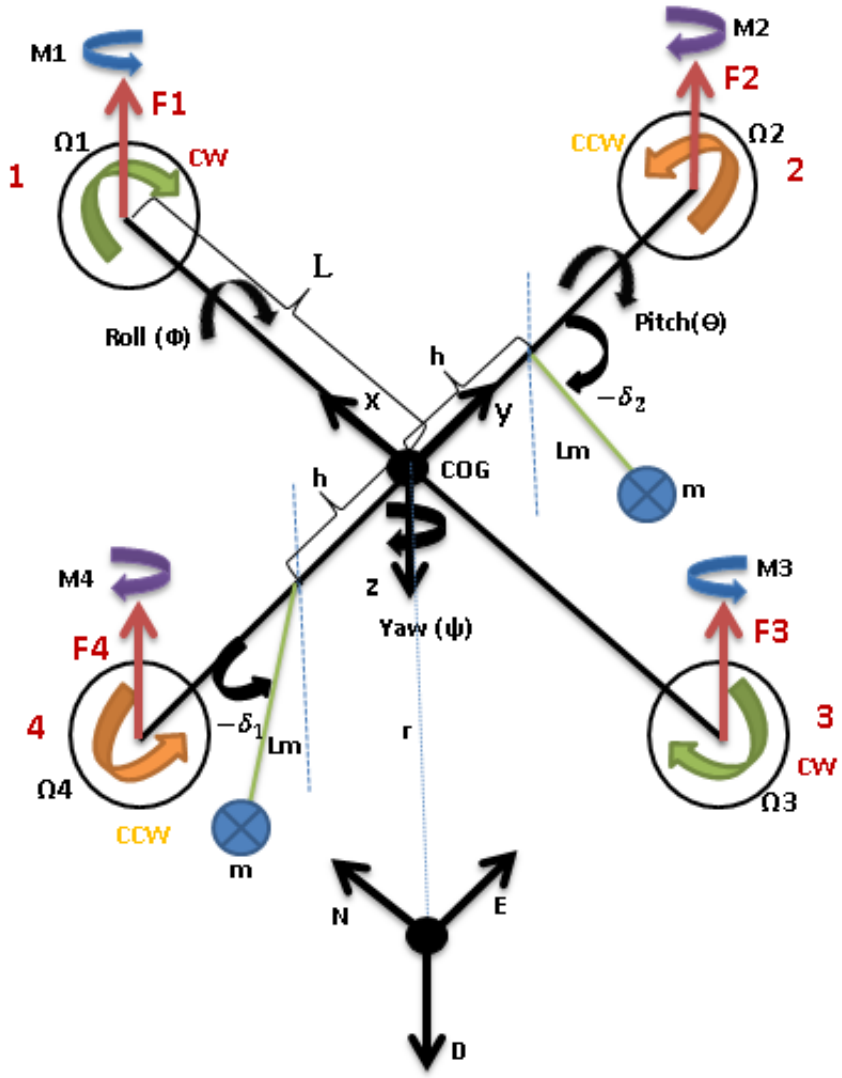

Figure 8: Quadrotor Reference Frame

Attitude of the quadrotor is described using roll, pitch and yaw angles; namely ϕ, θ, and ψ where rotations are about the x, y, and z-axes, respectively. The distance between the Earth frame and the body frame describes the absolute position of the center of mass of the quadrotor $r = [x\ y\ z]^T$ [33]. The rotation R from the body frame to the inertial frame describes the orientation of the quadrotor. It is possible to obtain the rotation matrix R with the following equations [55].

(z, y, x) $$R = (R_i^b)^T = R_3 R_2 R_1 \quad \textbf{(3.1)}$$

ψ Rotation, z $$R_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \textbf{(3.2)}$$

θ Rotation, y $$R_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\emptyset \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad \textbf{(3.3)}$$

ϕ Rotation, x $$R_3 = \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \quad \textbf{(3.4)}$$

Substituting R_1, R_2 and R_3 into (3.1),

$$R_i^b = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\psi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi s\psi & s\phi c\theta \\ c\phi s\theta + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix} \quad \textbf{(3.5)}$$

$$R = (R_i^b)^T = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\theta c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad \textbf{(3.6)}$$

Where c and s denote cos and sin respectively. The rotation matrix R will be used in formulating the dynamics model of the quadrotor, its significance is due to the fact that some states are measured in the body frame (e.g. thrust forces produced by the propellers) while some others are measured in the inertial frame (e.g. the gravitational forces and the quadrotor's position) [40].

Angular body rates, in view of angular body rates components of the quadrotor as illustrated in Figure 8 and the Euler rates that are measured in the inertial frame, can be written as;

$$Anuglar\ body\ rates \quad \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \textbf{(3.7)}$$

$$Euler\ rates\ \dot{\eta} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3.8}$$

A transformation matrix R_r is needed to write state equations of the Euler rates in terms of angular body rates as follow:

$$\omega = R_r \dot{\eta} \tag{3.9}$$

It is possible to obtain the rotation matrix R_r with the following equations [41].

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = R(\dot{\phi}) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R(\phi) R(\dot{\theta}) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R(\phi) R(\theta) R(\dot{\psi}) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{3.10}$$

Note that $R(\phi) = R_3$ in Equation (3.4), $R(\theta) = R_2$ in Equation (3.3). Also $\dot{\phi}, \dot{\theta}\ and\ \dot{\psi}$ are small so$R(\dot{\phi}) = R(\dot{\theta}) = R(\dot{\psi}) = I$, then

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\phi} - sin\theta\dot{\psi} \\ cos\phi\theta + sin\dot{\phi}cos\theta\dot{\psi} \\ -sin\phi\theta + cos\dot{\phi}cos\theta \end{bmatrix} \tag{3.11}$$

$$= \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$R_r = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \quad \textbf{(3.12)}$$

Applying the small angle assumptions around the hover position ($cos\phi \equiv 1, cos\theta \equiv 1\ and\ sin\phi = sin\theta = 0$), R_r can be simplified to an identity matrix I.

$$R_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \textbf{(3.13)}$$

Equations (3.9) can be written again,

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad \textbf{(3.14)}$$

3.2 Dynamics Model

The dynamics model of a quadrotor can be divided into two subsystems;

- Rotational Subsystems (Roll, Pitch and Yaw)
- Translational Subsystem (x & y positions and altitude(z))

The rotational subsystem is fully actuated while the translational subsystem is under actuated [42].

3.2.1 Rotational Equations of Motion

Dynamic equations related with the angular velocities can be derived by using the Newton's Law.

$$J\dot{\omega} + \omega x J\omega + M_G = M_B \quad \textbf{(3.15)}$$

Where:

J; Quadrotor's Inertia Matrix

$$J = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{yx} & J_{yy} & -J_{yz} \\ -J_{zx} & -J_{zy} & J_{zz} \end{bmatrix} \quad \textbf{(3.16)}$$

Due to the symmetry, quadrotor's inertia matrix is diagonal, the off-diagonal elements $J_{xy} = J_{xz} = J_{yx} = J_{yz} = J_{zx} = J_{zy} = 0$

ω; Angular Body Rates = $[p \quad q \quad r]^T = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$

M_G; Gyroscopic Moments due to rotor's inertia = $\omega x [0 \quad 0 \quad J_r \Omega_r]^T$

Gyroscopic torques or moments attempt to align the spin axis of the rotor along the inertial z-axis [42].

J_r; Rotor's Inertia

Ω_r; Rotor's Relative Speed = $\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$

M_B; Moments acting on the quadrotor in the body frame= $[M_x \quad M_y \quad M_z]^T$

The rotational equation of the quadrotor's motion can be written as

$$\begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} x \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} x \begin{bmatrix} 0 \\ 0 \\ J_r \Omega_r \end{bmatrix} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \quad \textbf{(3.17)}$$

3.2.2 Moments Acting on the Quadrotor

The forces and moments acting on the quadrotor are given in Figure 9. Moments acting on the quadrotor are a combination of the aerodynamics forces and moments which are produced by rotors and active landing gears moments which are caused by the movement of the gears in each axis.

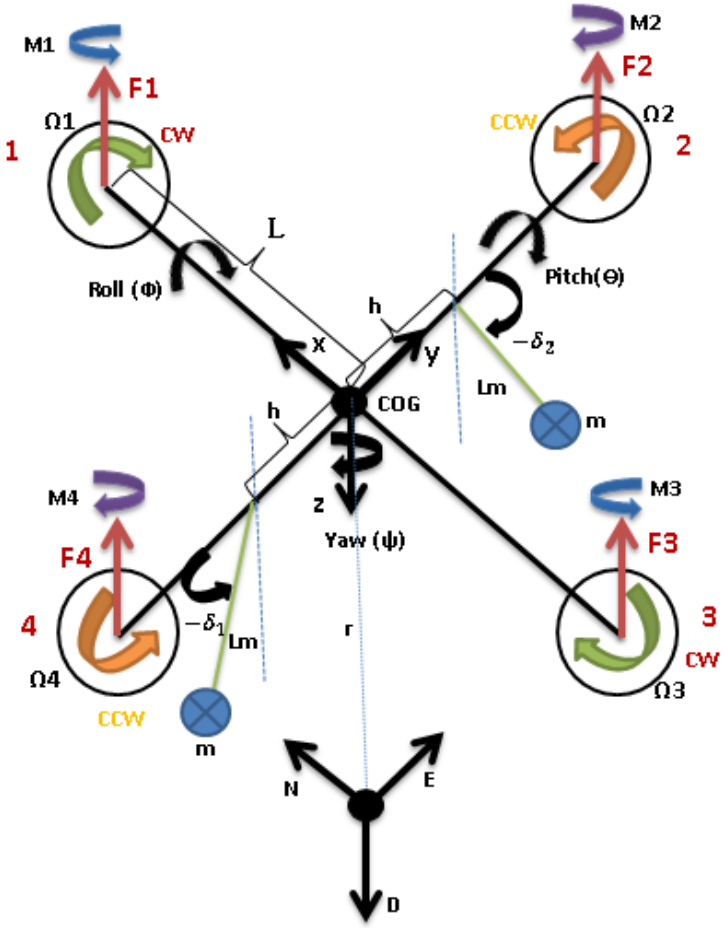

Figure 9: Forces and Moments Acting on Quadrotor

The Aerodynamic Force and Moment

As an effect of rotation, two physical effects which are the aerodynamic forces and moments are produced by a rotor. The aerodynamic force and moment equations which are produced by the i^{th} rotor are given in Equations (3.18) and (3.19) [45].

$$F_i = K_f \Omega_i^2 \tag{3.18}$$

$$M_i = K_m \Omega_i^2 \tag{3.19}$$

Where,

$$K_f = \frac{1}{2}\rho A C_T r^2 \Omega_i^2$$

$$M_i = \frac{1}{2}\rho A C_D r^2 \Omega_i^2$$

ρ;	Air Density
A;	Blade Area
C_T, C_D;	Aerodynamic Coefficients
r_b;	Radius of Blade
Ω_i;	Angular Velocity of Rotor i

The maximum altitude is usually limited, so the air density can be considered constant [42]. For each propeller, the aerodynamic force and moment constants can be determined experimentally.

After determining the forces and moments generated by the propellers, we can identify the moments M_B acting on the quadrotor which is produced by rotors.

x-axis

By using right hand rules, the moment which is produced by rotors about the x-axis can be written as

$$M_x = -F_2 L + F_4 L$$
$$M_x = -\left(K_f \Omega_2^2\right) L + (K_f \Omega_4^2) L$$

$$M_x = L K_f (-\Omega_2^2 + \Omega_4^2) \quad \textbf{(3.20)}$$

y-axis

By using right hand rules, the moment which is produced by rotors about the y-axis can be written as

$$M_y = F_1 L - F_3 L$$
$$M_y = \left(K_f \Omega_1^2\right) L - (K_f \Omega_3^2) L$$
$$M_y = L K_f (\Omega_1^2 - \Omega_3^2) \quad \textbf{(3.21)}$$

z-axis

By using right hand rules, the moment which is produced by rotors about the z-axis can be written as

$$M_z = -M_1 + M_2 - M_3 + M_4$$
$$M_z = (-K_m \Omega_1^2) + (K_m \Omega_2^2) - (K_m \Omega_3^2) + (K_m \Omega_4^2)$$
$$M_z = K_m(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \quad \textbf{(3.22)}$$

l is moment arm length which is between each rotor to the origin of the body frame.

Active Landing Gears Moments

Active landing gears create moment only Roll axis (ϕ) in two directions. The effect on the Pitch (θ) and Yaw (ψ) axes can be neglected Moment acting on Roll (ϕ) axis is given in Figure 10.

Roll Axis (ϕ) around x-Axis

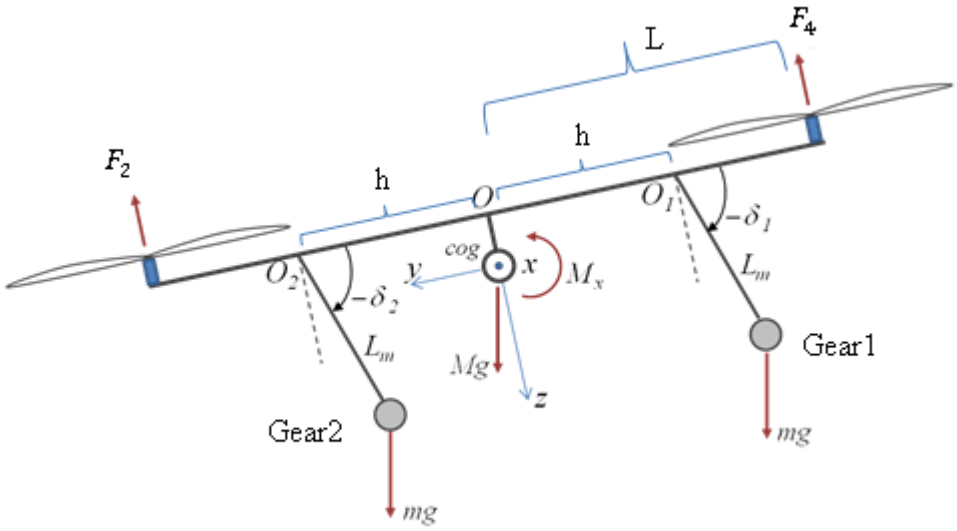

Figure 10: Landing Gear Moments on Roll Axis

$$M_{Gear1} = -(h - L_m cos\delta_1)mgcos\phi \; ; \; M_{Gear2} = (h + L_m cos\delta_2)mgcos\phi \tag{3.23}$$

$$M_{xg} \approx \sim(h - L_m cos\delta_2)mgcos\phi - (h + L_m cos\delta_1)mgcos\phi \tag{3.24}$$

Where;

h; the distance between O and O_1 points (O *and* O_2 points)

L_m; length of landing gear rods

m; mass of gears

δ_1, δ_2; landing gears angles

M_{xg}; the moment generated by landing gears

Combining Equations (3.20) – (3.24), total moment vector can be obtained;

$$M_B = \begin{bmatrix} lK_f(-\Omega_2^2 + \Omega_4^2) + M_{xg} \\ lK_f(\Omega_1^2 - \Omega_3^2) \\ K_m(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix} \tag{3.25}$$

3.3.3 Translational Equations of Motion

By using Newton's second law and Earth inertial frame, the translational equations of motion are obtained [42],

$$M\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ Mg \end{bmatrix} + RF_B \tag{3.26}$$

Where

$r = [x \quad y \quad z]^T$; Distance between the Earth frame and the body frame (m)

M; Quadrotor's mass

g; Gravitational acceleration = $9.81 m/s^2$

R; Rotation matrix which is given in Equations (3.6)

F_B; Force acting on the quadrotor (non-gravitational forces) (N)

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -F_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -K_f(\Omega_1 + \Omega_2 + \Omega_3 + \Omega_4) \end{bmatrix} \tag{3.27}$$

Rewrite Equations (3.28)

$$M\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ Mg \end{bmatrix} + R\begin{bmatrix} 0 \\ 0 \\ -K_f(\Omega_1 + \Omega_2 + \Omega_3 + \Omega_4) \end{bmatrix} \tag{3.28}$$

3.3 Rotor Dynamics

Three-phase brushless DC motors (BLDC) which provide high torque and less friction. The dynamics of a BLDC at steady state is the same as DC motor [40] is used on the Quadrotor system, so BLDC motors can be modeled similarly to DC motors. A circuit diagram representing the model to be used is given in Figure11. Rotor dynamic model is obtained with the following assumptions;

- The magnetic circuit is linear, which is an approximation because of some flux dispersion inside the motor [46].

- Mechanical friction is only linear.
- There is no rigid mechanical coupling between the motors and the propellers.
- The inductance is exceptionally small.

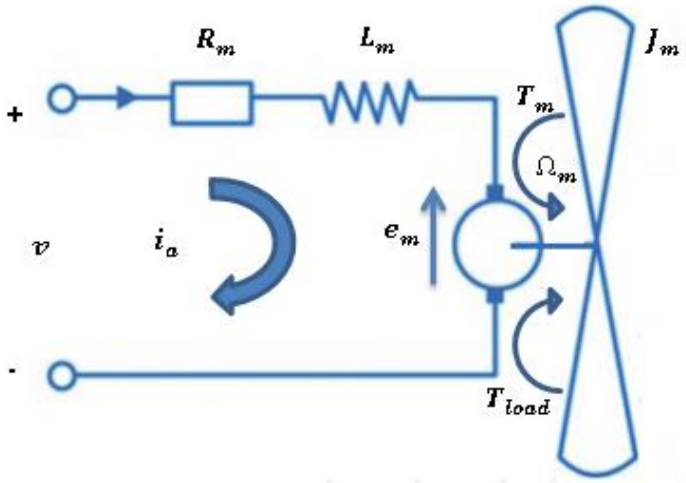

Figure 11: DC Motor Schematic Diagram

Using Kirchhoff's voltage law, the electrical equations can be written as;

$$v = v_{R_m} + v_{L_m} + v_{e_m} \quad \textbf{(3.29)}$$

Where;

v; The effective input voltage

v_{R_m}; The resistor voltage = $R_m i_a$

v_{L_m}; The inductor voltage =$L_m \frac{di_a}{dt}$

v_{e_m} The motor back-emf (electro motor force) voltage =$K_m \Omega_i$

R_m and $\mathbb{L}_m$ are the motor's resistance and inductance , i_a is the armature current, K_{mot} is the motor torque constant and Ω_i is the i_{th} motor angular velocity. Rewriting the Equations (3.31)

$$v = R_m i_a + \mathbb{L}_m \frac{di_a}{dt} + K_m \Omega_i \quad \textbf{(3.30)}$$

The effects of inductance can be ignored since for a typical modern DC motor of the type, the value of $\mathbb{L}_m$ is exceptionally small [43], so the voltage equation (3.32) can be simplified as;

$$v = R_m i_a + K_{mot} \Omega_i \quad \textbf{(3.31)}$$

or

$$i_a = \frac{v - K_{mot} \Omega_i}{R_m} \quad \textbf{(3.32)}$$

The mechanical equation can be expressed as follows:

$$J_m \Omega_i = T_m - T_{load} \quad \textbf{(3.33)}$$

Where

J_m; The rotor shaft inertia

T_m; The rotor torque $=K_e i_a$,

Where K_e is the motor's electric constant. For small motors K_e is approximately equal toK_{mot} [46].

T_{load}; The load torque which is generated from the propeller = $K_m \Omega_i^2$ (Eq. 3.19)

Substituting the Equation (3.34) into (3.35);

$$J_m \Omega_i = K_{mot} \frac{v - K_{mot} \Omega_i}{R_m} - K_m \Omega_i^2 \quad \textbf{(3.34)}$$

The relationship between angular velocity and voltage can be written as;

$$v = \frac{R_m}{K_{mot}} J_m \Omega_i^2 + K_{mot} \Omega_i + K_m R_m \Omega_i^2 \quad \textbf{(3.35)}$$

Typically a brushless DC motor has own embedded controller (ESC) that works with a Pulse Width Modulated (PWM) signal. The voltage supplied to the brushless DC motor is directly proportional to the rotational speed of their rotation [40]. By using a

system identification tool and black box identification process, the relationship of the motor and propeller pair can be found as a first order transfer function. It is sufficient to reproduce the dynamics between the propeller's speed set-point and its true speed [43]. Depending on the TurkUAV V2 motor data set which is given in Chapter 5 (Table.12), transfer can be given as;

$$G(s) = \frac{Actual\ rotor\ speed}{Commended\ Rotor\ Speed} = \frac{0{,}98}{0{,}062s + 1} \tag{3.36}$$

Also the aerodynamic force and moment equations are expressed as Equation (3.18) and Equation (3.19).

$$F_i = K_f \Omega_i^2$$
$$M_i = K_m \Omega_i^2$$

Where K_f is the aerodynamic force constant and K_m is the aerodynamic moment constant, the details of those constants are given in 'The Aerodynamic Force and Moment'. Depending on the TurkUAV V2 motor data set, those constants can obtain as;

$$K_f = 2.74x10^{-5}\ \frac{N}{(\frac{rad}{s})^2} \tag{3.37}$$

In literature it is seen that such ratio employed for aerodynamic force and moment constants [43], [44] and [45].

$$\frac{K_f}{K_m} = \frac{1}{58} \tag{3.38}$$

$$K_m = 0.047x10^{-5}\ \frac{Nm}{(\frac{rad}{s})^2} \tag{3.39}$$

Depending on the TurkUAV V2 motor data set which is given in Chapter 5 (Table 12), Duty [us] vs Thrust [N] Graph (Figure 12) and Duty [us] vs Angular Velocity [rad/s] Graph (Figure 13) can be obtained. By using line fit method, conversion equations can be obtained to use in simulation.

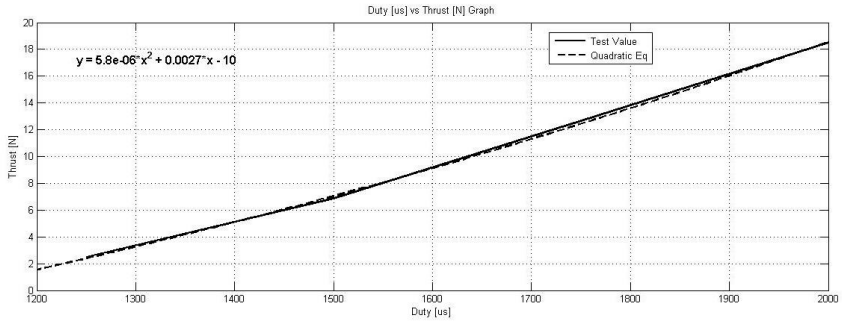

Figure 12: Duty [us] vs Thrust [N] Graph

Quadratic Equation of Duty vs Thrust is given as;

$$Thrust\ [N] = \ 5.86e-06Duty^2 + 0.0027Duty - 10 \quad \textbf{(3.42)}$$

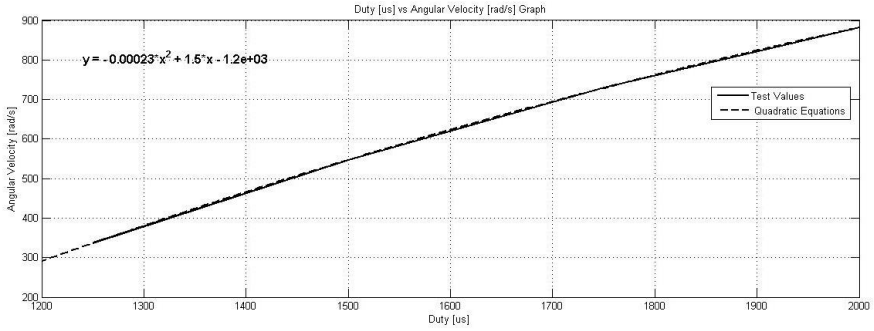

Figure 13: Duty [us] vs Angular Velocity [rad/s] Graph

Quadratic Equation of Duty vs Angular Velocity is given as;

$$Thrust\ [rad/s] = \ -0.00023Duty^2 + 1.5Duty - 1.2e + 03 \quad \textbf{(3.43)}$$

3.4 State Space Model

The mathematical model which is obtained in quadrotor can be written in the form of a state space model and that model will help make the control problem easier to overcome.

3.4.1 State Vector

Quadrotor's angular and linear velocity and position in space are defined in the state vector which is given in Equation 3.44 and 3.45.

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}] \quad \textbf{(3.44)}$$

$$X = [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ z \ \dot{z} \ x \ \dot{x} \ y \ \dot{y}] \quad \textbf{(3.45)}$$

3.4.2 Control Input Vector

The control input vector consisting of four inputs; U1 through U4 where U1 generates the desired altitude of quadrotor, U2 generates the desired roll angle, U3 generates the desired pitch angle and U4 generates the desired heading.

$$U = [U_1 \ U_2 \ U_3 \ U_4] \quad \textbf{(3.46)}$$

Where

$$U_1 = K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \ \text{(Throttle)} \quad \textbf{(3.47)}$$

$$U_2 = K_f(-\Omega_2^2 + \Omega_4^2) \ \text{(Roll)} \quad \textbf{(3.48)}$$

$$U_3 = K_f(\Omega_1^2 - \Omega_3^2) \ \text{(Pitch)} \quad \textbf{(3.49)}$$

$$U_4 = K_m(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \text{ (Yaw)} \tag{3.50}$$

Equations (3.47) through (3.50) can be written in a matrix form in Equation (3.51)

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -K_f & 0 & K_f \\ K_f & 0 & -K_f & 0 \\ -K_m & K_m & -K_m & K_m \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{3.51}$$

By using the control inputs, the rotor velocities can be calculated. To obtain the rotor's velocities, an inverse relationship between the control inputs and the rotor's velocities is needed which is given in Equation (3.52).

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4K_f} & 0 & \frac{1}{2K_f} & -\frac{1}{4K_m} \\ \frac{1}{4K_f} & -\frac{1}{2K_f} & 0 & \frac{1}{4K_m} \\ \frac{1}{4K_f} & 0 & -\frac{1}{2K_f} & -\frac{1}{4K_m} \\ \frac{1}{4K_f} & \frac{1}{2K_f} & 0 & \frac{1}{4 * K_m} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{3.52}$$

The rotor's velocities are given in Equation (3.53) through (3.56)

$$\Omega_1 = \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_3 - \frac{1}{4K_m}U_4} \tag{3.53}$$

$$\Omega_2 = \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_2 + \frac{1}{4K_m}U_4} \tag{3.54}$$

$$\Omega_3 = \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_3 - \frac{1}{4K_m}U_4} \tag{3.55}$$

$$\Omega_4 = \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_2 + \frac{1}{4K_m}U_4} \tag{3.56}$$

3.4.3 Rotational and Translational Equation of Motion

Rotational and Translational Equation of Motion are rearranged by substituting the control input and state vectors. In Dynamic Model section, those equations are obtained in Equation (3.17) and (3.30).

Rotational Equation of Motion

Substituting control input vector equation (3.51) in total moment's vector equation (3.27),

$$M_B = \begin{bmatrix} LU_2 + M_{xg} \\ lU_3 \\ U_4 \end{bmatrix} \tag{3.57}$$

Substituting equation (3.57) into the Rotational Equation of Motion (3.17),

$$\begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} x \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} x \begin{bmatrix} 0 \\ 0 \\ J_r\Omega_r \end{bmatrix} = \begin{bmatrix} LU_2 + M_{xg} \\ lU_3 \\ U_4 \end{bmatrix} \tag{3.58}$$

Rearranging equation (3.59),

$$\begin{bmatrix} J_{xx}\ddot{\phi} \\ J_{yy}\ddot{\theta} \\ J_{zz}\ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi}J_{zz}\dot{\psi} - \dot{\psi}J_{yy}\dot{\theta} \\ \dot{\psi}J_{xx}\dot{\phi} - \dot{\phi}J_{zz}\dot{\psi} \\ \dot{\phi}J_{yy}\dot{\theta} - \dot{\theta}J_{xx}\dot{\phi} \end{bmatrix} + \begin{bmatrix} \dot{\theta}J_r\Omega_r \\ -\dot{\phi}J_r\Omega_r \\ 0 \end{bmatrix} = \begin{bmatrix} LU_2 + M_{xg} \\ lU_3 \\ U_4 \end{bmatrix} \tag{3.59}$$

The angular accelerations in terms of the other variables are given in equations (3.60) through (3.62).

$$\ddot{\phi} = \frac{L}{J_{xx}} U_2 - \frac{J_r}{J_{xx}} \dot{\theta}\Omega_r + \frac{J_{yy}}{J_{xx}} \dot{\psi}\dot{\theta} - \frac{J_{zz}}{J_{xx}} + \frac{M_{Gear2}}{J_{xx}} - \frac{M_{Gear1}}{J_{xx}} \quad \textbf{(3.60)}$$

$$\ddot{\theta} = \frac{L}{J_{yy}} U_3 - \frac{J_r}{J_{yy}} \dot{\phi}\Omega_r + \frac{J_{zz}}{J_{yy}} \dot{\phi}\dot{\psi} - \frac{J_{xx}}{J_{yy}} \quad \textbf{(3.61)}$$

$$\ddot{\psi} = \frac{L}{J_{zz}} U_4 + \frac{J_{xx}}{J_{zz}} \dot{\theta}\dot{\phi} - \frac{J_{yy}}{J_{zz}} \dot{\phi}\dot{\theta} \quad \textbf{(3.62)}$$

Defining,

$a_1 = \frac{J_{yy} - J_{zz}}{J_{xx}}$; $a_2 = \frac{J_r}{J_{xx}}$; $a_3 = \frac{J_{zz} - J_{xx}}{J_{yy}}$; $a_4 = \frac{J_r}{J_{yy}}$; $a_5 = \frac{J_{xx} - J_{yy}}{J_{zz}}$

$b_1 = \frac{L}{J_{xx}}$; $b_2 = \frac{L}{J_{yy}}$; $b_3 = \frac{L}{J_{zz}}$

$c_1 = \frac{1}{J_{xx}}$;

Simplifying the equations (3.60) through (3.62),

$$\ddot{\phi} = b_1 U_2 - a_2 \dot{\theta}\Omega_r + a_1 \dot{\theta}\dot{\psi} + c_1 M_{Gear2} - c_1 M_{Gear1} \quad \textbf{(3.63)}$$

$$c1\, M_{xg} \approx c1(h - L_m cos\delta_2) mg cos\phi - c1(h + L_m cos\delta_1) mg cos\phi \quad \textbf{(3.64)}$$

$$\ddot{\psi} = b_3 U_4 + a_5 \dot{\phi}\dot{\theta} \quad \textbf{(3.65)}$$

Substituting active landing gears moment equations (3.64) into equations (3.63)

$$c1\, M_{xg} \approx c1 mg cos\phi \big((h - L_m cos\delta_2) - (h + L_m cos\delta_1)\big) \quad \textbf{(3.66)}$$

$$\ddot{\phi} = b_1 U_2 - a_2 \dot{\theta}\Omega_r + a_1 \dot{\theta}\dot{\psi} + c1 mg cos\phi\big((h - L_m cos\delta_2) - (h + L_m cos\delta_1)\big) \quad \textbf{(3.67)}$$

Rewriting equation (3.67),

$$\ddot{\phi} = b_1 U_2 - a_2 \dot{\theta}\Omega_r + a_1 \dot{\theta}\dot{\psi} + c1 mg cos\phi\big((h - L_m cos\delta_2) - (h + L_m cos\delta_1)\big) \quad \textbf{(3.68)}$$

$$\ddot{\theta} = b_2 U_3 + a_4 \dot{\phi}\Omega_r + a_3 \dot{\phi}\dot{\psi} \quad \textbf{(3.69)}$$

$$\ddot{\psi} = b_3 U_4 + a_5 \dot{\phi}\dot{\theta} \quad \textbf{(3.70)}$$

Rewriting in a simpler form in terms of the system states,

$$\ddot{\phi} = b_1 U_2 - a_2 x_4 \Omega_r + a_1 x_4 x_6 + c1 mg cos\phi\big((h - L_m cos\delta_2) - (h + L_m cos\delta_1)\big) \quad \textbf{(3.71)}$$

$$\ddot{\theta} = b_2 U_3 + a_4 x_2 \Omega_r + a_3 x_2 x_6 \quad \textbf{(3.40)}$$

$$\ddot{\psi} = b_3 U_4 + a_5 x_2 x_4 \quad \textbf{(3.72)}$$

Translational Equation of Motion

Substituting control input vector equation (3.51) in force acting on the quadrotor vector equation (3.29),

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix} \quad \textbf{(3.73)}$$

Substituting equation (3.73) into the Translational Equation of Motion (3.17),

$$M\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0\\ 0\\ Mg\end{bmatrix} + R\begin{bmatrix}0\\ 0\\ -U_1\end{bmatrix} \tag{3.74}$$

Expending the terns,

$$M\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0\\ 0\\ Mg\end{bmatrix} + \begin{bmatrix}c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi\\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\theta c\psi\\ -s\theta & s\phi c\theta & c\phi c\theta\end{bmatrix}\begin{bmatrix}0\\ 0\\ -U_1\end{bmatrix} \tag{3.75}$$

The accelerations in terms of the other variables are given in equations (3.76) through (3.78).

$$\ddot{x} = \frac{-U_1}{M}(sin\phi sin\psi + cos\phi cos\psi sin\theta) \tag{3.76}$$

$$\ddot{y} = \frac{-U_1}{M}(cos\phi sin\psi sin\theta - cos\psi sin\phi) \tag{3.77}$$

$$\ddot{z} = g - \frac{U_1}{M}(cos\phi cos\theta) \tag{3.78}$$

Rewriting in terms of the state variable,

$$\ddot{x} = \frac{-U_1}{M}(sinx_1 sinx_5 + cosx_1 cosx_5 sinx_3) \tag{3.79}$$

$$\ddot{y} = \frac{-U_1}{M}(cosx_1 sinx_5 sinx_3 - cosx_5 sinx_1) \tag{3.80}$$

$$\ddot{z} = g - \frac{U_1}{M}(cosx_1 cosx_3) \tag{3.81}$$

3.4.4 State Space Representation

The complete mathematical model of the quadrotor can be written in a state space representation as follows,

$$\dot{x}_1 = \dot{\phi} = x_2$$

$$\dot{x}_2 = \ddot{\phi} = b_1 U_2 - a_2 x_4 \Omega_r + a_1 x_4 x_6 + c1 mg cos x_1 \big((h - L_m cos\delta_2) - (h + L_m cos\delta_1)\big)$$

$$\dot{x}_3 = \dot{\theta} = x_4$$

$$\dot{x}_4 = \ddot{\theta} = b_2 U_3 + a_4 x_2 \Omega_r + a_3 x_2 x_6$$

$$\dot{x}_5 = \dot{\psi} = x_6$$

$$\dot{x}_6 = \ddot{\psi} = b_3 U_4 + a_5 x_2 x_4$$

$$\dot{x}_7 = \dot{z} = x_8$$

$$\dot{x}_8 = \ddot{z} = g - \frac{U_1}{M}(cos x_1 cos x_3)$$

$$\dot{x}_9 = \dot{x} = x_{10}$$

$$\dot{x}_{10} = \ddot{x} = \frac{-U_1}{M}(sin x_1 sin x_5 + cos x_1 cos x_5 sin x_3)$$

$$\dot{x}_{11} = \dot{y} = x_{12}$$

$$\dot{x}_{12} = \ddot{y} = \frac{-U_1}{M}(cos x_1 sin x_5 sin x_3 - cos x_5 sin x_1)$$

A state space matrix can be written as;

$$= \begin{bmatrix} x_2 \\ b_1U_2 - a_2x_4\Omega_r + a_1x_4x_6 + c1mgcosx_1\big((h - L_mcos\delta_2) - (h + L_mcos\delta_1)\big) \\ x_4 \\ b_2U_3 + a_4x_2\Omega_r + a_3x_2x_6 \\ x_6 \\ b_3U_4 + a_5x_2x_4 \\ x_8 \\ g - \frac{U_1}{M}(cosx_1cosx_3) \\ x_{10} \\ \frac{-U_1}{M}(sinx_1sinx_5 + cosx_1cosx_5sinx_3) \\ x_{12} \\ \frac{-U_1}{M}(cosx_1sinx_5sinx_3 - cosx_5sinx_1) \end{bmatrix} \quad \textbf{(3.82)}$$

CHAPTER 4
CONTROLLER DESIGN and SIMULATION

In this study; a PID controller is used to control system attitude dynamics. This controller is designed based on the control structure of embedded flight board's software as shown in Figure 14. Briefly a cascade PID controller structure is used for attitude stabilization.

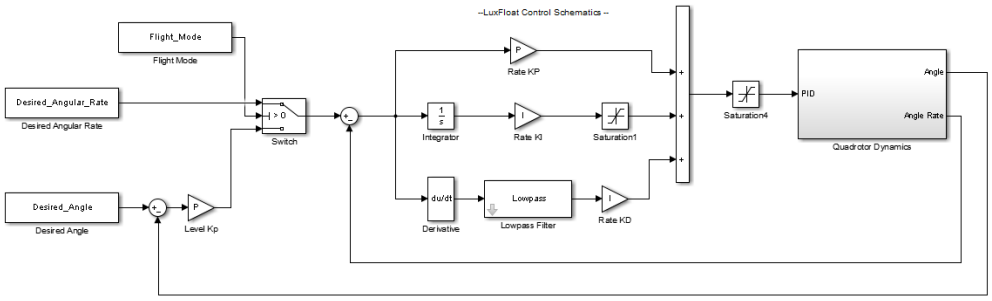

Figure 14: 'LuxFloat' Schematics Simple Presentation

The controller schematics and detailed information about controller are given in Section 4.1. In general application, this controller is tuned on test system by using trail-error method for each axis. For this method a test bench structure is designed in Section 5.1. By the help of test bench structure, controller parameters are tuned to obtain some default values for Roll axis which is given in Table 1. The step response results are given in Figure 15, 16 and 17.

Table 1- Some Default Controller Values for Roll Axis

		Kp	Ki	Kd	Kp_Level
1	(Figure 14)	2.5	0.6	0.06	4
2	(Figure 15)	2.0	0.3	0.03	2.0
3	(Figure 16)	2.8	0.48	0.07	4.8

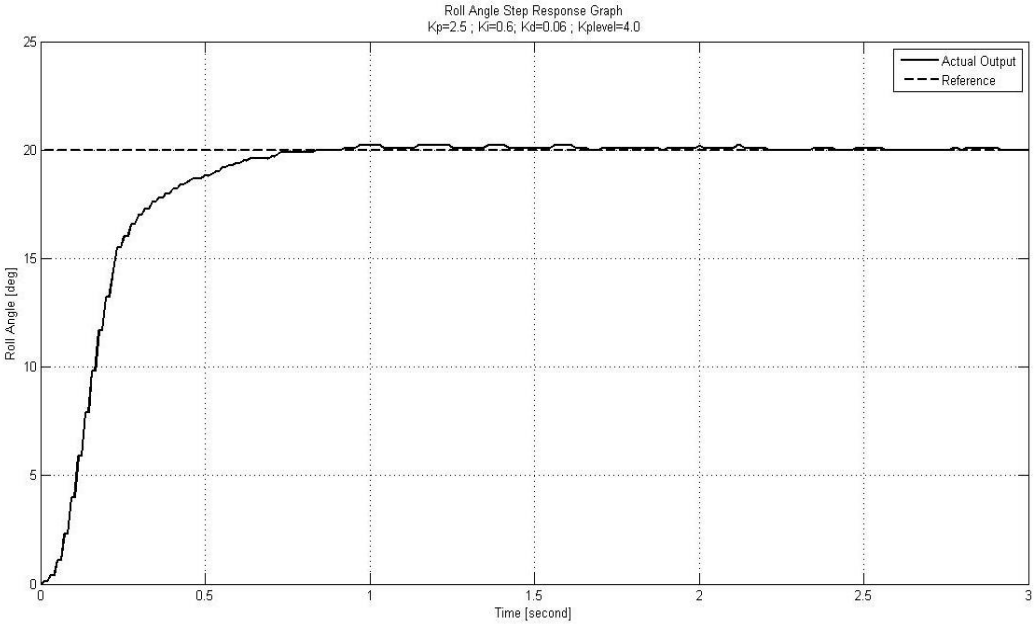

Figure 15: Roll Angle Step Response for Controller 1

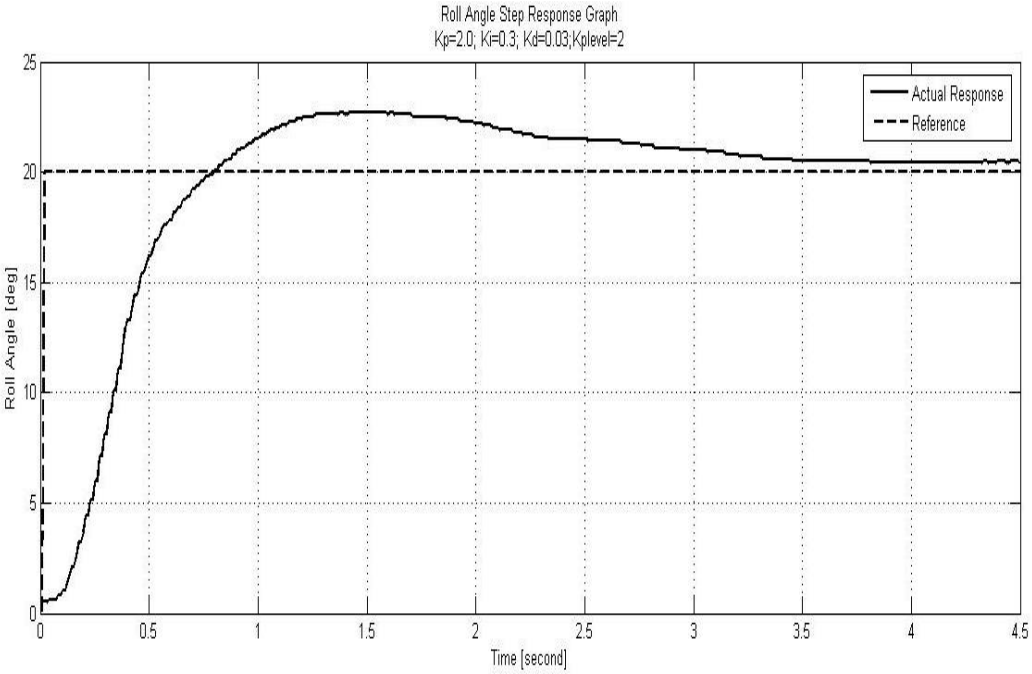

Figure 16: Roll Angle Step Response for Controller 2

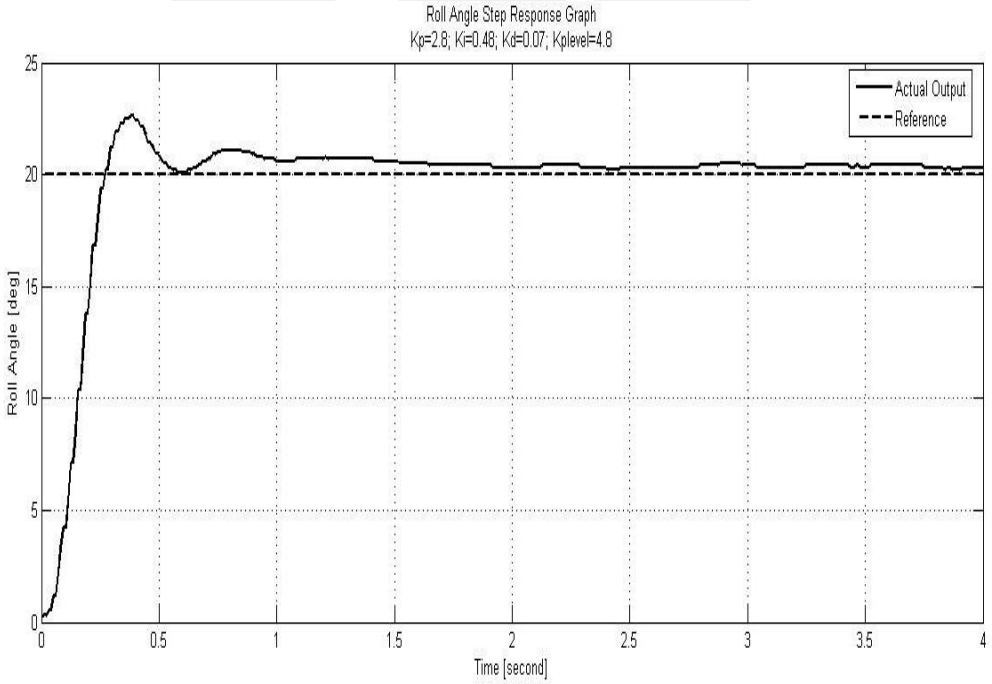

Figure 17: Roll Angle Step Response for Controller 3

Also this controller is modeled in MATLAB/Simulink environment and applied to one axis quadrotor model on test bench. By using default controller values data set, controller and quadrotor model is optimized by using Simulink optimization methods [76]. By using optimized model and controller, desired controller response is determined and controller parameters are obtained. Those values are applied to the test system and compared with model result. Also this controller is modelled on MATLAB/Simulink environment and applied to nonlinear quadrotor model in Section 4.2.

4.1 Embedded Control Board Software Control Architecture

To control the quadrotor system's attitude dynamics, an embedded flight controller board which is called Naze32 rev.5 [5] is used. This board is so popular and used in RC community. This controller card uses CleanFlight software [6] which is an open source project based on Multiwii project [57]. This software can be used in various controller cards. Controller cards divided in to two groups; 8 bits and 32bits. Naze 32 rev.5 is a 32 bit controller flight controller board, has built-in IMU sensors, built-in control and IO ports. Detailed information about card is given in Chapter 5. CleanFlight project and boards are developed quickly day after day by the help of the developers and testers. On the other hands, in the literature, there is a huge lack of studies about this topic. This board and software can be used in most academic studies. This offers researchers the opportunity to deal unnecessarily on hardware and software, just focus on applications. This project is still on progress and one of the main topics is controller structure. Cleanflight software offers five PID controller structures; each of them has same control methods, but shows some little differences. Controller structures are still developing and discussed the performance of each one.

In this study, only one PID structure which is named LuxFloat is determined to use and investigate. In this structure, a cascaded PID is used, inner loop is used for stabilizing desired angle, and outer loop is used for stabilized desired angle rate. In Figure 18, simple schematic is given.

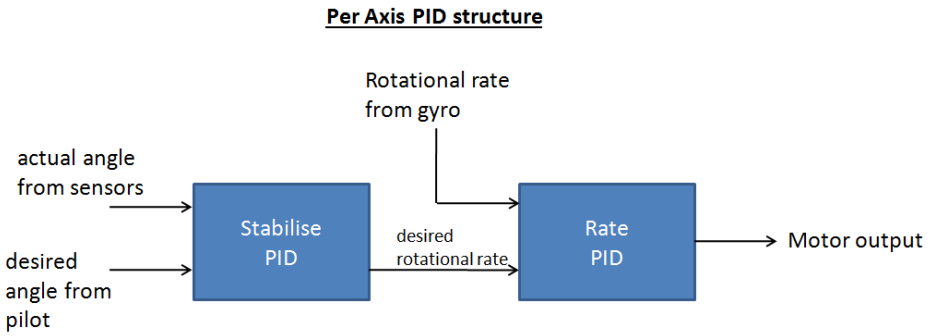

Figure 18: Cascade Control Structure

LuxFloat controller is designed for 32 bit controller, so there is no digital conversion processed and International Standard (IS) units are used in this control structure, because of that reason, modelling and working in simulation environment becomes as a conventional control method. Detailed of the control structure is given in Figure 19.

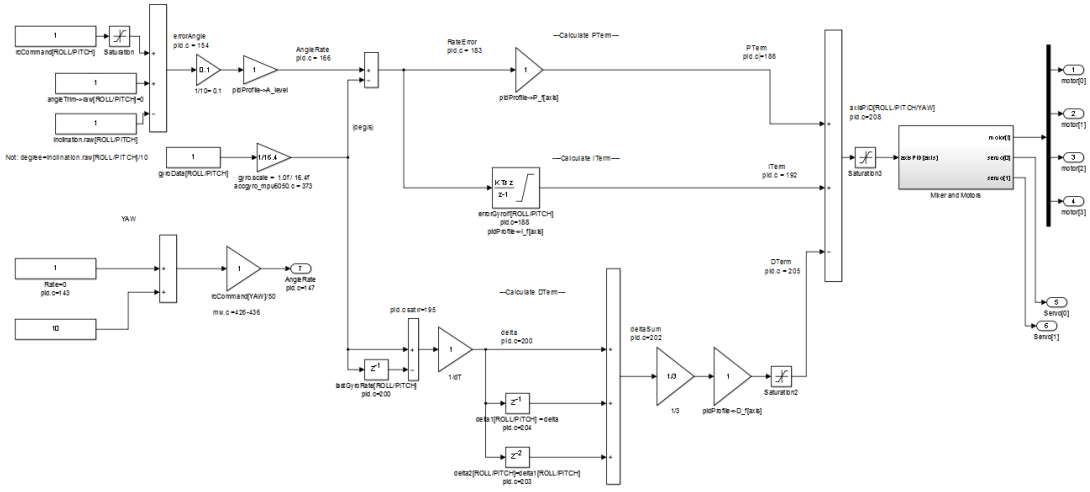

Figure 19:'LuxFloat' Shematics Detailed Presentation

C code of LuxFloat PID control structure is given in Appendix A.

4.2 Design Based on Test Bench and Model Simulations

Quadcopter platform is coupled to a revolute joint of a test bench that constrains the motion of the platform. Pivoting point *O* is on the chassis beam of the system as shown in Figure 20 and application on test bench system can be seen in Figure 21.

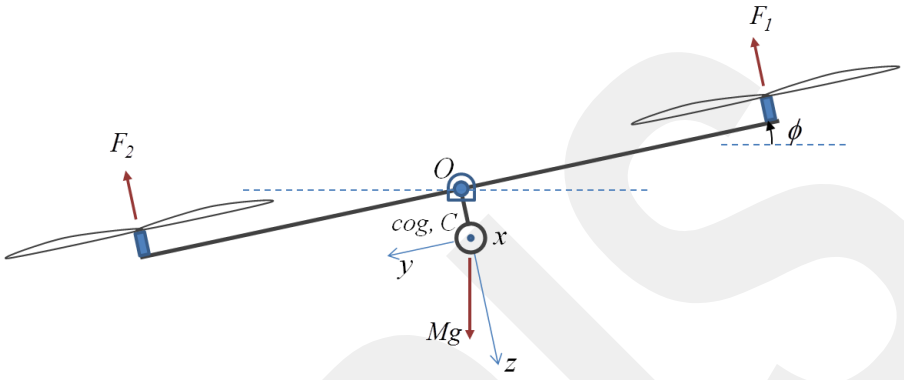

Figure 20: Platform on the test bench pivoted at O

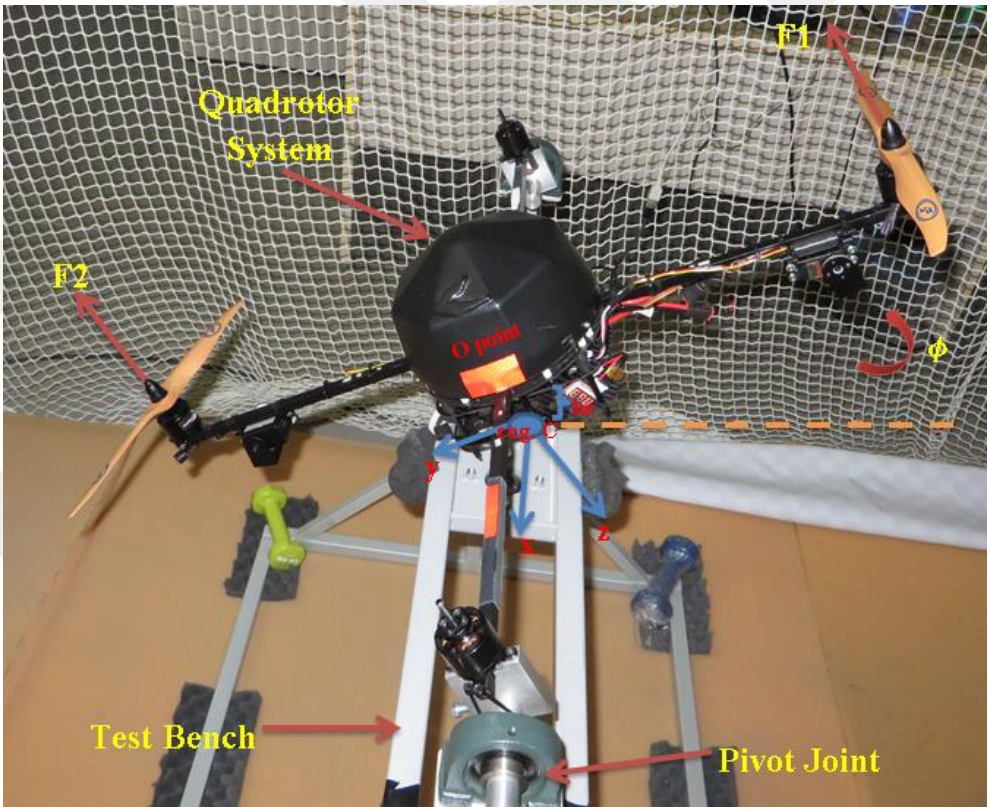

Figure 21: Application on Test Bench System

Designed control systems are tested and tuned on this bench. However, as it is seen in Figure 22, pivoting point does not match the center of gravity point C. This provides additional stiffness due to the location of the center of gravity with respect to the chassis. In air, rigid body of the quadcopter will rotate about its center of gravity. Designed and tuned controller as in Figure 22 must also perform the desired performance when the rotation occurs about point C. In order to simulate the performance of the controller that is tuned as in Figure 22 on the conditions that platform rotates about point C following models are built up.

Equation of motion that governs the dynamics of the platform in Figure 22 is given as below.

$$(J_{xx} + Mh_c^2)\ddot{\emptyset} + b\dot{\emptyset} + Mgh_c sin\emptyset = (F_1 - F_2)L \quad \textbf{(4.1)}$$

By using Parallel-axis Theorem [77];

$$J_o = (J_{xx} + Mh_c^2) \quad \textbf{(4.2)}$$

$$M_x = (F_1 - F_2)L \quad \textbf{(4.3)}$$

Where;

J_{xx}; Inertia on x axis ($kg.m^2$)

J_o; Inertia on O point ($kg.m^2$)

M; Quadrotor body mass (kg)

g; Gravitational force(m/s^2)

b; Viscous damping (N.m.s)

L; Length from rotor to COG, C point (m)

h_c; Length from O point to COG, C point (m)

$\emptyset$; Roll angle (rad)

F_1; Rotor 1 thrust (N)

F_2; Rotor 2 thrust (N)

M_x; Total moment produced by propellers

Dissipative torque due to viscous damping in revolute joint is also assumed and modeled in Equation 4.1. It is linearized assuming small roll angles in Equation 4.4.

$$J_o\ddot{\emptyset} + b\dot{\emptyset} + Mgh_c\emptyset = M_x \tag{4.4}$$

The system that is coupled to the test bench thru point C is given in Figure 22. This figure and governing equations of motion show what if the platform is coupled to the test bench at point C.

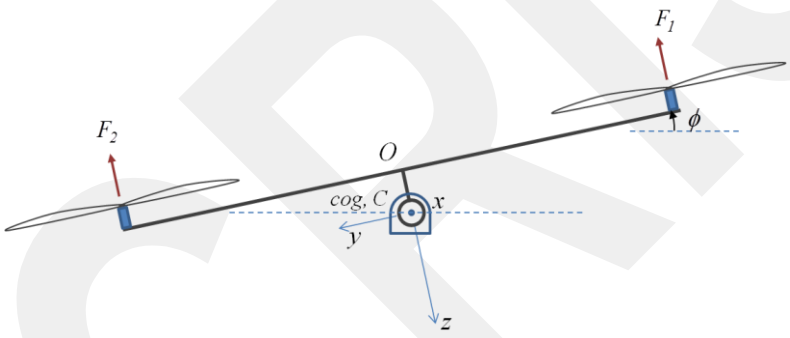

Figure 22: Platform on the test bench pivoted at C

Considering the system in Figure 22, governing equation of motion is given in Equation 4.5.

$$J_{xx}\ddot{\emptyset} + b\dot{\emptyset} = M_x \tag{4.5}$$

Control system-in the form of 'LuxFloat' structure is tuned on test bench. Obtained controller gains are applied on the hybrid model that includes digital controller model together with continuous time system model derived from Equation 4.4. Controller and test bench model Simulink block diagram is given in Figure 23. Data of the actual response is employed to estimate some of the physical parameters namely viscous damping ratio, *b*, location of the center of gravity with respect to

pivoting point, h_c, and the propulsion unit model parameters c_4, c_5, c_6 which are obtained in Rotor Dynamics Section (Equation 3.42) That model is given as below.

$$Thrust = c_4(duty\ ratio)^2 + c_5(duty\ ratio) + c_6 \quad \textbf{(4.6)}$$

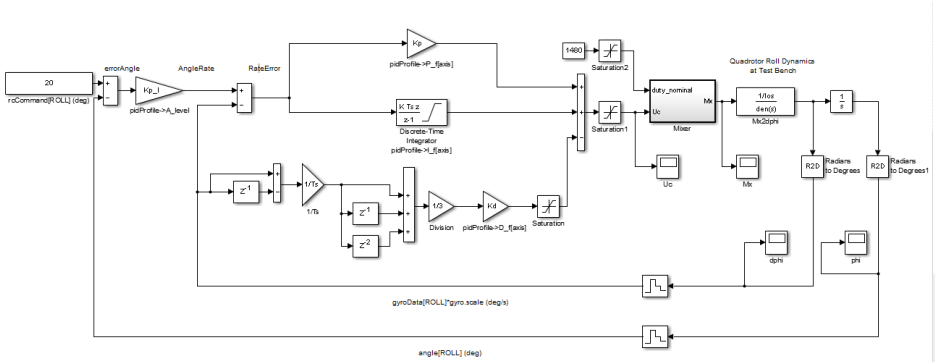

Figure 23: Controller and test bench model Simulink block diagram

Identified model response and actual system response on test bench are given in Figure 24.

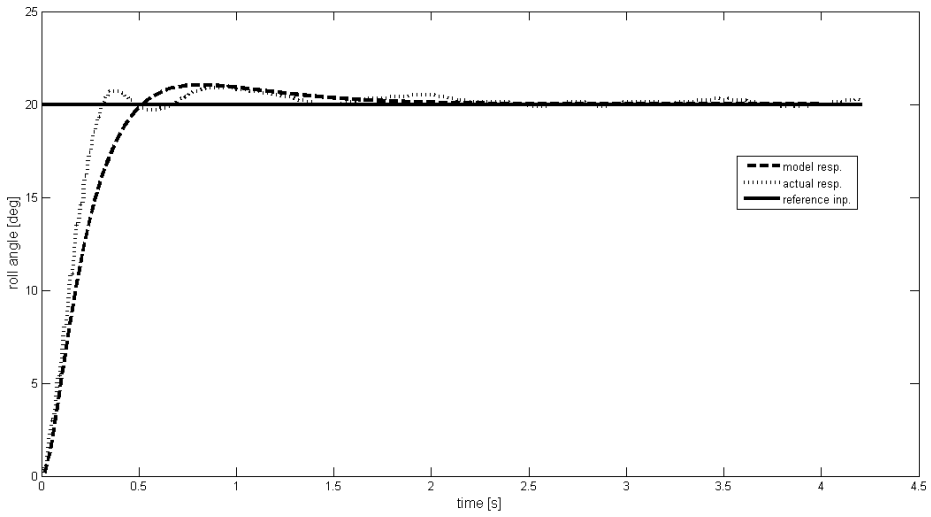

Figure 24: Actual system response and model response

Control parameters and estimated parameters are applied on the model based on the equations of motion given in Equation 4.5. This model simulates the behavior of the quadcopter on test bench when it is pivoted at its center of gravity (Figure 25).

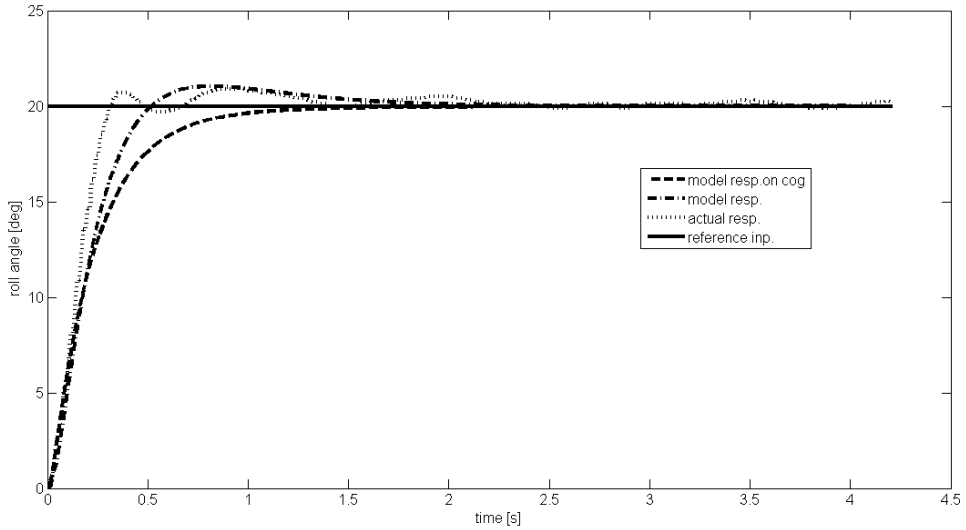

Figure 25: Actual system response, model responses

It is seen that simulated rotation behavior about center of gravity is somehow different from responses of actual system and the one that simulates the behavior in case of pivot at O. However, it is also an accepted behavior for the quadcopter. Next simulation is performed on nonlinear attitude dynamics model which is given in Chapter 3. Tuned controller gains and estimated parameters are utilized in the nonlinear model (Figure.26). The simulations are given for roll and pitch axis in Figure 27 and 28.

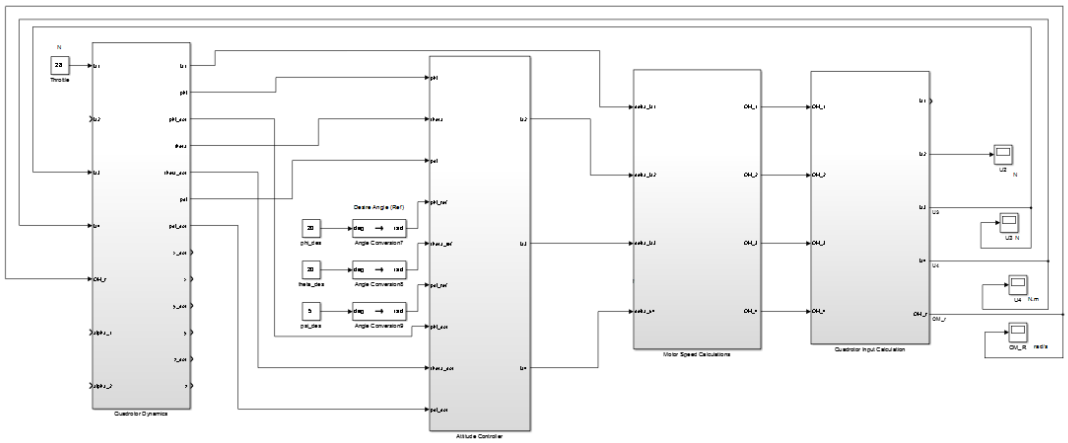

Figure 26: Nonlinear Quadrotor Model Simulink Block Diagram

Tuned parameters Kp=2.5; Ki=0.6; Kd=0.06, Kp_Level= 4

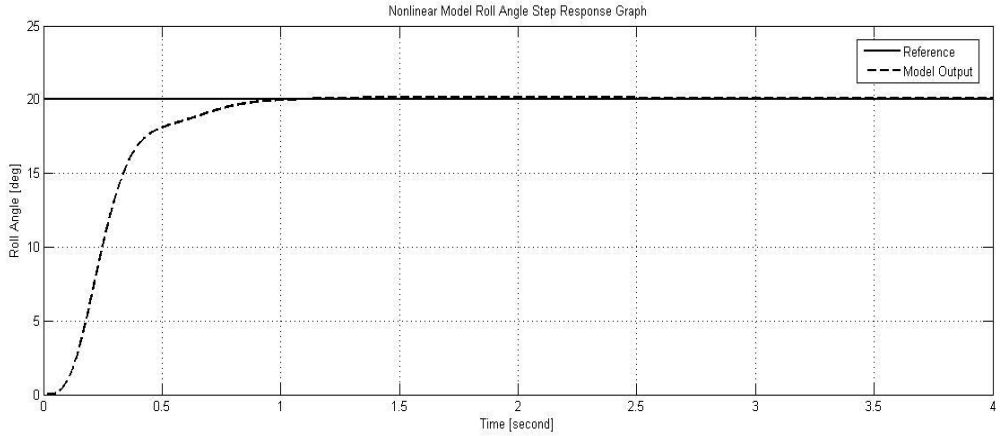

Figure 27: Nonlinear model Roll Angle Step Response Graph

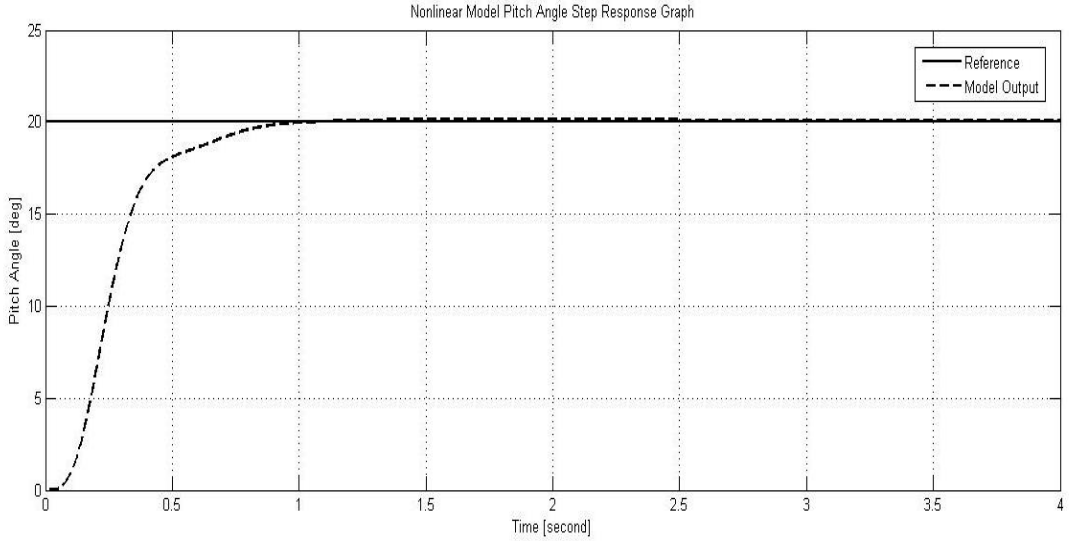

Figure 28: Nonlinear model Pitch Angle Step Response Graph

Design procedure can applied in the other direction also. Designing controller (optimizing the controller gains on simulation models) is the initial step. Second step is to apply the gains on test bench and finally they are applied and tuned during free flight tests. The constructed simulation and hardware architecture enables the realization of both directions. A typical controller optimization based on the Figure 23 having the following criterion;

- Rise time must be less than 0.25 seconds,

- Settling time must be less than 1 seconds,
- Maximum overshoot must be less than 10%.

Optimized controller gains are also applied to the model based on Figure 20 and Figure 22. Comparison of the 2 model responses are given in Figure 29. As expected, model representing the system pivoted at point *O* has a higher overshoot and more oscillations. Because, system on test bench as pivoted at *O* behaves like an inverted pendulum. Test bench response is given in Figure 30 for tuned parameters; Kp=2.8; Ki=0.48; Kd=0.07, Kp_Level= 4.8;

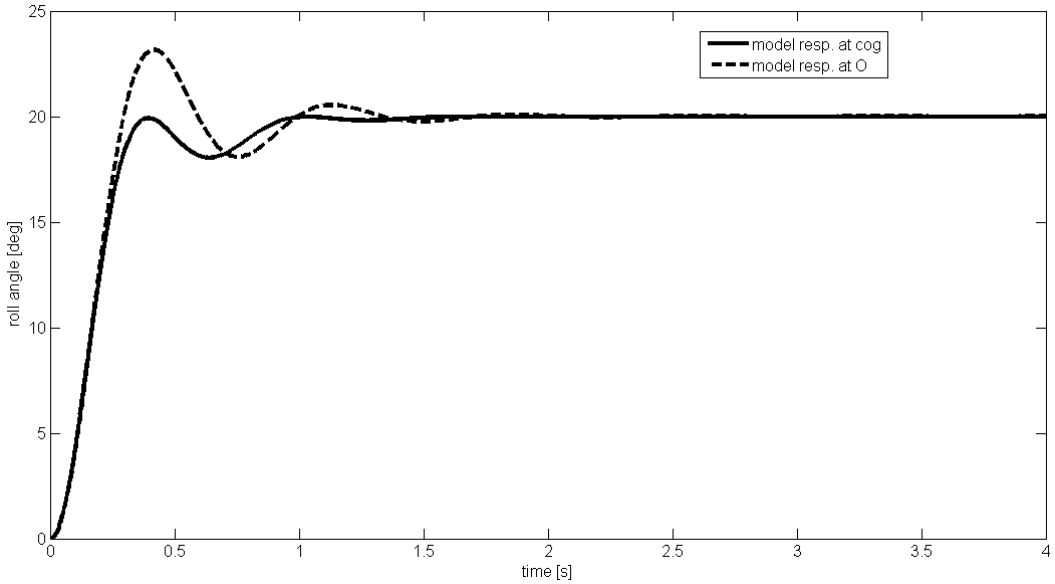

Figure 29: Comparison of model responses

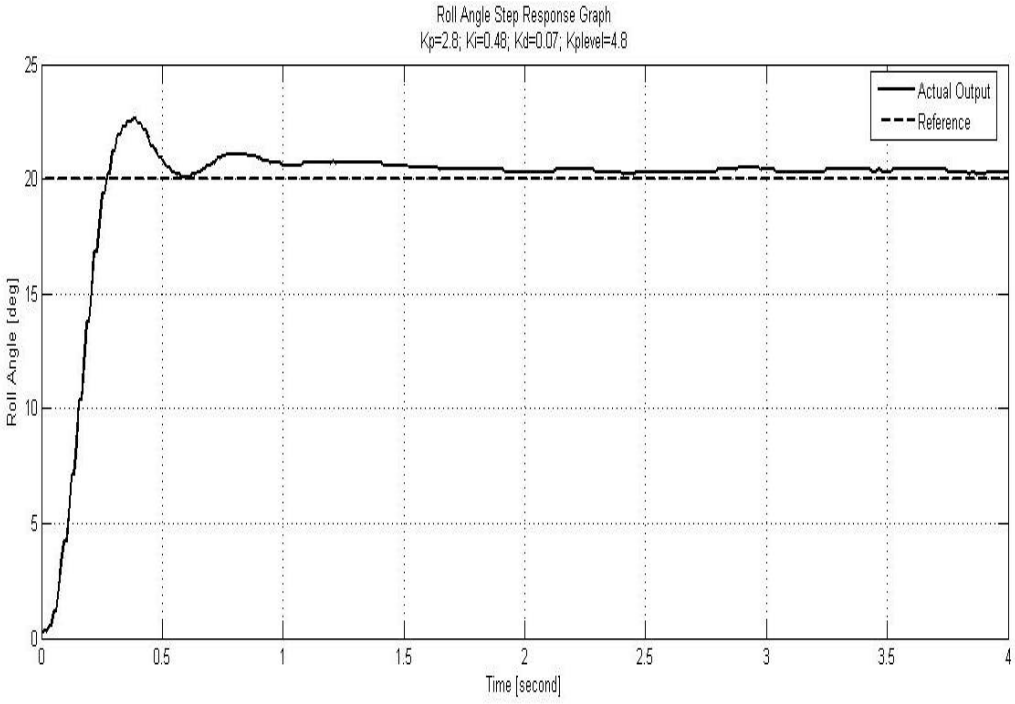

Figure 30: Test bench response

The test bench pivoted at C test result will be obtained in flying test.

4.3 Simulations with the Effects of Active Landing Gears

Active landing gears are actuated, single degree of freedom (DOF) links. They are used to have a stable assist mode on ground and they can be opened in extended mode in air to have a wide range of vision for the camera system. There are 2 links on the quadcopter platform in this thesis (Figure 31 and 32). In this study, it is desired to evaluate the effect of these links on the attitude dynamics. Considering the coupled dynamics, they can be used to steer the robot and/or assist the main control by rejecting the disturbances, etc. In fact, recent studies show that multi degrees of freedom robot arms and similar sub systems will be placed on multicopters and they will be designed to perform specific tasks such as being a flying manipulator. Single DOF links are the simplest dynamical sub systems that can be placed on a flying robot. Therefore available landing gears can be used to assess the coupled dynamics.

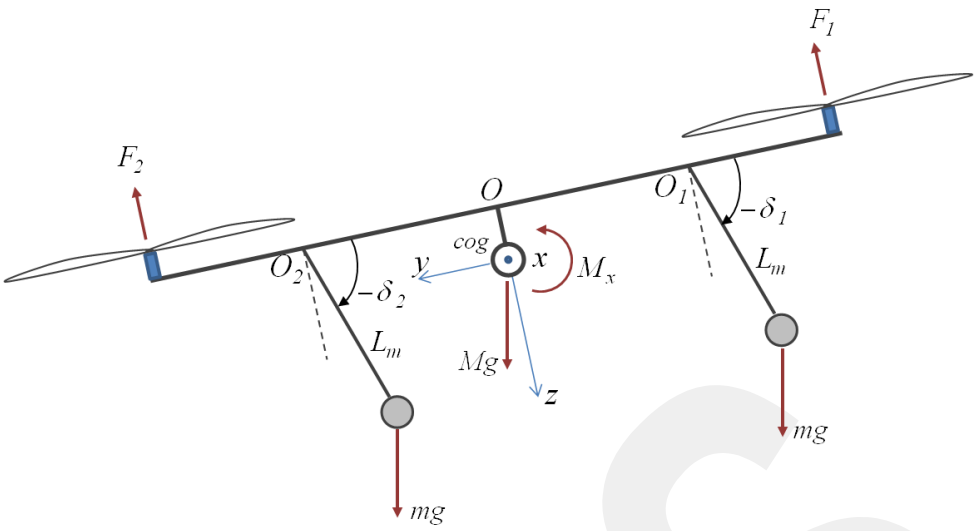

Figure 31: Landing Gears on Quadcopter Platform Freebody diagram

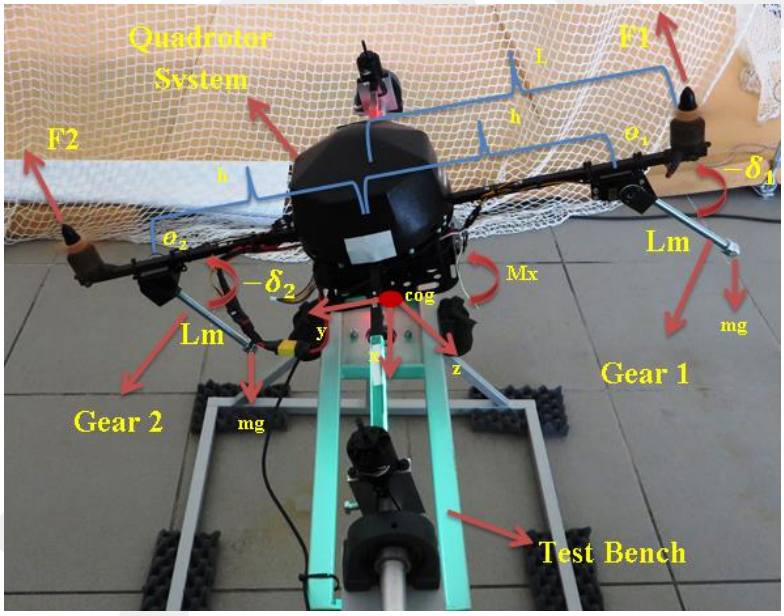

Figure 32: Landing gears on quadcopter system

Effect of the landing gears about body x-axis is simply modeled as a moment term, M_{xg}, as follows;

$$M_{xg} \approx \sim(h - L_m cos\delta_2)mgcos\emptyset - (h + L_m cos\delta_1)mgcos\emptyset \quad \textbf{(4.7)}$$

Where;

h; the distance between O and O_1 points (O and O_2 points)

L_m; length of landing gear rods

m; mass of gears

δ_1, δ_2; landing gears angles

M_{xg}; the moment generated by landing gears

M_{xp} is the moment generated by the propulsion units is given in Equation 4.8.

$$M_{xp} = (F_1 - F_2)L \tag{4.8}$$

Where;

L; the distance between COG and Rotors (m)

F_1; Rotor 1 thrust (N)

F_2; Rotor 2 thrust (N)

Total moment, M_x, generated about body x-axis is the summation of M_{xg} and M_{xp} is given in Equation 4.10. As the angular positions of the links with respect to the chassis change, applied moment due to the landing gears about body x-axis can be manipulated.

$$M_x = M_{xp} + M_{xg} \tag{4.9}$$

$$M_x = [(F_1 - F_2)L) + ((h - L_m cos\delta_2)mgcos\emptyset - (h + L_m cos\delta_1)mgcos\emptyset] \tag{4.10}$$

The control approach of the gears is given in Figure 33. Regulatory feedback controller is used to control the active landing gears.

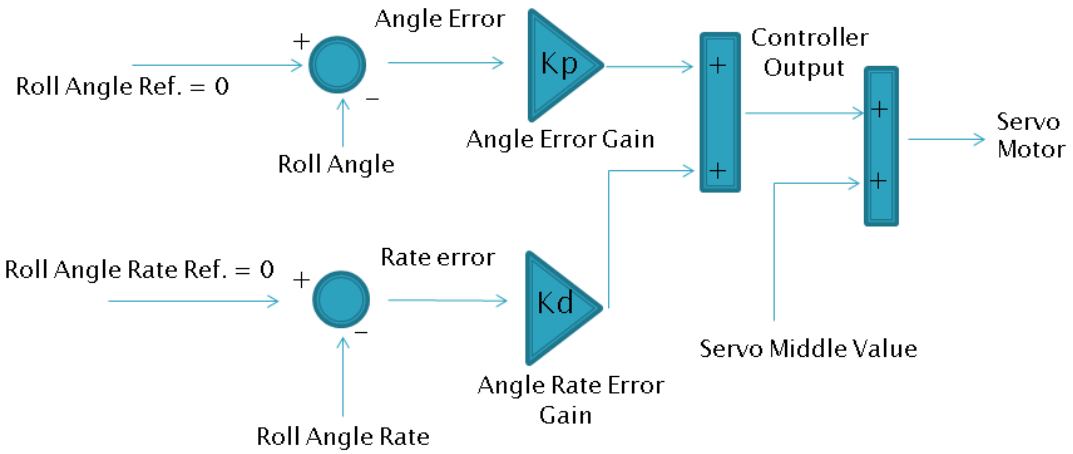

Figure 33: Control Blok Diagram

Briefly, gears always try to remain at the position which is determined with "Servo Middle Value". That's why the angle reference and angle rate reference are always zero. By adding the rate error, it is expected to react more quickly from gears. The controller approach can be represented in Equation 4.15. C code of this approach is given in Appendix B.

$$Angle\ Error = 0 - Roll\ Angle\ (\phi) \tag{4.11}$$

$$Rate\ Error = 0 - Roll\ Angle\ Rate(\dot{\phi}) \tag{4.12}$$

$$Controller\ Ouput = Angle\ Error * K_p + Rate\ Error * K_d \tag{4.13}$$

$$Servo\ Motor = Controller\ Output + Servo\ Middle\ Value(\mathrm{C}) \tag{4.14}$$

Rewrite Equation 4.14;

$$Servo\ Motor\ [us] = \left(-(\phi) * K_p + -\left(\dot{\phi}\right) * K_d\right) + \mathrm{C} \tag{4.15}$$

This control approach is modeled in simulation environment (Figure 34) and applied to the quadrotor roll dynamics.

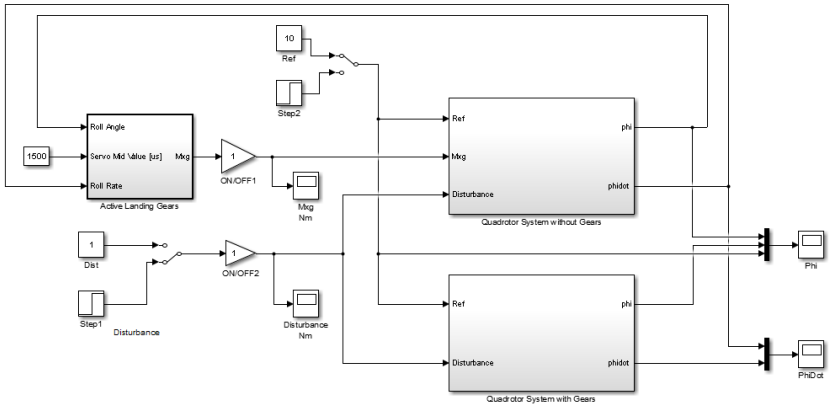

Figure 34: Simulation Blok Diagram

Effects of active landing gears are observed in simulation environment. In simulation, one of the quadrotor systems is used active landing gear and other is not used active landing gear. Same reference and disturbance are applied two of them and the result is given in Figure 35. In this test, active landing gears variables and controller gains are given below;

Gears Variables

L_m; length of landing gear rods = 0.30 m

m; mass of gears = 0.1 kg

h; the distance between O and O_1 points (O and O_2 points) = 0.17 m

Gears Controller Gain

K_p; angle gain = 30 ;

K_d; angle rate gain= 3 ;

LuxFloat Controller Gain

Kp_Level= 5; Kp=2.5; Ki= 0.45; Kd= 0.045

Reference = 0 degree, Disturbance = 2 Nm

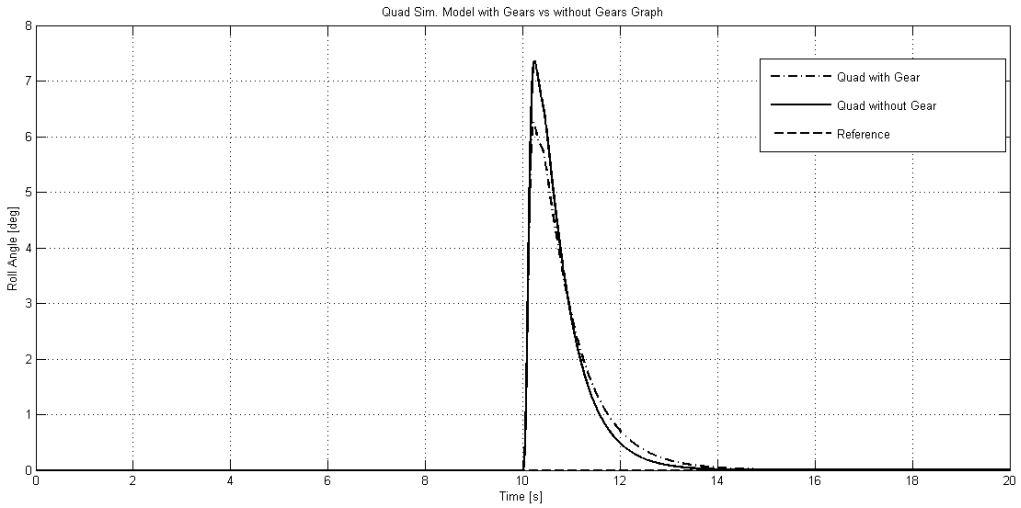

Figure 35: Simulation Result Graph

As shown in Figure 35, when applied a same disturbance to two system. Quadrotor with gear gives %1.2 less responsive than Quadrotor without Gear. This means, active landing gear provides a positive effect on the Quadrotor attitude dynamics. Active landing gear creates a 0.37 Nm (Mxg) moment on the total Quadrotor's roll moment. By the help of this moment (Mxg), Quadrotor gives less reaction to the disturbance than other system. In Chapter 6, flight test results are given.

CHAPTER 5

PHYSICAL SYSTEM

In this chapter, the components of the quadrotor test system which are used in all tests are given. Quadrotor platform basically consists of;

- Flight Controller
- Inertial measurement unit
- Electronic Speed Controller
- Telemetry Units
- Brushless DC motors
- Propellers
- Battery
- RC Transmitter
- Active landing gears

In Figure 36, Quadrotor system can be seen.

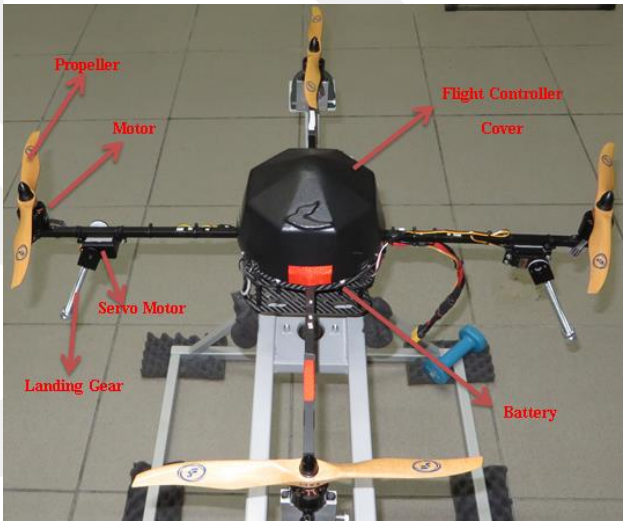

Figure 36: Quadrotor System

In this thesis, a commercial quadrotor frame which is produced by Robonik Mekatronik Teknolojileri Company [56] is used. This frame is used in commercial applications. The frame is made from carbon fiber, aluminum (6063-T6), derlin and

ABS materials. Also brushless motors and electronic speed controller are produced by same company. The controller hardware is embedded onto the system. The controller hardware is based on an open source project which is called Multiwii [57]. The controller hardware consists of a microcontroller unit, an IMU unit and the input/output (I/O) ports. All of them are combined on the same controller board namely NAZE32 [5]. The software which is used for programming, tuning and logging data is an open source as well [6]. For telemetry system, two Xbee modules are used and an RC transmitter which has 8 channels is used to control the system manually. System has two servo motors for actuating active landing gears; those motors are driven from flight controller card directly. The required power is provided by one 4S (14.8V 4A) Li-Po battery. A simple schematics representation of the relation among components is given in Figure 37.

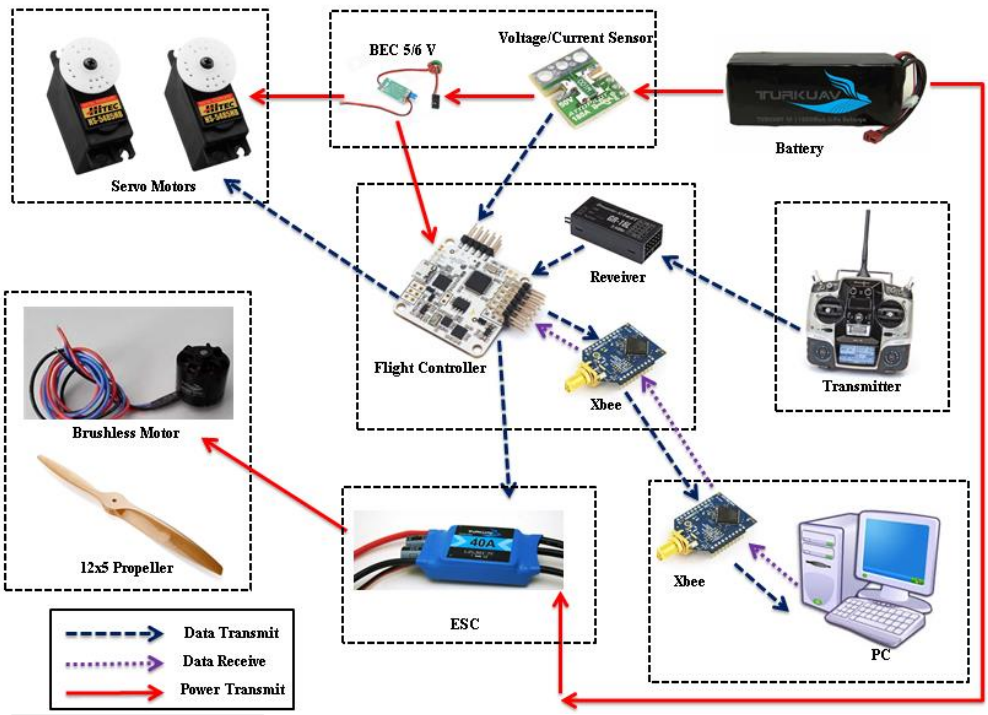

Figure 37: Simple Physical Schematics

In this Chapter, the mechanical structure, sensor system, controller system, actuator system and other components which are the elements of quadrotor system are described in details.

5.1 Mechanical System

In this thesis, a commercial frame is used. This frame is made from carbon fiber, aluminum (6063-T6), derlin, plastic and ABS materials. This frame consists of four aluminum riggers, two carbon fiber main body plates, one protect cover, four derlin main body support elements, two carbon fiber battery holder plates, and the other connections elements. In Figure 38, the components of the quadrotor frame are presented.

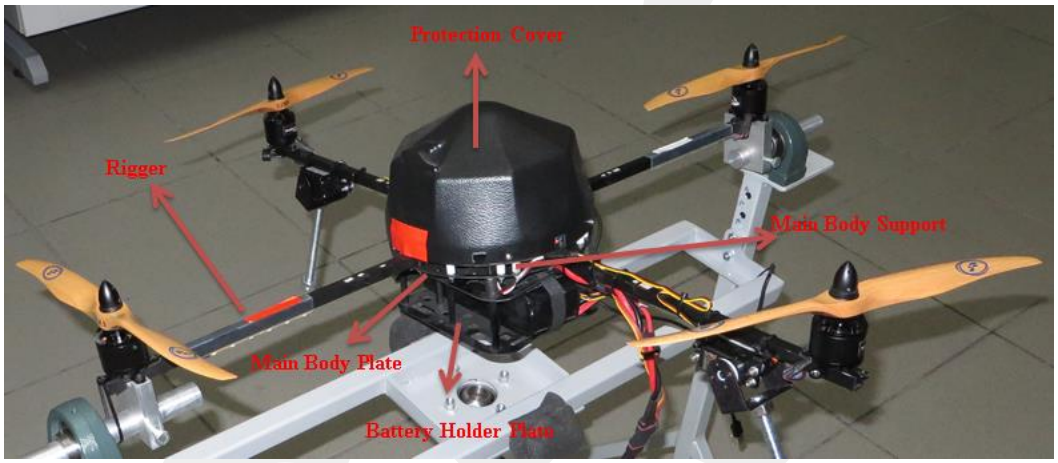

Figure 38: Quadrotor System Frame Parts

The total mass of the quadrotor system is 2468.04 gr. Quadrotor system physical parameters are given in Table 2. System moment of inertia matrix parameters are given in Inertia Test section.

Table 2- Quadrotor System Physical Parameters

Mass	2655.09 gr
Rotor to Rotor Dimensions	687 mm
Rigger Length	355 mm
Center of Gravity (From Riggers)	65.2 mm
J_{xx} (Roll)	$0.0457 kg.m^2$
J_{yy} (Pitch)	$0.0457 kg.m^2$
J_{zz} (Yaw)	$0.0846 kg.m^2$

5.1.2 Moment of Inertia Test

The quadrotor system's moment of inertia which is given in mathematical modeling chapter in Equation 3.16 can't be determined easily. Those parameters are important for mathematical model and control designing chapters. By the help of CAD design, inertia parameters of quadrotor system are calculated experimentally. Due to the symmetry, it is assumed that inertia matrix is diagonal. Therefore products of inertia terms are all zero. To obtainJ_{xx} , J_{yy}, and J_{zz} , bifilar pendulum experiments are done [58]. In this experiment, quadrotor system is hanged from two sides by ropes to make it free to oscillate on each axis. Two ropes are tied from the points which are approximately same distance to each rotation axis of quadrotor. Figure 39 shows the experimental setup for Roll (J_{xx}) axis.

Figure 39: Inertia Test on Roll (Jxx) axis

After establishing the test setup, quadrotor is released to oscillate from a small initial angle. This response is acquired by an IMU. In this experiment, Microstrain 3DM GX3-25 IMU [59] in Figure 40 is used. By using Microstrain Data Monitor software

[60] (Figure 41), all data are logged. Data is analyzed to derive the period of the oscillations, Figure 42.

Figure 40: Microstrain 3DM GX3-25 [59]

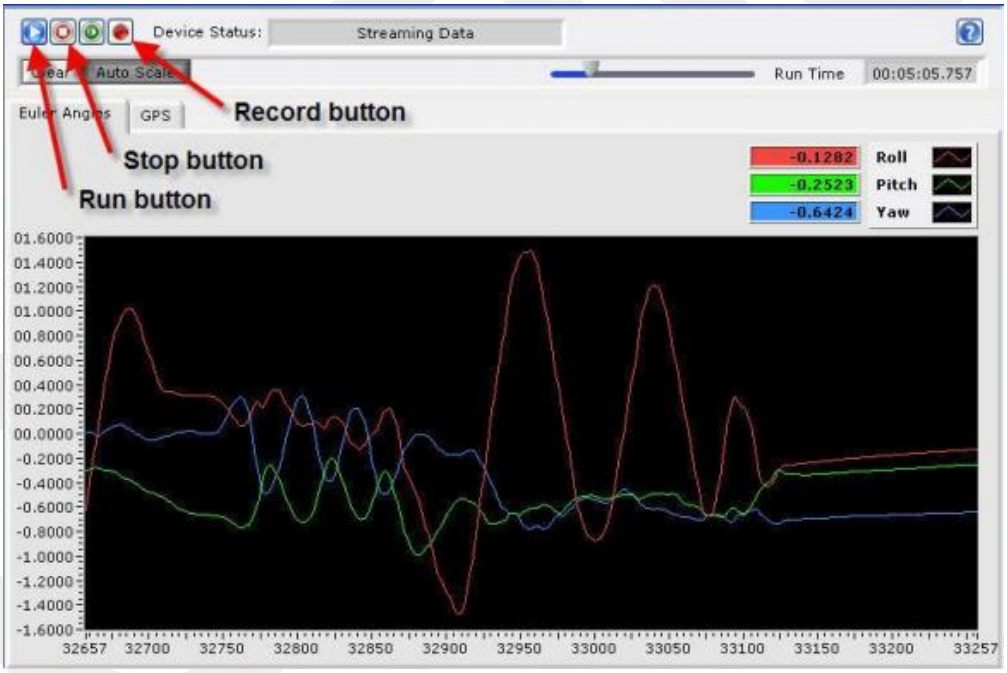

Figure 41: Microstrain Data Monitor Software [60]

In Figure 42, the period of oscillation on X axis is found 1.25 s. To determine the moment of inertia about rotation axis, the following formula is used;

$$J = \left[\frac{T_n}{2\pi}\right]^2 \frac{mgR^2}{L} \qquad \textbf{(5.1)}$$

Table 3- Parameters for Roll (Jxx) Axis Inertia Test

Length [L]	0.90 m
Radius [R]	0.40 m
Mass [m]	2.7 kg
Gravity [g]	9.8 m/s^2
Period [T_n]	1.25 s

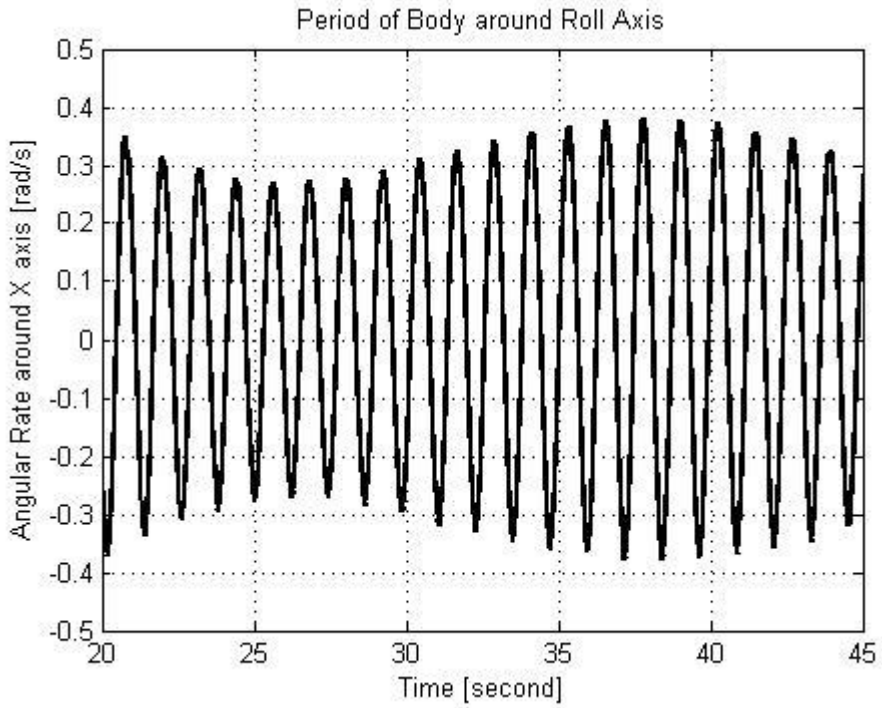

Figure 42: Period of Body around Roll Axis

By using Equation 5.1, Roll axis (J_{xx}) moment of inertia is 0.0457$kg.m^2$. Quadrotor system is symmetric, moment of inertia about Roll (J_{xx}) and Pitch (J_{yy}) are assumed to be same.

Same experiment processes is performed for Yaw axis (j_{zz}) . Figure 43 shows the experimental setup for Yaw (J_{zz}) axis.

Figure 43: Inertia Test on Yaw (Jzz) axis

In Figure 44, the period of oscillation on Z axis is found 1.7 s. By using Equation 5.1, Yaw axis (J_{zz}) moment of inertia is 0.0846$kg.m^2$.

Table 4- Parameters for Yaw (Jzz) Axis Inertia Test

Length [L]	0.90 m
Radius [R]	0.40 m
Mass [m]	2.7 kg
Gravity [g]	9.8 m/s^2
Period [T_n]	1.7s

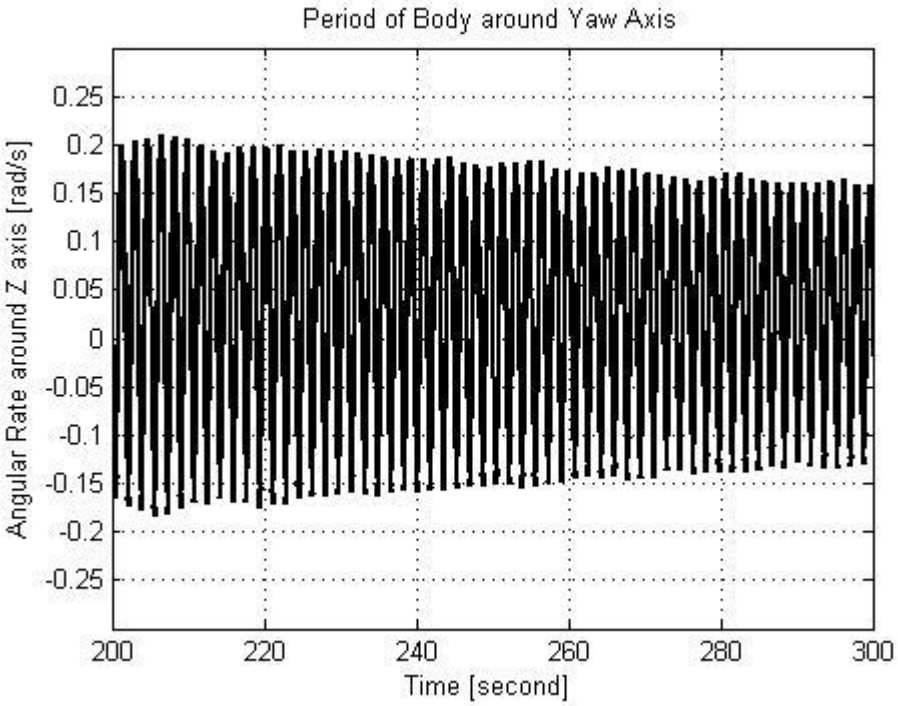

Figure 44: Period of Body around Roll Axis

Experimentally obtained moment of inertia values about rotation axes are given in Table 5;

Table 5- Moment of Inertias

Roll (J_{xx})	$0.0457 kg.m^2$
Pitch (J_{yy})	$0.0457 kg.m^2$
Yaw (J_{zz})	$0.0846 kg.m^2$

5.1.3 Test Bench

In order to test quadrotor system and tune designed controller parameters safely, a test bench is designed and built (Figure 45). Test bench is designed in CAD environment (Figure 46) and built by using aluminum square profile materials.

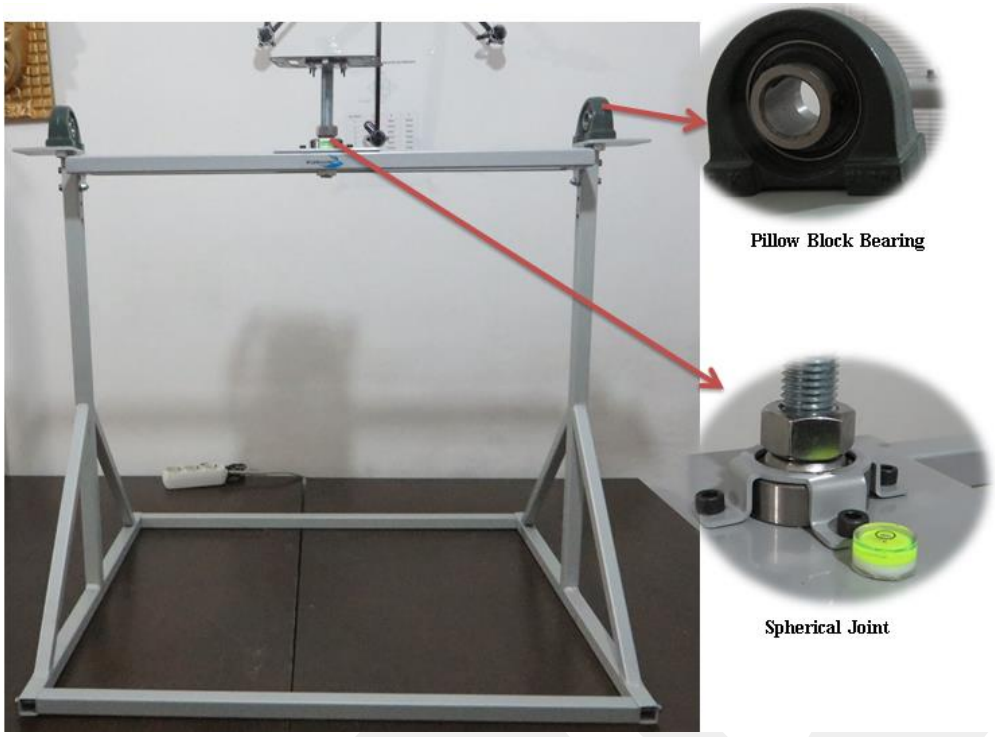

Figure 45: Test Bench

Figure 46: CAD model of Test Bench

This test setup can be used in one axis (Figure 47) and three axes (Figure 48) test procedures by the help of two universal pillow block bearings and one spherical bearing.

Figure 47: One Axis Test Bench

Figure 48: Three Axes Test Bench

Test bench physical parameters are given in Table 6.

Table 6- Test Bench Physical System Parameters

Width	800 mm
Depth	500 mm
Height	650 mm
Mass	4785.2 gr

5.1.4 Active Landing Gear Mechanism

Quadrotor system has two active landing gears on Roll axis. Each of them consist of one servo motor, one 'C' type servo bracket, one multi-purpose servo bracket, one rod and extra weights (Figure 49), active landing gear is shown.

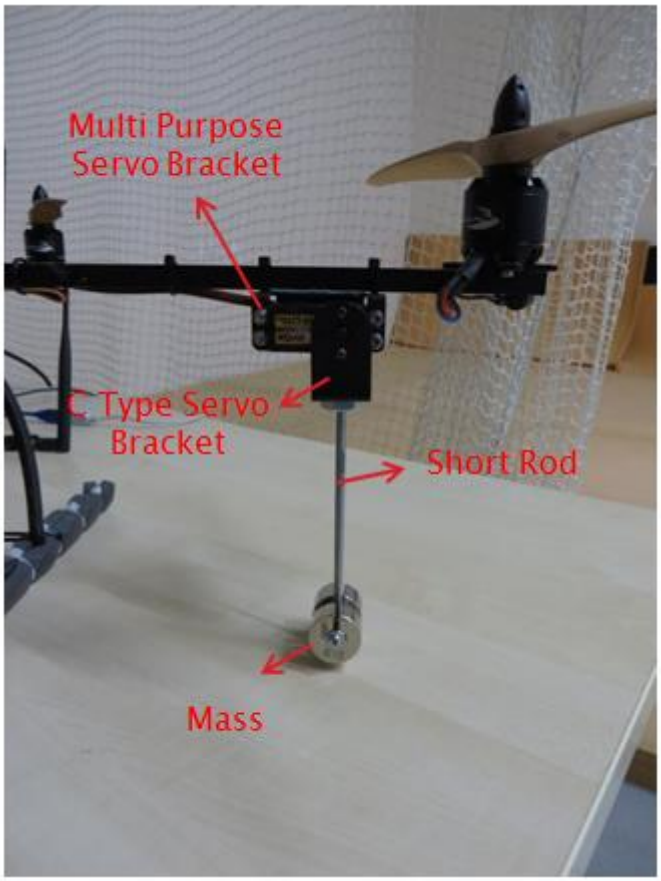

Figure 49: Active Landing Gear Mechanism

Active landing gear is modelled in CAD environment to integrate on the Quadrotor system. CAD model is given in Figure 50. The total mass of the each active landing

gear system is 112.95 gr. Three different rod lengths can be used (0.14m, 0.20m, and 0.30m). The extra weight can be moved up and down.

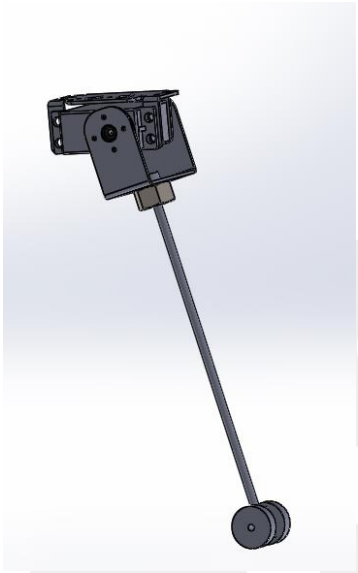

Figure 50: Active Landing Gear CAD Model

Servo brackets are produced by Lynxmotion Company [61] and servo motor which is used for actuating the gear is a standard digital Savöx SH-1290MG [78] (Figure 51), also technical specs are given in Table 7. Each servo motor is driven separately by using Flight Controller Boards.

Figure 51: SH-1290MG Standard Digital Servo [78]

Table 7- SH-1290MG Technical Specifications

Operating Voltage:	4.8-6.0 Volts
Operating Temperature Range:	-20 to +60 Degree C
Operating Speed (6.0V):	0.05sec/60° at no load
Torque (6.0V):	0.049 kg-m
Operating Angle:	100 degree (when 1000-2000 microsecond)
Current Drain (6.0V):	Idle Current: 5mA (at stopped) Stall Current: 3A (at locked)
Dimensions:	40.3 x 20.2 x 37.2mm
Weight:	56.4 gr

5.2 Flight Controller Hardware

In this thesis, the controller hardware is embedded onto the system. The controller hardware is based on an open source project which is called Multiwii [57]. The controller hardware consists on microcontroller unit, IMU unit, blackbox unit and input/output (I/O) ports. All of them are combined on the same controller card which is called NAZE32 [5] (Figure 52).

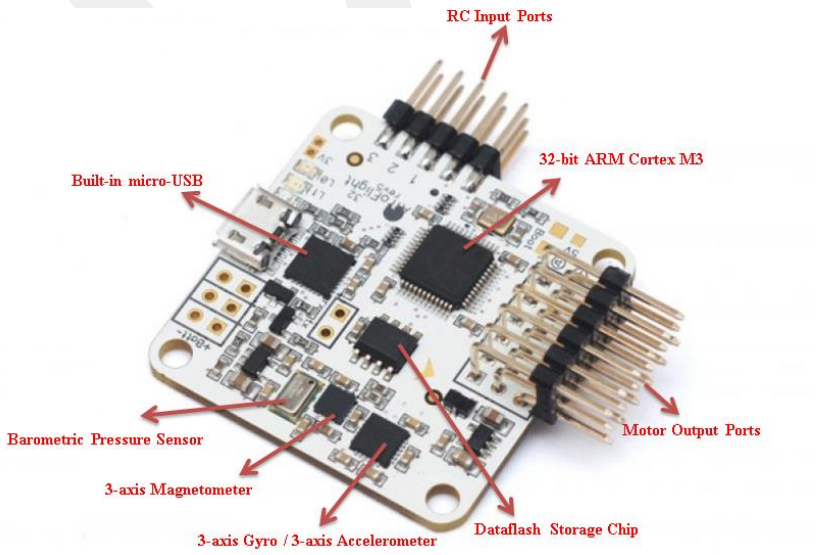

Figure 52: NAZE32 Flight Controller [5]

This controller board is used a 32-bit ARM Cortex M3 processor and has 3-axis Gyro, 3-axis Accelerometer, 3-axis Magnetometer and barometric pressure sensor which are used for IMU unit. It has a dataflash storage chip for logging flight data and RC receiver signal input ports and motor output ports. By using built-in micro-USB, it can be programmed easily without a third part programmer. In Figure 54, flight controller layout can be seen. Technical specs of the flight controller card are given in Table 8.

Table 8- NAZE32 Flight Controller Technical Specs

Processor:	STM32F103CBT6 32-bit ARM Cortex M3 processor (72MHz, 3.3V)[62]
Gyro / Accelerometer Sensor:	Invensense MPU6050 [63]
Magnetometer Sensor:	Honeywell HMC5883L [64]
Barometric Pressure Sensor:	Meas-Spec MS5611[65]
Dataflash Storage Chip:	Micron M25p16 (2MB)[66]
Supported RC Inputs:	Standard, PPM Sum, Spektrum
Output Port Number	6
Input Port Number	8
Operating Voltage:	3.3V
Dimensions:	36x36mm
Mounting holes:	30.5mm
Weight:	8gr
Operating Temperature Range:	-20 to +60 Degree C

5.3 Voltage and Current Sensor

In order to measure the quadrotor system's voltage and current draw, a voltage current sensor is used (Figure 53). This sensor is named as ATTOPILOT sensor [67]. The technical specs are given in Table 9.

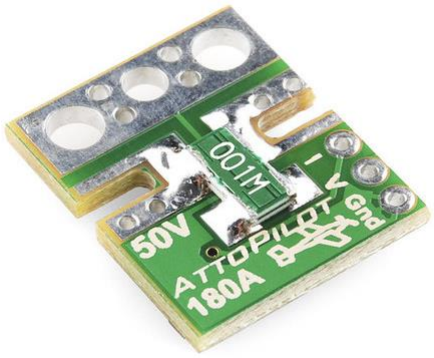

Figure 53: Voltage / Current Sensor [67]

Table 9- ATTOPILOT Voltage /Current Sensor

Max. Working Voltage Value:	50V
Max. Working Current Value:	180A
Dimensions:	4x15x19 mm
Analog Output Scaled:	3.3V
Sensed Parameters:	63.69mV/Volt – 18.3mV/Amp
Operating Temperature:	-55 to +125 Degree C

5.4 Telemetry and RC Transmitter Units

Quadrotor system has telemetry units for wireless communication. By the help of those units, flight data can be logged into the PC, also controller tuning can be done wireless. For telemetry unit, two Xbee Pro900 [68] units are used (Figure 54).

Figure 54: Xbee Pro 900 [68]

Technical specs of the Xbee Pro900 are given in Table 10.

Table 10- Xbee Pro 900Technical Specs

Operating Voltage / Current:	3.3V / 210mA
Power Output:	50mW
RD Data Rate:	Fast 156Kbps
Range:	10km (RF LOS with high antennas)
Operating Frequency:	900 MHz
Encryption:	128 bit
Operating Temperature Range:	-40 to +85 Degree C

A RC transmitter system is used to manually control the quadrotor and send some control commands. An 8-channel 2.4Ghz Graupner MX-16 transmitter [69] unit is used (Figure 55). This transmitter has PPM sum signal output function and rechargeable battery.

Figure 55: Graupner MX16 Transmitter[69]

Technical specs of the transmitter are given in Table 11.

Table 11-Graupner MX16 Transmitter Technical Specs

Frequency band:	2.4Ghz
Modulation:	FHSS
Control Functions (Channel):	8
Operating Temperature Range:	-10 to +55 Degree C
Operating Voltage:	3.4V to 6V
Current draw:	180mA
Dimensions:	190x195x90mm
Weight:	770gr (with battery)

5.5 Actuation System

In order to actuate the system, four brushless motors are used to rotate the propellers. In this system, TurkUAV MT2814 750KV [70] brushless motor is used (Figure 56).

Figure 56: TurkUAV MT2814 750KV brushless motor [70]

Brushless motor technical specs are given in Table 12.

Table 12- TurkUAVMT2814 750KV Brushless Motor Technical Specs

KV Value:	750
Internal Resistance:	0.0063mΩ
No-Load Current:	0.65A
Max Current:	26.75A
Continuous Current:	18.9A
Max Watts:	375W
Li-po Cells:	3-4s
Weight:	94g
Dimensions:	35x36mm
Wire Length:	450mm
Shaft Diameter:	4.0mm
Recommended Prop:	10”x7/11”x4.7/12”x5
Recommended ESC:	30A

Also 12x5 Beachwood Propellers are used with that brushless motor (Figure 57).

Figure 57: 12 x 5 Beachwood Propellers

In order to drive brushless motor, Electronic Speed Controller (ESC) is used. In this system, TurkUAV 40A brushless ESC [71] is used (Figure 58). Brushless ESC technical specs are given in Table 13.

Figure 58: TurkUAV 40A Brushless ESC [71]

Table 13- TurkUAV 40A Brushless ESC Technical Specs

Input Voltage:	2-4S
Input Rate Rage:	50-400Hz
Output Current:	18.9A
Max Watts:	40A Continuous, 50A Burst (10 Sec)
BEC Output:	5V / 3A
Weight:	38g
Dimensions:	55x26x13mm

Tests results of actuation system are given in Table 14. Those test results are provided by the company. Information about that table is given in Chapter 3 in Rotor Dynamics section.

Table 14- Test Result of Actuation System

Throttle	Volts	Amps	Watts	RPM	Thrust (g)	Winding Temp (F/C)
25%	16.38	1.63	27	3219	248	91/33
50%	16.21	5.46	89	5224	687	88/32
75%	15.88	12.90	205	6972	1266	92/33
100%	15.33	24.76	380	8427	1850	122/50

5.6 Battery

The required power is provided by one 4S 4000mAh GensAce Li-Po battery [72] (Figure 59). Technical specs of battery are given in Table 15.

Figure 59: 4S 4000mAh Li-Po Battery [72]

Table 15- Battery Technical Specs

Voltage:	4S (14.8V)
Capacity:	4000mAh
Max. Continuous Discharge:	25C (100A)
Max. Burst Discharge:	50C (200A)
Weight:	438.4gr
Dimensions:	136x41x36mm
Balance Connector Type:	JST-XH

5.7 Controller Software

In this study, Cleanflight software is used with flight embedded controller (Naze32 rev5). This software is supported by a large RC community and developers. This software is an open source project and can be redistributed it and/or modified it under the terms of the GPL (GNU) General Public License [73] as published by the Free Software Foundation.

Cleanflight software has a graphical user interface [GUI] to calibrate sensor, edit controller parameters, graph the sensor data and log the data (Figure 60, and 61).

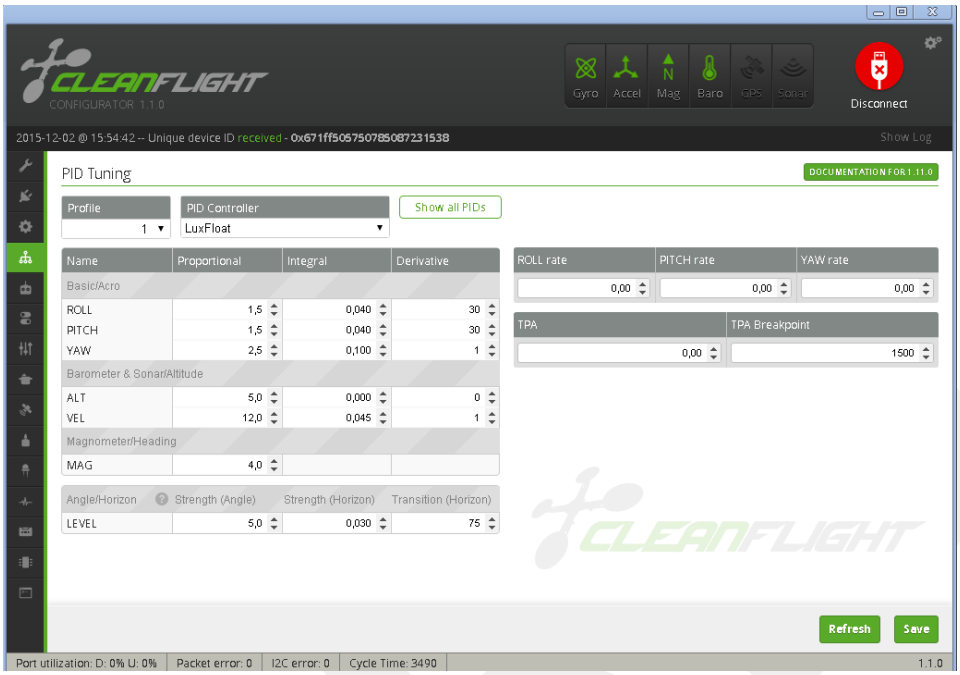

Figure 60: Controller Gain Edit Tab

Figure 61: Sensor Data Graph Tab

This GUI software is an open source project and can be redistributed it and/or modified it under the terms of the GPL (GNU) General Public License as published by the Free Software Foundation.

To edit controller software Notepad++ [74] which is a free C / C++ editor is used. C code modification is performed with this software (Figure 62).

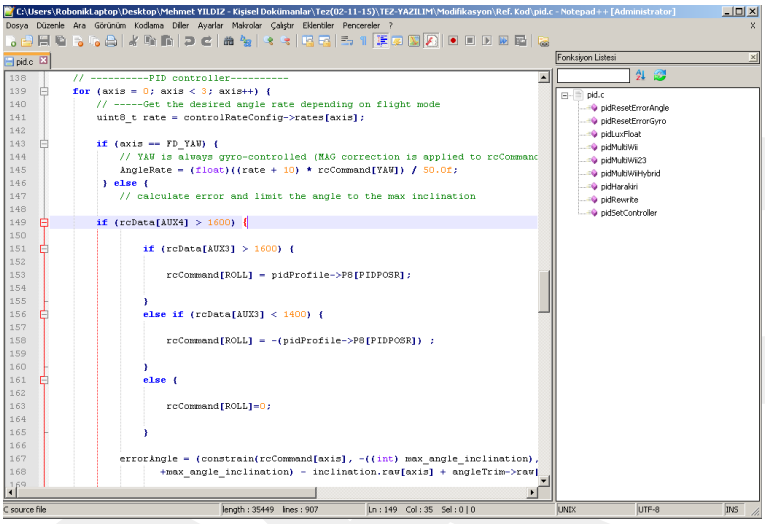

Figure 62: Notepad ++ Editor

The modified C code is compiled by using Cygwin software [75]. This is a large collection of GNU and Open Source tools which provide functionality similar to a Linux distribution on Windows. This software creates a hex file (Figure 63) and this file is loaded into the embedded controller by using Cleanflight GUI (Figure 64).

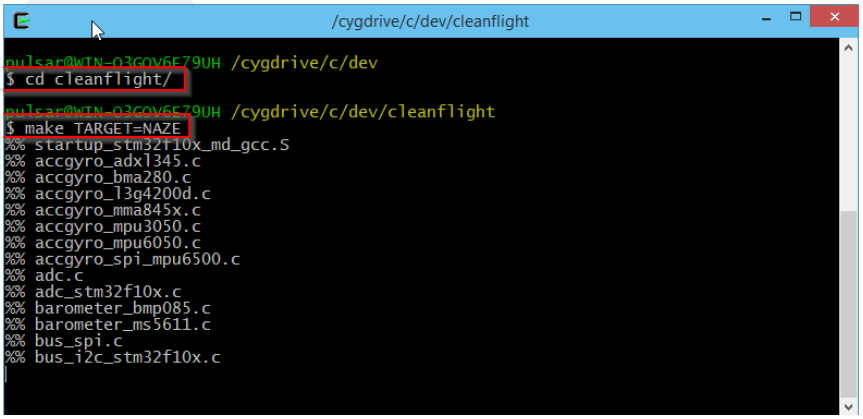

Figure 63: Cywing Software

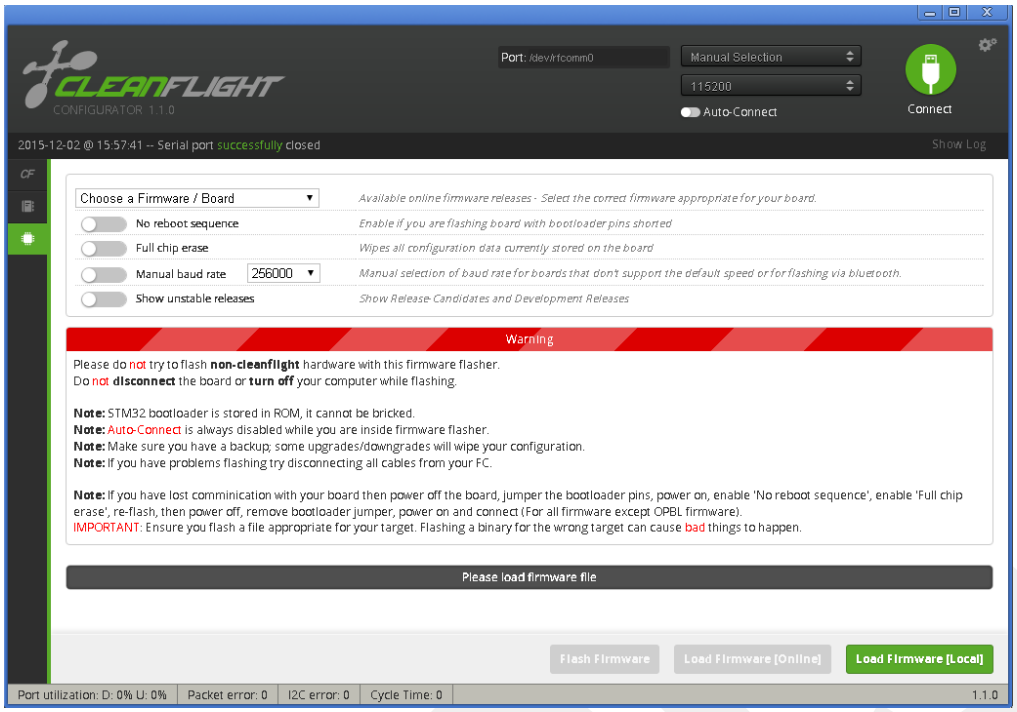

Figure 64: Firmware Upload Tab

CHAPTER 6
EXPERIMENTS

In this chapter, the results of active landing gear controller are explained in details. In order to perform the experiments on the physical system, a test system (Quadrotor) is prepared by using equipment which are discussed in Chapter 5 . Test system is given in Figure 65.

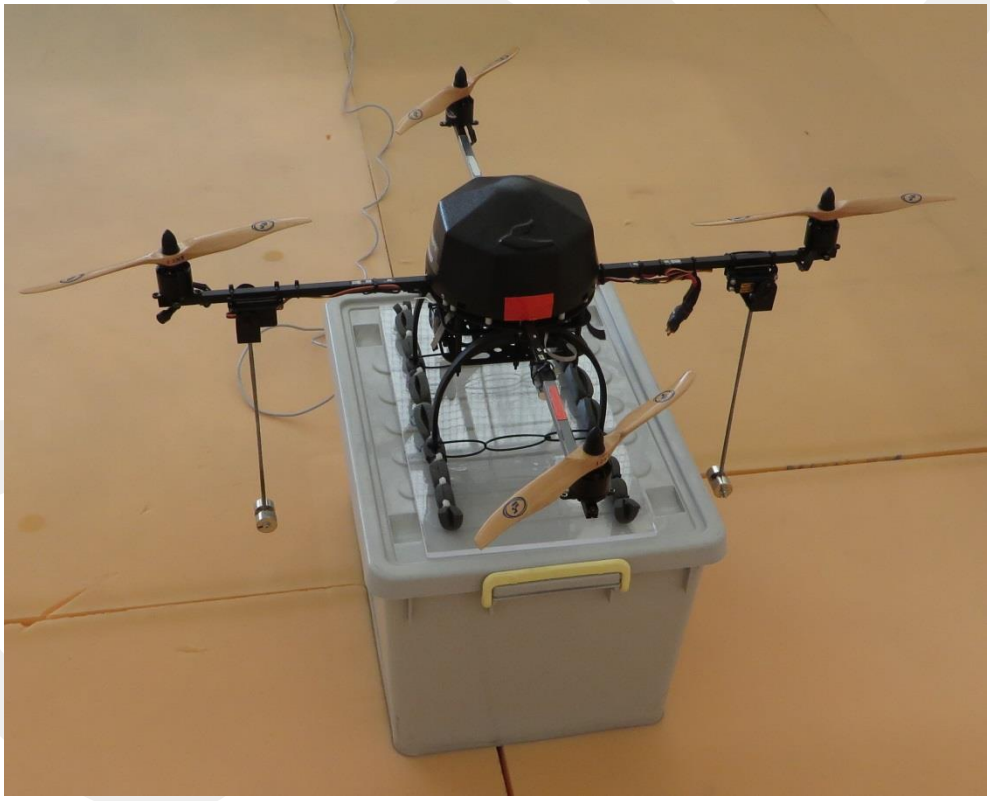

Figure 65: Quadrotor Test System

To investigate the results of the active landing gears on the attitude dynamics of, three different actions are considered. They are stability assistance, steering and improved agility modes. Details of the each mode are given in below.

6.1 Assistance Mod

This mode is simulated in Section 4.5. Simulations show that by the help of the counter moment which caused by gears about the body x-axis, relative stability of roll dynamics is enhanced and disturbance rejection ability is improved. Due to the limitations of the test arena in horizontal plane, tests are performed in an unusual way. Gear controller acts as a regulator and acts to regulate the roll angle and angular velocity at zero. This controller can be modified to enhance the tracking capability of the system. However, regulatory action is more representative to monitor the effects of landing gears. A reference input is given for roll axis from transmitter. As the gears are active in improved stability mode, vehicle resists to the reference and the bandwidth is reduced due to such a 'virtual' disturbance. Propulsion actuators try to track the reference. However, regulatory control of the gears tries to reject this virtual disturbance. Test details are given in below and test result is given in Figure 66.

Gears Variables

L_m; length of landing gear rods = 0.20 m

m; mass of gears = 0.14 kg

h; the distance between O and O_1 points (O and O_2 points) = 0.17 m

Gears Controller Gain

K_p; angle gain = 30 ;

K_d; angle rate gain= 3 ;

LuxFloat PID Controller Gain

Kp_Level= 3; Kp=2.5; Ki= 0.45; Kd= 0.045

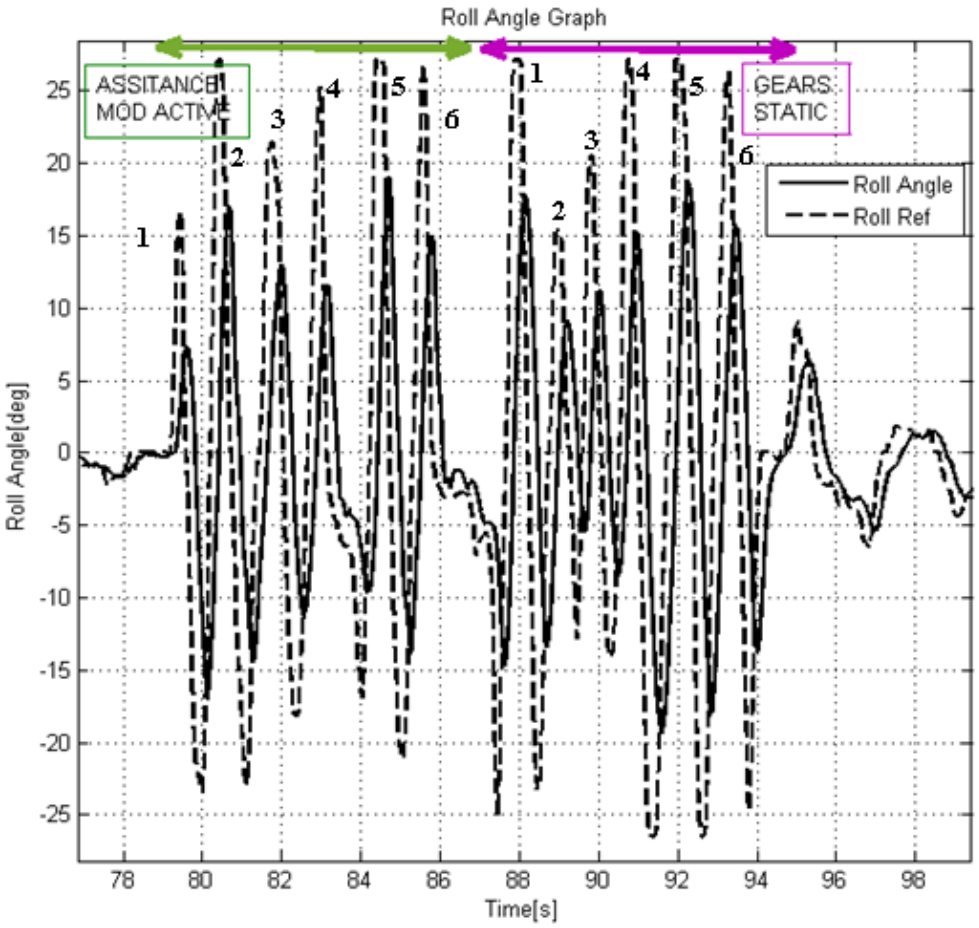

Figure 66: Assistance Mod Results

To observe the effect of the active landing gears, the percentage of the differences between the roll angle and reference input for active landing gear assistance mode active and static gear case can be compared. Those compared values are given in Table 16.

Table 16- Assistance Mod Results

	Assistance Mod Active			**Gears Static**		
1	Reference	16.8 deg / 79.42s	%43	Reference	27.2 deg / 87.97s	%65
	Controller	7.3 deg / 79.62s	0.2s	Controller	17.7 deg / 88.14s	0.17s
2	Reference	27.1 deg / 80.41s	%62	Reference	15.5 deg / 88.99s	%58
	Controller	17 deg / 80.67s	0.26s	Controller	9.1 deg / 88.14s	0.22s
3	Reference	21.4 deg / 81.7s	%60	Reference	20.5 deg / 89.91s	%55

	Controller	13 deg / 81.99s	0.29s	Controller	11.3 deg / 90.01s	0.2s
4	Reference	25.3 deg / 82.96s	%43	Reference	27.2 deg / 90.76s	%55
	Controller	11.6 deg / 83.14s	0.18s	Controller	15.2 deg / 90.95s	0.19s
5	Reference	27.2 deg / 84.39s	%70	Reference	27.2 deg / 92s	%68
	Controller	19 deg / 84.69s	0.3s	Controller	18.6 deg / 92.28s	0.28s
6	Reference	26.9 deg / 85.57s	%50	Reference	26.4 deg / 93.24s	%59
	Controller	15 deg / 85.78s	0.21s	Controller	15.7 deg / 93.46s	0.22s
	Mean : %55 / 0.24 s			Mean: %60 / 0.21 s		

According to Table 16, when active landing gear is used as assistance mode, system reacts more resistive and sluggish than gears static mod. When the assistance mode is active, system gives the response to the reference commend %5 less than gear static. Also there is a 0.03 s time differences between assistance mode and gear static mod. This means that quadrotor with active landing gear reacts more slow than the quadrotor with static legs.

6.2 Steering Mode

In this mode, by moving the gears position to the left and right, system can be steered by only the help of the gears movement. Position of the gears can be controller via Transmitter directly (Figure.67). Active landing gears on the left and right in steering mode is given in Figures 68 and 69.

Figure 67: Gear Positions via Transmitter

Figure 68: Gears on the Left

Figure 69: Gears on the Right

Test details are given in below and test result is given in Figure 70.

Gears Variables

L_m; length of landing gear rods = 0.20 m

m; mass of gears = 0.14 kg

h; the distance between O and O_1 points (O and O_2 points) = 0.17 m

LuxFloat PID Controller Gain

Kp_Level= 3; Kp=2.5; Ki= 0.45; Kd= 0.045

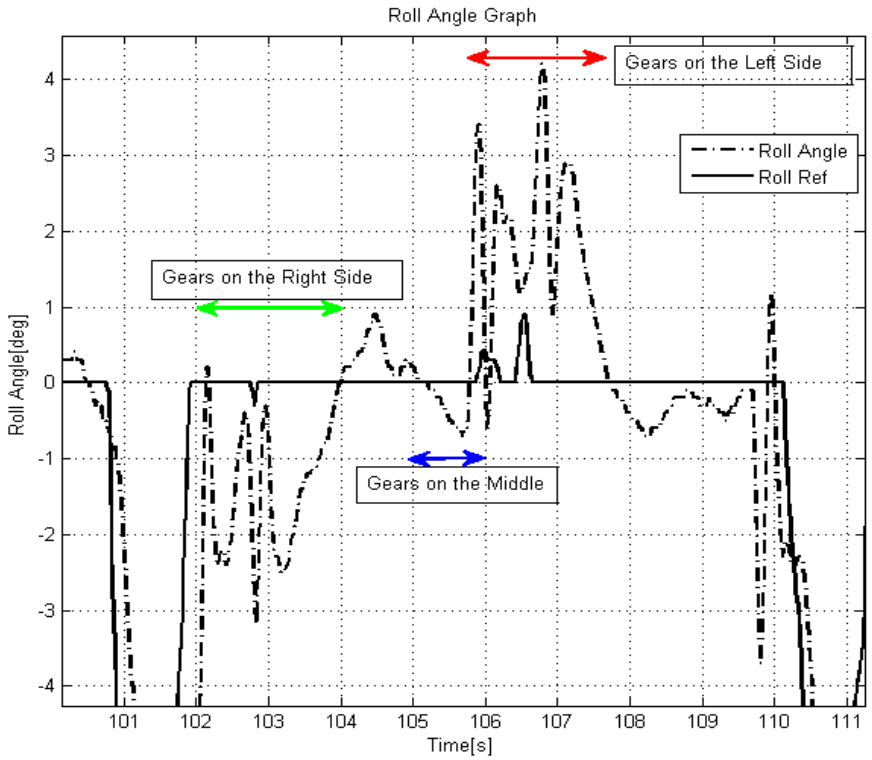

Figure 70: Steering Mode Results

By moving the gears, system can be steered without giving a reference input via transmitter. In Figure 70, it is clearly seen that when the gears move left or right side, system's roll dynamics tends to the left or right side around +-3 degree and it causes to move in the body x-axis of the quadrotor. Even the roll axis is controlled via propulsion unit controller; quadrotor tends to slide in horizontal plane due to the action of gears. This makes the system steer by using the gears.

6.3 Improved Agility Mod

This mode operates in the opposite way as that of the assistance mode. The difference with assistance mod is that gears are creating momentum in the opposite direction of the assistance mode to make the Quadrotor more agile in roll axis. Test details are given in below and test result is given in Figure 71.

Gears Variables

L_m; length of landing gear rods = 0.20 m

m; mass of gears = 0.14 kg

h; the distance between O and O_1 points (O and O_2 points) = 0.17 m

Gears Controller Gain

K_p; angle gain = 20 ;

K_d; angle rate gain= 0 ;

LuxFloat PID Controller Gain

Kp_Level= 3; Kp=2.5; Ki= 0.45; Kd= 0.045

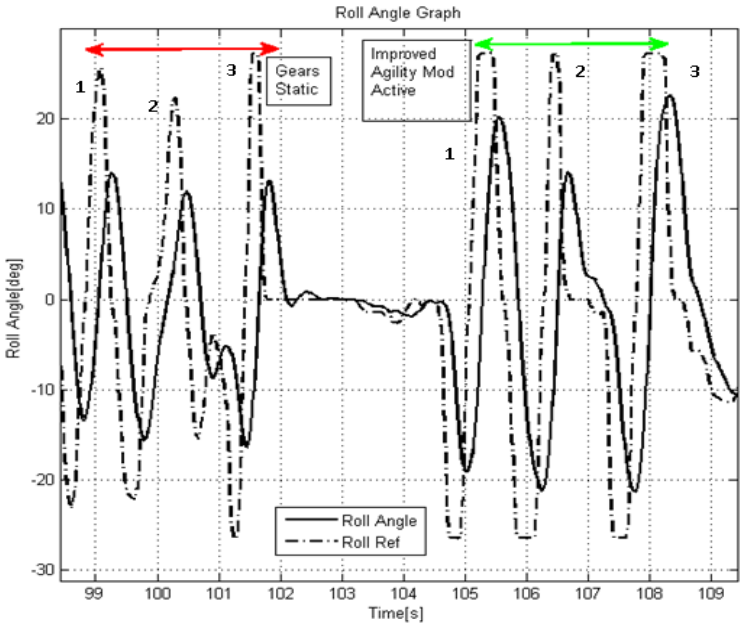

Figure 71: Improved Agility Mod Results

As the case of assistance mode, similar experiment is performed. A reference input is given for roll axis from transmitter. To observe the effect of the active landing gears, the percentage of the differences between the controller output and reference values for active landing gear improved agility mod active and gear static (passive) can be compared. Those compared values are given in Table 17.

Table 17- Improved Agility Mod Results

	Gears Static			**Improved Agility Mod Active**		
1	Reference	25.7 deg / 99.09s	%54	Reference	27.2 deg / 105.3s	%85
	Controller	13.9 deg / 99.28s	0.19s	Controller	24.3 deg /105.5s	0.2s
2	Reference	22.3 deg / 100.3s	%53	Reference	27.2 deg / 106.5s	%50
	Controller	11.9 deg / 100.5s	0.2s	Controller	13.8 deg / 106.7s	0.2s
3	Reference	27.2 deg / 101.6s	%48	Reference	27.2 deg / 108.1s	%83
	Controller	13.2 deg / 101.8s	0.2s	Controller	22.8 deg / 108.3s	0.2s
	Mean : %51.6 / 0.2 s			Mean: %72 / 0.2 s		

According to Table 17, when active landing gear is used as improved agility mode, effort to track the reference roll angle is better than the case with static gears.

CHAPTER 7
CONCLUSION AND DISCUSSION

This thesis is the starting point of a study that aims to investigate the effect of the retractable (active) landing gear on Quadrotor's attitude dynamics. At first, system is modeled. Active landing gears create moments only Quadrotor body x-axis and not considered to have an effect on other two axes.

A commercial Quadrotor frame is used in this thesis. Frame and motor-drive-propeller parameters are given by producer, but the moment of inertia of the system is derived experimentally. Those parameters are used in deriving mathematical modelling. Also a servo mechanism with rod and mass is used as active landing gears.

The scope of the thesis covers to design a Quadrotor test bench with embedded flight controller and software. One of the most popular embedded flight controller boards which is called Naze32 is used in this thesis. This controller board is composed of microcontroller unit, IMU unit, blackbox unit and input/output (I/O) ports. For software, an open source project which is called CleanFlight is used. By the help of the Graphical User Interface (GUI) software, modified code can be loaded into the system and parameters can be changed, graphed and logged easily.

Cleanflight software uses a cascaded PID controller namely "LuxFloat". In this thesis, this control structure is used to control of the attitude dynamics of the Quadrotor. Luxfloat control architecture is analyzed and modeled in Simulink.

Initially LuxFloat PID controller gains are tuned by trial and error method on single axis Quadrotor test bench. The behavior of the Quadrotor on the test bench is modeled in simulation environment as well. LuxFloat digital controller model is applied to this model. Tuned controller parameters are employed on the model. In

addition, the behavior of the system as it is coupled to the test bench thru its COG is modeled and digital controller is applied to this model as well. The model and the actual responses are compared. It is seen that simulated rotation behavior about center of gravity is somehow different from responses of actual system and the one that simulates the behavior in case of pivot at O. Uncertain parameters such as viscous damping coefficient, test bench moment of inertia and connection distance to the COG effects which are used in test bench model are determined by applying optimization methods. Least squares minimization by genetic algorithm is applied to minimize the difference between the actual response and the model response.

In addition to the abovementioned models, a non-linear 3D Quadrotor dynamics model with the effects of active landing gears is obtained. However, it is not used for controller design purposes. It can be utilized for design and simulation purposes within the further studies.

Effect of the landing gears about Quadrotor roll axis is simply modeled as a moment term and a regulatory controller is designed. Roll angle and roll rate are fed back and multiplied by constant gains to regulate the roll angle and angular rate. The output of this controller manipulates angular position of the landing gears. In simulations, this regulatory nature assists to reject the disturbances. This controller structure is formed in C language and loaded into the embedded controller board by modifying the active gimbal control subroutine.

After simulations, flight tests are applied to Quadrotor system. Three different actions are considered for active landing gears during tests. The first one is assistance mode. Active landing gears are controlled to improve the stability. Flight test results show that active landing gears create a counter moment to regulate the roll angle and roll rate at zero. This action assists to reject disturbances. The second one is the steering mode. In this mode, by shifting the gears to the left and right terminal positions, quadrotor system is intended to steer. Flight tests presented the expected behavior. The last mode is the improved agility mode. This mode can be considered the opposite of the assistance mode. Active landing gears create a moment in same direction to the propeller moment in this mode. Flying test results shows that

Quadrotor with active landing gears becomes more agile than Quadrotor with static gear by increasing the bandwidth of the closed loop roll dynamics.

To conclude, simulation results and flying test results clearly show that active landing gears have critical effects on attitude dynamics of the Quadrotor. These effects can be used for various purposes. One of the most important application areas may be the redundancy flight controller algorithm, i.e., when a motor or motor driver failure has occurred, active landing gear would be used to compensate the attitude dynamics till emergency landing succeeds. Disturbance rejection ability may be utilized under severe external disturbances such as wind gusts.

In the future studies, embedded controller software flowchart and function list should be obtained. It will help to understand the code and modifications will be easily handled. Altitude controller should be studied within the embedded controller as well. During the flight tests, the lack of the altitude controller creates some difficulties. The digital Luxfloat controller would be applied on the non-Linear model of the Quadrotor with active landing gear. This will give more realistic simulations for various maneuvers.

The existing mechanism should be improved and must be brought to a more realistic landing gear form. Instead of servo motors, brushless dc motors with encoders can be used to get a faster response with higher accuracy. A second embedded controller card may be used for the control of active landing gears. This embedded controller should exchange data with main flight controller. By using second controller, more advanced gear control algorithms such as active disturbance rejection control, control allocation, etc. can be applied.

REFERENCES

[1] DJI Phantom, 2010, http://www.dji.com/product/phantom

[2] V. Kumar, Robots that fly and cooperate, GRASP, University of Pennsylvania, access 01.03.2012 available: https://www.youtube.com/watch?v=4ErEBkj_3PY

[3] R. D'Andrea, The astounding athletic power of quadrocopters, Zürich University, access 11.06.2013 available: https://www.youtube.com/watch?v=w2itwFJCgFQ

[4] DJI S900, 2014, http://www.dji.com/product/spreading-wings-s900

[5] NAZE32, 2013, http://www.abusemark.com/downloads/naze32_rev3.pdf

[6] Multiwii Project, 2011, http://www.multiwii.com

[7] D. Mellinger, N. Michael,and V. Kumar, 2012 , "Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors", The International Journal of Robotics Research April 2012 vol. 31 no. 5 pp.664-674 (Download Date:12-01-2014)

[8] Mikrokopter, 2005, www.mikrokopter.de

[9] S. Bouabdallah, P. Murrieri, and R. Siegwart, 2004, "Design and control of an indoor micro Quadrotor", In Proc. of ICRA, Vol. 5, pp.4393–4398, Barcelona, Spain (Download Date: 10-12-2013)

[10] R. Czyba, 2009, "Design of Attitude Control System for an UAV Type-Quadrotor Based on Dynamic Contraction Method", IEEE/ASME International Conference on Advanced Intelligent Mechatronics Suntec Convention and Exhibition Center Singapore, July 14-17, 2009. (Download Date: 11-10-2013)

[11] S. Kim, S. Choi, and H. J. Kim, 2013, "Aerial Manipulation Using a Quadrotor with a Two DOF Robotic Arm", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Tokyo, Japan, November 3-7, 2013. (Download Date: 06-12-2013)

[12] A.E. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero,and R. Cano, 2013, "Control of an aerial robot with multi-link arm for assembly tasks", IEEE

International Conference on Robotics and Automation (ICRA) Karlsruhe, Germany, May 6-10, 2013. (Download Date: 06-12-2013)

[13] P. Castillo, R. Lozano, and A. Dzul, 2005, "Stabilization of a mini rotorcraft with four rotors", Control Systems Magazine, IEEE, 25, 45–55 (Download Date: 11-10-2013)

[14] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, 2007, "Quadrotor helicopter flight dynamics and control" Theory and experiment In AIAA Guidance, Navigation and Control Conference and Exhibit Reno, NV, USA. (Download Date: 11-10-2013)

[15] S. Bouabdallah, P. Murrieri, R. Siegwart, 2005,"Towards autonomous indoor micro VTOL", Autonomous Robots, 18(2), pp. 171-183, 2005. (Download Date: 11-10-2013)

[16] A.Tayebi, S. McGilvray, 2006, "Attitude Stabilization of a VTOL Quadrotor Aircraft", IEEE Transactions on Control Systems Technology, Vol. 14, No. 3, pp.562-571 , May 2006. (Download Date: 03-11-2013)

[17] P. Pounds, R. Mahony,and P. Corke, 2010, "Modelling and Control of a Large Quadrotor Robot", Control Engineering Practice 18, 2010, pp.691–699. (Download Date: 03-11-2013)

[18] K. U. Lee, Y. H. Yun, W. Chang, J. B. Park, and Y. H. Choi, 2011, "Modeling and Altitude Control of Quad-rotor UAV", 11th International Conference on Control, Automation and Systems Oct. 26-29, 2011 in KINTEX, Gyeonggi-do, Korea. (Download Date: 08-12-2013)

[19] E. Ö. EFE, 2007, "Dört Motorlu Bir Dönerkanat Sisteminin Dinamik Modeli ve PD Kontrolör Yörünge Kontrolü", TOK'07 Declaration Book, pp. 194-199,September 5-7, Istanbul, TURKEY (Download Date: 05-01-2014)

[20] D. Mellinger, Q. Lindsey, M. Shomin, and Vijay Kumar, 2011, "Design, Modeling, Estimation and Control for Aerial Grasping and Manipulation", IEEE/RSJ International Conference on Intelligent Robots and Systems, September 25-30, 2011, San Francisco, CA, USA. (Download Date: 12-12-2013)

[21] M. Orsag, C. Korpela, and P. Oh, 2012, "Modeling and Control of MM-UAV: Mobile Manipulating Unmanned Aerial Vehicle", Journal of Intelligent & Robotic Systems, Volume 69, Issue 1-4, pp 227-240. (Download Date: 12-12-2013)

[22] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, 2011, "Grasping From the Air: Hovering Capture and Load Stability", IEEE International Conference on Robotics and Automation Shanghai International Conference Center May 9-13, 2011, Shanghai, China. (Download Date: 12-12-2013)

[23] M. Bisgaard, A. Cour-Harbo,and J. D.Bendtsen, 2010, "Adaptive control system for autonomous helicopter slung load operations", Control Engineering Practice 18, 2010 pp.800–811. (Download Date: 12-12-2013)

[24] ARCAS: Aerial Robotics Cooperative Assembly System, 2011, http://www.arcas-project.eu/

[25] I. Palunko, R. Fierro, 2011, "Adaptive Control of a Quadrotor with Dynamic Changes in the Center of Gravity", Preprints of the 18th IFAC World Congress August 28 - September 2, 2011, Milano, Italy. (Download Date: 14-12-2013)

[26] V. Ghadiok, J. Goldin,and W.Ren, 2011, "Autonomous Indoor Aerial Gripping Using a Quadrotor", IEEE/RSJ International Conference on Intelligent Robots and Systems September 25-30, 2011. San Francisco, CA, USA. (Download Date: 14-12-2013)

[27] C. Korpela,M. Orsag,M. Pekala, and P. Oh, 2013, "Dynamic Stability of a Mobile Manipulating Unmanned Aerial Vehicle", IEEE International Conference on Robotics and Automation (ICRA); May 6-10, 2013, Karlsruhe, Germany. (Download Date: 14-12-2013)

[28] E. Beyer, M. Costello, 2008, "Performance of A Hopping Rotochute", 34 th Europan Rotorcraft Forum September 16-19, 2008, Liverpool, ENGLAND. (Download Date: 22-11-2013)

[29] C. Korpela,M. Orsag,M. Pekala, T. Danko, B. Kobe, C. McNeil, R. Pisch, and P. Oh, 2012, "Flight Stability in Aerial Redundant Manipulators", International Conference on Robotics and Automation RiverCentre, May 14-18, 2012,Saint Paul, Minnesota, USA. (Download Date: 21-12-2013)

[30] T. Lan, Q. Fei, Q. Geng, and Q. Hu, 2012, "Control Algorithm Research Of Quadrotor System Based on 3-DOF Simulation Platform", Automatic Control and Artificial Intelligence (ACAI 2012), pp. 697 – 700, March 3-5, 2012, Xiamen. (Download Date: 10-01-2014)

[31] Y. M. Al-Younes, M. A. Al-Jarrah,and A. A. Jhemi, 2010, "Linear vs. Nonlinear Control Techniques for a Quadrotor Vehicle", Proceeding of the

7th International Symposium on Mechatronics and its Applications (ISMA10), April 20-22, 2010 , Sharjah, UAE. (Download Date: 10-01-2014)

[32] D. Lee, V. Chitrakaran, T. Burg, D. Dawson,and B. Xian, 2007, " Control of a Remotely Operated Quadrotor Aerial Vehicle and Camera Unit Using a Fly-The-Camera Perspective", Proceedings of the 46th IEEE Conference on Decision and Control New Orleans, Dec. 12-14, 2007, LA, USA. . (Download Date: 25-12-2013)

[33] H. Yang, D. Lee, 2013, "Preliminary Results on Quadrotor Manipulation Control", 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI) October 31-November 2, 2013 / Ramada Plaza Jeju Hotel, Jeju, Korea. (Download Date: 25-12-2013)

[34] X. Gong, Y. Tian, Y. Bai ,and C. Zhao, 2012, "Trajectory Tacking Control of a Quadrotor Based on Active Disturbance Rejection Control", Proceeding of the IEEE International Conference on Automation and Logistics, August 2012, Zhengzhou, China. (Download Date: 10-10-2013)

[35] C. Peng, Y. Tian, Y. Bai, X. Gong, C. Zhao, Q. Gao,and D. Xu, 2013, "ADRC Trajectory Tracking Control Based on PSO Algorithm for a Quad-Rotor", Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on Melbourne, VIC, pp 800-805 ,19-21 June 2013. (Download Date: 10-10-2013)

[36] G. Arleo, F. Caccavale, G. Muscio and F. Pierri, 2013, "Control of Quadrotor Aerial Vehicles Equipped with a Robotic Arm", 21st Mediterranean Conference on Control & Automation (MED) Platanias-Chania, June 25-28, 2013, Crete, Greece. (Download Date: 27-12-2013)

[37] I. Palunko, R. Fierro, and P. Cruz, 2012, "Trajectory Generation for Swing-Free Maneuvers of a Quadrotor with Suspended Payload: A Dynamic Programming Approach", Robotics and Automation (ICRA), 2012 IEEE International Conference on; May 14-18, 2012, Saint Paul, MN. (Download Date: 27-12-2013)

[38] V. Matteo, 2013, "Modeling, Identification and Navigation of Autonomous Air Vehicles", Maste's Degree Project, Automatic Control Scholl of Electrical Engineering, Kungliga Tekniska Högskolan, May 2013, Stockholm, Sweden (Download Date: 27-04-2015)

[39] S. Micheal David, 2011, “Simulation and Control of a Quadrotor Unmanned Aerial Vehicle”, University of Kentucky Master’s Theses, April 2011, Kentucky, USA (Download Date: 28-04-2015)

[40] E. Mohamed Nabil, 2014, “Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches”, University of the American Master’s Theses, April 15, 2014, Cairo, Egypt (Download Date: 10-02-2015)

[41] B. Randal, 2008, “Quadrotor Dynamics and Control”, Brigham Young University, Utah, ABD (Download Date: 10-02-2015)

[42] N. Amr, S. Sajad, T.Carl, S. Mae, and L. Howard, 2012, “Control and Navigation Framework for Quadrotor Helicopters”, Journal of Intelligent & Robotic Systems, April 2013, Issue 1-4, Vol. 70, pp. 1-12, Springer, Netherlands. (Download Date: 11-02-2015)

[43] B. Samir, 2007, “Design and Control of Quadrotors with Application to Autonomous Flying”, Phd. Thesis, Ecole Polytechnique Federale de Laussane, Switzerland (Download Date: 11-12-2014)

[44] Küçük Doğanç, 2010, “Design of Two Wheeled Twin Rotored Hybrid Robotic Platform”, University of Atılım Master’s Theses, July, 2010, Ankara, Turkey (Download Date: 11-08-2014)

[45] G. Anıl, 2012, “Attitude and Altitude Control of an Outdoor Quadrotor”, University of Atılım Master’s, July, 2012, Ankara, Turkey (Download Date: 15-08-2014)

[46] C. Roman, and S. Grzegorz, 2012 “Control Structure Impact on the Flying Performance of the Multi-Rotor VTOL Platform-Design, Analysis and Experimental Validation”, International Journal of Advanced Robotic Systems, Vol. 10, pp. 62-70, January 22, 2013, Intech

[47] Ascending Technologies GmbH (AscTec), http://www.asctec.de.

[48] Tai Shih Hobby Corporation, 1996, http://www.gaui.com.tw

[49] Gumstix Inc., 2003, https://www.gumstix.com/about-us/

[50] Arducopter, 2008, http://copter.ardupilot.com/

[51] Vicon Motion Systems Ltd, 1980, http://www.vicon.com/

[52] Phoenix Technologies Inc. , 1996, http://www.ptiphoenix.com/

[53] OptiTrack, 2012, http://www.naturalpoint.com/optitrack/

[54] ROS on MATLAB , 2012, http://wiki.ros.org/groovy/Planning/Matlab

[55] CHRobotics, 2011, “Understanding Euler Angles”, http://www.chrobotics.com/library/understanding-euler-angles

[56] TurkUAV Quadro XL V2 Frame, 2014, http://www.robomore.com/U198,161,turkuav-quadro-xl-sase-frame-v2-saseler.htm

[57] Multiwii Project, 2011, http://www.multiwii.com

[58] Bifilar Pendulum Experiment, 1999, http://voyager.egglescliffe.org.uk/physics/gravitation/bifilar/bif.html

[59] MicroStrain 3DM-GX3-25, 2010, http://www.microstrain.com/inertial/3DM-GX3-25

[60] MicroStrain Data Monitor Software, 2010, http://www.microstrain.com/documentation/8200-0019-mip-monitor-software-3dm-gx3-3dm-gx4-3dm-rq1

[61] Lynxmotion Company, 1995, http://www.lynxmotion.com

[62] STM32F103CBT6 32-bit ARM Cortex M3 processor, 2013, http://www.st.com/web/en/resource/technical/document/datasheet/CD00161566.pdf

[63] Invensense MPU6050, 2013, http://www.farnell.com/datasheets/1788002.pdf

[64] Honeywell HMC5883L, 2013, http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf

[65] Meas-Spec MS5611, 2012, http://www.amsys.info/sheets/amsys.en.ms5611_01ba03.pdf

[66] Micron M25p16, 2011, http://media.digikey.com/pdf/Data%20Sheets/Numonyx,Intel/M25P16.pdf

[67] ATTOPILOT sensor, 2011, https://www.sparkfun.com/products/10644

[68] Xbee Pro900, 2012, http://tr.mouser.com/new/Digi-International/digixbeeprop/

[69] Graupner MX-16, 2012, http://www.graupner.de/en/products/1d80881f-8353-4c86-8a6d-2ebefc64e1c1/33116/product.aspx

[70] TurkUAV MT2814 750KV, 2013, http://www.robomore.com/U38,121,turkuav-2814-v2-multikopter-motoru-fircasiz-brushless-motor-turkuav.htm

[71] TurkUAV 40A ESC, 2013, http://www.robomore.com/U199,125,turkuav-40a-400hz-esc-simon-k-esc-turkuav.htm

[72] 4S 4000mAh GensAce Li-Po Battery, 2014, http://www.gensace.de/recommend-products/gens-ace-4000mah-14-8v-25c-4s1p-lipo-battery-pack.html

[73] The GPL (GNU) General Public License, 2007, http://www.gnu.org/licenses/gpl-3.0.en.html

[74] Notpad ++, 2015, https://notepad-plus-plus.org/

[75] Cygwin, 2000, https://www.cygwin.com/

[76] Simulink Design Optimization Toolbox, 2013, http://www.mathworks.com/products/sl-design-optimization/

[77] Paralle Axis Theorem, 2000, http://hyperphysics.phy-astr.gsu.edu/hbase/parax.html

[78] SH-1290MG Savöx Servo Motor, 2012, http://www.savoxusa.com/Savox_SH1290MG_Digital_Servo_p/savsh1290mg.htm

APPENDIX

APPENDIX A

```
static void pidLuxFloat(pidProfile_t *pidProfile, controlRateConfig_t *controlRateConfig,
      uint16_t max_angle_inclination, rollAndPitchTrims_t *angleTrim, rxConfig_t *rxConfig)
{
   float RateError, errorAngle, AngleRate, gyroRate;
   float ITerm,PTerm,DTerm;
   int32_t stickPosAil, stickPosEle, mostDeflectedPos;
   static float lastError[3];
   static float delta1[3], delta2[3];
   float delta, deltaSum;
   int axis;
   float horizonLevelStrength = 1;

   if (FLIGHT_MODE(HORIZON_MODE)) {

      // Figure out the raw stick positions
      stickPosAil = getRcStickDeflection(FD_ROLL, rxConfig->midrc);
      stickPosEle = getRcStickDeflection(FD_PITCH, rxConfig->midrc);

      if(ABS(stickPosAil) > ABS(stickPosEle)){
         mostDeflectedPos = ABS(stickPosAil);
      }
      else {
         mostDeflectedPos = ABS(stickPosEle);
      }

      // Progressively turn off the horizon self level strength as the stick is banged over
      horizonLevelStrength = (float)(500 - mostDeflectedPos) / 500;  // 1 at centre stick, 0 = max stick
deflection
      if(pidProfile->H_sensitivity == 0){
         horizonLevelStrength = 0;
      } else {
         horizonLevelStrength = constrainf(((horizonLevelStrength - 1) * (100 / pidProfile-
>H_sensitivity)) + 1, 0, 1);
      }
   }

   // ----------PID controller----------
   for (axis = 0; axis < 3; axis++) {
      // -----Get the desired angle rate depending on flight mode
      uint8_t rate = controlRateConfig->rates[axis];

      if (axis == FD_YAW) {
         // YAW is always gyro-controlled (MAG correction is applied to rcCommand) 100dps to
1100dps max yaw rate
         AngleRate = (float)((rate + 10) * rcCommand[YAW]) / 50.0f;
       } else {
         // calculate error and limit the angle to the max inclination
#ifdef GPS
         errorAngle = (constrain(rcCommand[axis] + GPS_angle[axis], -((int) max_angle_inclination),
               +max_angle_inclination) - inclination.raw[axis] + angleTrim->raw[axis]) / 10.0f; // 16
bits is ok here
#else
         errorAngle = (constrain(rcCommand[axis], -((int) max_angle_inclination),
               +max_angle_inclination) - inclination.raw[axis] + angleTrim->raw[axis]) / 10.0f; // 16
bits is ok here
#endif
```

```
            if (FLIGHT_MODE(ANGLE_MODE)) {
                // it's the ANGLE mode - control is angle based, so control loop is needed
                AngleRate = errorAngle * pidProfile->A_level;
            } else {
                //control is GYRO based (ACRO and HORIZON - direct sticks control is applied to rate
PID
                AngleRate = (float)((rate + 20) * rcCommand[axis]) / 50.0f; // 200dps to 1200dps max
roll/pitch rate
                if (FLIGHT_MODE(HORIZON_MODE)) {
                    // mix up angle error to desired AngleRate to add a little auto-level feel
                    AngleRate += errorAngle * pidProfile->H_level * horizonLevelStrength;
                }
            }
        }

        gyroRate = gyroADC[axis] * gyro.scale; // gyro output scaled to dps

        // --------low-level gyro-based PID. ----------
        // Used in stand-alone mode for ACRO, controlled by higher level regulators in other modes
        // -----calculate scaled error.AngleRates
        // multiplication of rcCommand corresponds to changing the sticks scaling here
        RateError = AngleRate - gyroRate;

        // -----calculate P component
        PTerm = RateError * pidProfile->P_f[axis] * PIDweight[axis] / 100;

        // Pterm low pass
        if (pidProfile->pterm_cut_hz) {
            PTerm = filterApplyPt1(PTerm, &PTermState[axis], pidProfile->pterm_cut_hz, dT);
        }

        // -----calculate I component.
        errorGyroIf[axis] = constrainf(errorGyroIf[axis] + RateError * dT * pidProfile->I_f[axis] * 10, -
250.0f, 250.0f);

        // limit maximum integrator value to prevent WindUp - accumulating extreme values when
system is saturated.
        // I coefficient (I8) moved before integration to make limiting independent from PID settings
        ITerm = errorGyroIf[axis];

        //-----calculate D-term
        delta = RateError - lastError[axis];
        lastError[axis] = RateError;

        // Correct difference by cycle time. Cycle time is jittery (can be different 2 times), so calculated
difference
        // would be scaled by different dt each time. Division by dT fixes that.
        delta *= (1.0f / dT);
        // add moving average here to reduce noise
        deltaSum = delta1[axis] + delta2[axis] + delta;
        delta2[axis] = delta1[axis];
        delta1[axis] = delta;

        // Dterm low pass
        if (pidProfile->dterm_cut_hz) {
            deltaSum = filterApplyPt1(deltaSum, &DTermState[axis], pidProfile->dterm_cut_hz, dT);
        }
```

```
        DTerm = constrainf((deltaSum / 3.0f) * pidProfile->D_f[axis] * PIDweight[axis] / 100, -300.0f,
300.0f);

        // -----calculate total PID output
        axisPID[axis] = constrain(lrintf(PTerm + ITerm + DTerm), -1000, 1000);

#ifdef GTUNE
        if (FLIGHT_MODE(GTUNE_MODE) && ARMING_FLAG(ARMED)) {
            calculate_Gtune(axis);
        }
#endif

#ifdef BLACKBOX
        axisPID_P[axis] = PTerm;
        axisPID_I[axis] = ITerm;
        axisPID_D[axis] = DTerm;
#endif
    }
}
```

APPENDIX B

```
Servo1_Kp = ((int32_t)servoConf[SERVO_GIMBAL_PITCH].rate *
(inclination.values.rollDeciDegrees / 10)); // Pitch Servo -> Servo1

Servo2_Kp = ((-1*((int32_t)servoConf[SERVO_GIMBAL_PITCH].rate)) *
(inclination.values.rollDeciDegrees / 10)); // Roll Servo -> Servo2
gear_gyroRate = gyroADC[ROLL] * gyro.scale; // gyro output scaled to dps ->( 16.4 dps/lsb
scalefactor gyro.scale = 1.0f / 16.4f)

        delta_s=gear_gyroRate;
        deltaSum_s = delta_s1 + delta_s2 + delta_s;
        delta_s2 = delta_s1;
        delta_s1 = delta_s;

deltaSum_s = filterApplyPt1(deltaSum_s, &KDTermState_s,10);

        gear_gyroRate_s=deltaSum_s / 3.0f;

Servo1_Kd = ((int32_t)servoConf[SERVO_GIMBAL_ROLL].rate * gear_gyroRate_s ); // Pitch
Servo -> Servo1

Servo2_Kd = ( (-1*((int32_t)servoConf[SERVO_GIMBAL_ROLL].rate)) * gear_gyroRate_s ); //
Roll Servo -> Servo2

Servo1_PID= Servo1_Kp + Servo1_Kd; // Pitch Servo -> Servo1
Servo2_PID= Servo2_Kp + Servo2_Kd; // Roll Servo -> Servo2

servo[SERVO_GIMBAL_PITCH] = Servo1_PID +
determineServoMiddleOrForwardFromChannel(SERVO_GIMBAL_PITCH);
servo[SERVO_GIMBAL_ROLL] =  Servo2_PID +
determineServoMiddleOrForwardFromChannel(SERVO_GIMBAL_ROLL);
```