ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

School of Electrical and Computer Engineering

Control of Quad Rotor Unmanned Aerial Vehicle (UAV) Using LQG Track controller

A thesis Submitted to the Addis Ababa Institute of Technology, School of Graduate Studies, Addis Ababa University In partial fulfillment of the Requirement for the Degree of

MASTERS OF SCIENCE IN ELECTRICAL ENGINEERING (ELECTRICAL CONTROL ENGINEERING)

By

Bekele Negese

Advisor: Professor Nagendra Prasad Singh

June, 2016

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

School of Electrical and Computer Engineering

Control of Quad Rotor Unmanned Aerial Vehicle (UAV) Using LQG Track controller

BY

BEKELE NEGESE

APPROVAL BY BORD OF EXAMINERS

CHAIRMAN DEPARTMENT OF GRADUATE COMMITTEE	SIGNATURE
Professor Nagendra Prasad Singh ADVISOR	SIGNATURE
INTERNAL EXAMINER	SIGNATURE
EXTERNAL EXAMINER	SIGNATURE

DECLARATION

I, the undersigned, declare that this thesis is my original work, has not been presented for a degree in this or other universities, all sources of materials used for this thesis work have been fully acknowledged.

Name: Bekele Negese

Signature: _____

Place: Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa

This thesis has been submitted for examination with my approval as a university advisor

Professor Nagendra Prasad Singh _____

Advisor's Name Signature

DEDICATION

I would like to dedicate this documentation to my father **Negese Entele Negewo** and my mother **Wude Desta Jima** for their in comparison support.

ACKNOWLEDGEMENT

I would like to thank my thesis advisor, **Nagendra Singh (prof.)** for his insight and support throughout the duration of this thesis. No matter the length of barrier between us, he has covered all my problem through the email.

Thank you my department for giving me the chance to carry out this experience. Many thanks to my family for the patience and the support in all my choices, no matter what they were for that they have spent their labor more than enough on me both financially and ideologically.

Last but not least, I would also like to thank my friends and fellow students for the great time we spent together and the experiences we shared together. Their support is not forgettable.

ABSTRACT

This thesis is focused towards the studies on Vertical Take-Off and landing (VTOL) Unmanned Aerial Vehicle (UAV) quad rotor control. The quad rotor is controlled by four BLDC motors which act in different directions to control the yaw angle, roll angle and pitch angle and z-axis position. The control action basically depends on the controlled voltages fed to the four motors.

The dynamic modelling of quad rotor is discussed and the design of Linear Quadratic Gaussian (LQG) is presented. A simulator based on MATLAB/SIMULINK model of UAV quad rotor is developed to carry out simulation studies. The effectiveness of the proposed LQG control algorithm to control the hovering position and cruising position in the presence of disturbances such as plant noise and sensor noise is investigated through simulation studies on the simulator.

The effectiveness of the proposed controller is tested with and without disturbance through simulation studies. The first test is observed from model verification of open loop response. It is observed from the open loop response that the altitude control is done through three conditions. This was done by controlling the four input controls through rotors frequencies which are compared with the hovering frequencies.

Closed loop simulation studies are carried out with the proposed LQG controller. It is observed that the desired tracking is achieved almost within 9 seconds after giving the step input command signals.

Similarly, it is observed that the hovering position is achieved almost within 8 seconds after giving the step input command signals. It is further observed that the quad rotor tracks the command signals almost after 10 seconds under the influence of disturbance of covariance value 0.9. It is further observed that the quad rotor attains the desired hovering position after 9 seconds under the influence of disturbance of covariance value 0.9.

Key words: VTOL, UAV, LQG, LQR, Kalman Filter, Tracking control, Hovering position Stabilization

It is also observed from simulation results that the three dimensional position of the quadrotor for ramp and sinusoidal commands is not optimum. This is due to fast change of command magnitudes on the ramps and sinusoidal signals compared to step inputs. However, the tracking control and hovering stabilization is achieved with small magnitude of error.

Thus, it is concluded that the LQG controller stabilizes the UAV quad rotor both for hovering position and tracking control with different magnitudes of oscillation and time durations depending on the white noise covariance and types of commanded signal.

Key words: VTOL, UAV, LQG, LQR, Kalman Filter, Tracking control, Hovering position Stabilization

CONTENTS

LIST OF FIGURES

LIST OF TABLES

UNITS OF MEASUREMENTS

Symbol	*Name*	*Quantities*	*Equivalences*
$\boldsymbol{deg}$	Degree	angle	$Rad * 180/pi$
$\boldsymbol{g}$	gram	weight	$kg/1000$
$\boldsymbol{kg}$	Kilogram	weight	kg
$\boldsymbol{m.s^{-1}}$	Meter per second	Speed; velocity	$3.6 * km/hr$
$\boldsymbol{m}$	meter	length	$km.10^{-3}$
$\boldsymbol{m.s^{2}}$	Meter per second square	Linear acceleration	$m.s^{2}$
$\boldsymbol{Rad.s^{-1}}$	Rad per second	Angular speed; angular velocity	$Rad.s^{-1}$
$\boldsymbol{sec}$	second	time	$hr/3600$
$\boldsymbol{J}$	Joule	Work	J
$\boldsymbol{N}$	Newton	Force	N
$\boldsymbol{N.m}$	Newton Metre	Torque	$N.m$
$\boldsymbol{N.m.s^{2}}$		Inertia	$kg.m^{2}$
$\boldsymbol{V}$	Volt	Voltage	$kg.m^{2}.s^{3}.A^{-1}$
$\boldsymbol{w}$	watt	power	$N.m.s^{-1}$
Ω	ohm	Resister	$kg.m^{2}.s^{-3}.A^{-2}$

LIST OF CONSTANTS

symbol	*SI Unit*	*value*		*name*
$\boldsymbol{b}$	$N.s^2$	$54.2 * 10^{-6}$	**[2]**	Thrust Factor
$\boldsymbol{d}$	$N.m.s^2$	$1.1 * 10^{-6}$	**[2]**	Drag Factor
$\boldsymbol{g}$	$kg.m.s^{-2}$	9.81	---	Gravitational acceleration
$\boldsymbol{l}$	m	0.24	[**2**]	length
$\boldsymbol{m}$	kg	0.65	[**2**]	weight
$\boldsymbol{J_{xx}}$	$N.m.s^2$	$8.1 * 10^{-3}$	[**2**]	x-axis inertia
$\boldsymbol{J_{yy}}$	$N.m.s^2$	$8.1 * 10^{-3}$	[**2**]	y-axis inertia
$\boldsymbol{J_{zz}}$	$N.m.s^2$	$1.42 * 10^{-2}$	[**2**]	y-axis inertia
$\boldsymbol{mu}$	$---$	0.75	[**20**]	White noise covariance
$\boldsymbol{J_{Tm}}$	$N.m.s^2$	$3.68 * 10^{-6}$	[**2**]	Motor Inertial constant
$\boldsymbol{k_m}$	$N.m.A^{-1}$	$6.3 * 10^{-3}$	[**2**]	Mechanical motor constant
η	$--$	0.9	[**2**]	Gear box efficiency
$\boldsymbol{R_A}$	Ω	0.6	[**2**]	Motor armature Resistor

LIST OF VARIABLES

Symbol	*SI Unit*	*Description*
$\mathbf{e}$	volt	Back electro-motive force of the motor
$\mathbf{e}$	$+$	Error
$\boldsymbol{p}$	$\frac{Rad}{sec}$	Quad rotor angular velocity around x_B w.r.t. B-frame
$\dot{\boldsymbol{p}}$	$\frac{Rad}{sec^2}$	Quad rotor angular acceleration around x_B w.r.t. B-frame
$\boldsymbol{q}$	$\frac{Rad}{sec}$	Quad rotor angular velocity around y_Bw.r.t. B-frame
$\dot{\boldsymbol{q}}$	$\frac{Rad}{sec^2}$	Quad rotor angular acceleration around y_B w.r.t. B-frame
$\boldsymbol{r}$	$\frac{Rad}{sec}$	Quad rotor angular velocity around z_B w.r.t. B-frame
$\dot{\boldsymbol{r}}$	$\frac{Rad}{sec^2}$	Quad rotor angular acceleration around z_B w.r.t. B-frame
$\boldsymbol{t}$	$seconds$	Time
$\boldsymbol{u}$	$\frac{m}{sec}$	Quad rotor linear velocity around x_B w.r.t. B-frame
$\dot{\boldsymbol{u}}$	$\frac{m}{sec^2}$	Quad rotor linear acceleration around x_Bw.r.t. B-frame
$\boldsymbol{v}$	$\frac{m}{sec}$	Quad rotor linear velocity around y_Bw.r.t. B-frame
$\dot{\boldsymbol{v}}$	$\frac{m}{sec^2}$	Quad rotor linear acceleration around y_B w.r.t. B-frame
$\boldsymbol{V}$	volt	Input voltage to the motor
$\boldsymbol{w}$	$\frac{m}{sec}$	Quad rotor linear velocity around z_Bw.r.t. B-frame
$\dot{\boldsymbol{w}}$	$\frac{m}{sec^2}$	Quad rotor linear acceleration around z_Bw.r.t. B-frame
$\boldsymbol{F_x}$	N	Quad rotor force along x_B w.r.t. B-frame
$\boldsymbol{F_y}$	N	Quad rotor force along z_Bw.r.t. B-frame

Symbol	*SI Unit*	*Description*
$\boldsymbol{F_z}$	N	Quad rotor force along z_B w.r.t. B-frame
$\boldsymbol{M_x}$	N. m	Quad rotor Moment along x_B w.r.t. B-frame
$\boldsymbol{M_y}$	N. m	Quad rotor Moment along y_B w.r.t. B-frame
$\boldsymbol{M_z}$	N. m	Quad rotor Moment along z_B w.r.t. B-frame
$\boldsymbol{\tau x}$	N. m	Quad rotor torque along x_B w.r.t. B-frame
$\boldsymbol{\tau y}$	N. m	Quad rotor torque along y_B w.r.t. B-frame
$\boldsymbol{\tau z}$	N. m	Quad rotor torque along z_B w.r.t. B-frame
$\boldsymbol{R^1_{0,x}}$	-----	rotation matrix around the x- axis
$\boldsymbol{R^1_{0,y}}$	-----	rotation matrix around the y- axis
$\boldsymbol{R^1_{0,z}}$	-----	rotation matrix around the z- axis
$\boldsymbol{T}$	N. m	Trust to the motor
$\boldsymbol{R^b_v}$	----	rotation matrix (roll-pitch-yaw)
$\boldsymbol{U_1}$	N. m	Vertical Trust control w.r.t. body frame
$\boldsymbol{U_2}$	N. m	Pitch Torque control w.r.t. body frame
$\boldsymbol{U_3}$	N. m	Roll torque control w.r.t. body frame
$\boldsymbol{U_4}$	N. m	Yaw torque control w.r.t. body frame
$\boldsymbol{x}$	m	Quad rotor linear position along x-axis w.r.t. E-frame
$\boldsymbol{\dot{x}}$	$\frac{m}{sec}$	Quad rotor linear velocity along x-axis w.r.t. E-frame
$\boldsymbol{\ddot{x}}$	$\frac{m}{sec^2}$	Quad rotor linear acceleration along x-axis w.r.t. E-frame
$\boldsymbol{y}$	m	Quad rotor linear position along y-axis w.r.t. E-frame
$\boldsymbol{\dot{y}}$	$\frac{m}{sec}$	Quad rotor linear velocity along y-axis w.r.t. E-frame
$\boldsymbol{\ddot{y}}$	$\frac{m}{sec^2}$	Quad rotor linear acceleration along y-axis w.r.t. E-frame
$\boldsymbol{z}$	m	Quad rotor linear position along z-axis w.r.t. E-frame
$\boldsymbol{\dot{z}}$	$\frac{m}{sec}$	Quad rotor linear velocity along z-axis w.r.t. E-frame

Symbol	*SI Unit*	*Description*
$\ddot{z}$	$\frac{m}{sec^2}$	Quad rotor linear acceleration along z-axis w.r.t. E-frame
ϑ	Rad	Quad rotor angular position around x-axis w.r.t. E-frame (pitch)
$\dot{\vartheta}$	$\frac{Rad}{sec}$	Quad rotor angular velocity around x-axis w.r.t. E-frame (pitch)
$\ddot{\vartheta}$	$\frac{Rad}{sec^2}$	Quad rotor angular acceleration around x-axis w.r.t. E-frame (pitch)
θ	Rad	Quad rotor angular position around y-axis w.r.t. E-frame (roll)
$\dot{\theta}$	$\frac{Rad}{sec}$	Quad rotor angular velocity around y-axis w.r.t. E-frame (roll)
$\ddot{\theta}$	$\frac{Rad}{sec^2}$	Quad rotor angular acceleration around y-axis w.r.t. E-frame (roll)
φ	Rad	Quad rotor angular position around z-axis w.r.t. E-frame (yaw)
$\dot{\varphi}$	$\frac{Rad}{sec}$	Quad rotor angular velocity around z-axis w.r.t. E-frame (yaw)
$\ddot{\varphi}$	$\frac{Rad}{sec^2}$	Quad rotor angular acceleration around z-axis w.r.t. E-frame (yaw)
Ω_{rr}	$\frac{Rad}{sec}$	Quad rotor back motors speed
Ω_{rg}	Rad/sec	Quad rotor right motors speed
Ω_{l}	$\frac{Rad}{sec}$	Quad rotor left motors speed
Ω_{f}	Rad/sec	Quad rotor front motors speed
$\Omega_{m,\ p}$	$\frac{Rad}{sec}$	rotor speed of quad rotor, propeller
A_{x_m}	$\frac{m}{sec^2}$	Accelerometer x-axis acceleration measurement
A_{y_m}	$\frac{m}{sec^2}$	Accelerometer y-axis acceleration measurement

Symbol	*SI Unit*	*Description*
$\boldsymbol{A_{z_m}}$	$\frac{m}{sec^2}$	Accelerometer z-axis acceleration measurement
$\boldsymbol{\lambda_{p_m}}$	$\frac{Rad}{sec}$	Gyro bias for body angular rate [pitch]
$\boldsymbol{\lambda_{q_m}}$	$\frac{Rad}{sec}$	Gyro bias for body angular rate [roll]
$\boldsymbol{\lambda_{r_m}}$	$\frac{Rad}{sec}$	Gyro bias for body angular rate [yaw]
$\boldsymbol{w_{p_m}}$	$\frac{Rad}{sec}$	White noise due to gyroscope[pitch]
$\boldsymbol{w_{q_m}}$	Rad/sec	White noise due to gyroscope[roll]
$\boldsymbol{w_{r_m}}$	Rad/sec	White noise due to gyroscope[yaw]
$\boldsymbol{\lambda_{A_{x_m}}}$	$\frac{m}{sec^2}$	Bias of accelerometer from x-axis
$\boldsymbol{\lambda_{A_{y_m}}}$	$\frac{m}{sec^2}$	Bias of accelerometer from y-axis
$\boldsymbol{\lambda_{A_{z_m}}}$	$\frac{m}{sec^2}$	Bias of accelerometer from z-axis
$\boldsymbol{w_{A_{x_m}}}$	$\frac{m}{sec^2}$	White noise due to accelerometer on x-axis
$\boldsymbol{w_{A_{y_m}}}$	$\frac{m}{sec^2}$	White noise due to accelerometer on y-axis
$\boldsymbol{w_{A_{z_m}}}$	$\frac{m}{sec^2}$	White noise due to accelerometer on z-axis
$\boldsymbol{\tau_m}$	N. m	Torque due to motor
$\boldsymbol{\tau_l}$	N. m	Torque due to loads
$\boldsymbol{V_{R_A}}$	V	Voltage drops on armature resistor
$\boldsymbol{k_f}$	-----	Controlled Frequency factor for front motor
$\boldsymbol{k_{rg}}$	-----	Controlled Frequency factor for right motor
$\boldsymbol{k_l}$	-----	Controlled Frequency factor for left motor
$\boldsymbol{k_{rr}}$	-----	Controlled Frequency factor for rear motor
$\boldsymbol{V_{L_A}}$	V	Voltage drops on armature inductor
$\boldsymbol{\Gamma}$	-----	Time constant

LIST OF ABBREVIATIONS

3D	Three Dimension
AAIT	Addis Ababa Institute of Technology
ARE	Algebraic Riccati Equation
BEMF	Back Electro Motive Force
BLDC	Brushless Direct Current Motor
CCW	Counter Clock Wise
COM	Centre OF Mass
CW	Clock Wise
DC	Direct Current
D-K	Controller gain– Scaling gain
DOF	Degree Of Freedom
EKF	Extended Kalman Filter
ENSA	Ethiopian National Security Agency
FB	Body Frame
FBL	Feedback Linearization
FI	Inertial Frame
GPS	Geographic position sensor
g_r	Gear Ratio
H	Hub Force
H∞	H-infinity
IB	Integral Back Stepping
IMU	Inertial Measurement Unit
IR	InfraRed
Kare	LQR Gain
Kest	Kalman estimate gain
KF	Kalman Filter
LHR	Left Handed Rotation
LQG	Linear Quadratic Gaussian
LQI	Linear Quadratic Regulator

LQR	Linear Quadratic Regulator
LQT	Linear Quadratic Tracker
MEMS	Micro Electro-Mechanical System
MIMO	Multiple Input Multiple Output
NASG	Nihon Aero Student Group
NED	North East Dawn
PD	Proportional and Derivative
PID	Proportional, Integral and Derivative
PWM	Pulse Width Modulation
Rm	Rolling Moment
Rrad	Radius of radiation
SISO	Single Input Single Output
SMC	Sliding Mode Controller
SONAR	SOund Navigation And Ranging
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take Off and Landing
w.r.t.	With Respect To

CHAPTER ONE

1. INTRODUCTION

1.1. Background

This thesis focuses on the study of a Vertical Take-Off and Landing (VTOL) Unmanned Aerial Vehicle (UAV). This is a vehicle of four propeller UAV called quad rotor. Quad rotors are a special form of rotorcraft UAV that use two pairs of contra rotating rotors to provide lift and directional control **[3].**

Unlike conventional helicopters, quad rotors typically have fixed-pitch blades and vary their thrust by changing rotor speed. Flight attitude is regulated entirely by rotor speed. The mechanical controls used in a convectional vehicle are ***'swash plate'*** and or '***Bell-Hillier stabilizer linkage'***. Swash plates are structure that consists of two ***parallel moving bearings*** fixed on the rotor mast to transmit angular displacement to the pitch horns of rotor blades. Failure of the swash plate causes catastrophic loss of cyclic control and, typically, destruction of the vehicle whereas the problem due to ***Bell-Hillier stabilizer linkage*** slows down the dynamics of the vehicle. The easy and inexpensive maintenance required by quad rotors is key consideration for craft that must operate reliably in proximity to humans, without regular skilled maintenance **[1].**

The compactness of quad rotor helicopters is due to reduced rotor diameters and closely spaced layout. They do not have a single large rotor or long tail boom that can readily collide with nearby obstacles, and instead use small rotors that are easily shrouded for protection. This makes them ideal for tasks indoors or in enclosed spaces, such as inspecting ceilings of a factory, flying down mine shafts or scanning close to civil infrastructure such as bridges or dam walls **[1].**

For the sake of easy construction, several structures and configurations have been developed to allow 3D movements. For example, there are blimps, fixed-wing planes, single rotor helicopters, bird-like prototypes, quad rotors. Each of them has advantages and drawbacks **[2].**

Table 1_ 1: VTOL Concepts Comparison ($\mathbf{1} \xrightarrow{implies} \boldsymbol{Poor/bad}; \mathbf{4} \xrightarrow{implies} \boldsymbol{Excellent}$)

Categories	**A**	**B**	**C**	**D**	***E***	**F**	**G**	**H**
Power Cost	2	2	2	2	*1*	4	3	3
Control Cost	1	1	4	2	*3*	3	2	1
Payload/volume	2	2	4	3	*3*	1	2	1
Maneuverability	4	2	2	3	*3*	1	3	3
Mechanics Simplicity	1	3	3	1	*4*	4	1	1
Aerodynamics Complexity	1	1	1	1	*4*	3	1	1
Low Speed Flight	4	3	4	3	*4*	4	2	2
High Speed	2	4	1	2	*3*	1	3	3
Miniaturization	2	3	4	2	*3*	1	2	4
Survivability	1	3	3	1	*1*	3	2	3
Stationary Flight	4	4	4	4	*4*	3	1	2
Total	**24**	**28**	**32**	**24**	***33***	**28**	**22**	**24**

A=Single Rotor, B=Axial Rotor, C=Coaxial Rotor, D=Tandem Rotors, E=Quad rotor, F=Blimp, G=Bird-like, H=Insect-like **[4].**

Employing four rotors to create differential thrust, the quad rotor is able to hover and move without the complex system of linkages and blade elements present on standard single rotor vehicles. They are classified as an ***under actuated*** system. This is due to the fact that only four actuators (rotors) are used to control all six degrees of freedom (DOF). This means the four actuators directly impact z-axis translation (altitude) and rotation about each of the three principal axes where the other two DOF are translation along the x- and y-axis. These two remaining DOF are coupled, meaning they depend directly on the overall orientation of the vehicle (the other four DOF) **[5]**.

This 6 DOF was developed after extensive researches were done. The strangest application of the quad rotor concept is perhaps the ***Yuri I***, a human powered vehicle designed and built by a team from the Nihon Aero Student Group (***NASG***) lead by Dr. Akira Naito. Even though some features were solved, still there are advantages and dis advantages exist as long as the model was simplified at least as minimum as possible for that the controller to be designed is robust, simple and efficient.

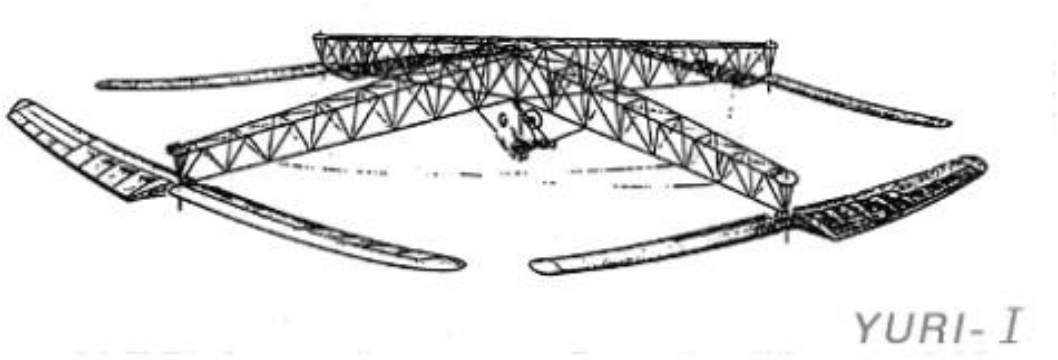

Figure 1_ 1: The Yuri I, a human powered quad rotor **[6].**

A) Major advantages of the Quad rotor

These advantages, compared to the other configurations, are the following **[6]:**

i) Higher payload capacity

The more obvious advantage is this one. The thrust developed by a rotor increases with its diameter. Thus, by increasing the diameter it is possible to increase the thrust and therefore the payload which can be lifted. However, compressibility effect will be introduced. It should be noted that the ***configurations with an odd number of rotors are unadvisable,*** because it is not possible to arrange them in pairs, for a simple counter balance of torques.

ii) Simplicity of the control action

It is possible to control the attitude of the quad rotor just by adjusting separately the ***rpm*** of each rotor. In all the other configurations, the attitude control is achieved by varying the pitch angle of the blade, while the rotational speed of the rotors remains constant. In order to vary the pitch angle, complex mechanical systems are required. These systems are prone to failure, increase the weight and need frequent maintenance. But if the attitude control can be achieved just by modifying the rpm of the rotors, like quad rotor, then there is no need for those systems which saves costs, weight and volume.

iii) Reduced gyroscopic effects

Gyroscopic effects can affect every rotating body, including the rotor of a helicopter. However, depending on the type of rotor, these gyroscopic effects are different. Its angular momentum vector tends to keep the same orientation when the helicopter changes its attitude. In quad rotor, two rotors are turning clockwise and the other two, counter clockwise. If the rpm are identical in the four rotors then the gyroscopic effects are almost cancelled out.

iv) Improved controllability and stability

For the same mass, a quad rotor has larger moments of inertia around its three axes, compared to a single rotor helicopter. Hence, it will have larger time constants than the single rotor helicopter, at least in theory. Larger time constants mean that the pilot has more time to react to divergent modes and to make the necessary corrections or, in other words, that the controllability is better.

B) Major Disadvantages of the Quad rotor Concept

i) Higher weight, lower payload to weight ratio

As for the higher takeoff weight, it is an obvious conclusion of the fact that, instead of one or two main rotors, there are four. Regarding the low payload/takeoff weight ratio, it is not so obvious. On the other hand, the takeoff weight is larger and the payload is also larger, because the thrust available is bigger.

ii) Bigger power consumption

It implies bigger power plants and bigger energy reserves (either batteries or fuel tanks), and this in turn implies higher takeoff weight, which was already high because of the increased number of rotors. In small unmanned quad rotors powered by electric motors, this issue can be very important. The power consumption is indeed very large and this reduces significantly their flight endurance.

iii) Coupling between controllability and motor dynamics

It has already been said that the possibility of controlling the attitude of the quad rotor just by independently modifying the speeds of each rotor was a great advantage, because it rendered unnecessary all the complex mechanical systems needed to change the blade pitch angle. However, it should not be forgotten that the speed of the rotors depends strongly on the dynamics of the motor(s) driving them. Any motor or engine, no matter of what type (electric, internal combustion, gas turbine, steam powered) has a certain inertia to changes in its regime (i.e., speed). The larger the inertia, the larger the time lag.

1.2. Statement of the Problem

Although much progress has been made in the field of UAV quad rotors, it is still a great challenge to build a quad rotor capable of fully autonomous flight. To do so, it is necessary to implement the appropriate control algorithms and to have a suitable set of sensors. As for the design of the control algorithms, in order to be successful in that task, it is essential to have a complete understanding of quad rotor flight dynamics. Failure to adequately model these dynamics may lead to the selection of an unsuitable controller.

The most difficult task to control the UAV quad rotor is that the disturbance exerted on the plant and sensors are much difficult to be controlled depending on the degree of external influences. However, this effects can be minimized by using the controller which is optimum and can reject the disturbance. This controller is Linear Gaussian Regulator (LQG).

In this thesis Linear Gaussian Regulator (LQG) controller, which is a separate study of Linear Quadratic Regulator (LQR) and Kalman filter, is used to study the effect of noise on the UAV quadrotor. The estimator estimates the states due to sensor reading whereas LQR control is used to regulate the states and inputs. Twelve states were feedback to fully control the four control parameter.

1.3.Objectives

General objective:

The general objective of this thesis is to investigate and design a LQG controller for effective control of the UAV quad rotor in the presence of random disturbance using MATLAB/ Simulink model.

Specific objectives:

The specific objectives of these thesis are as follows:

1) To study the linear dynamics of quad rotor and to select a suitable model.
2) To design LQG controller for Trajectory following while cruising and, attitude stabilization at hovering condition of UAV quad rotor.
3) To develop a simulator model of UAV quad rotor.
4) To recommend a suitable and effective control scheme for a UAV quad rotor modelling.

1.4. Methodology

Many thesis writers follow different ways for that they need the model which can suit the controller they need to use more or less perfectly. Since LQG controller, which is a separate study of state estimation and LQR control is used, the model is derived in a manner that the position of quad rotor is observed both at hovering and cruising movements.

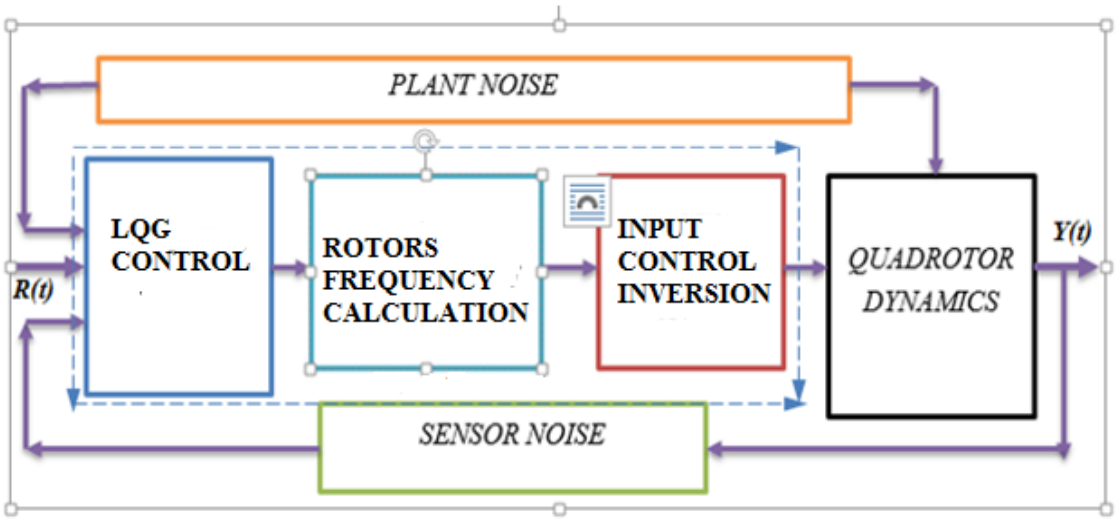

Figure 1_ 2: General structure of the Quad rotor control

1) LQG controller system

In this system, the control design is made. This control has both estimator design and LQR design for which the optimal control is investigated. The estimated parameters from the sensor readings are corrected using the estimator designed in this block.

2) Rotors frequency calculation block

In this block the frequencies of rotors are calculated from the trust that exerted on the rotors of quad rotors.

3) Input inversion block

Because there are no sensors used to measure high frequencies online, they are to be omitted or bounded through the inverse principle so that the control algorithm shall not be overloaded due to high memory compensation.

4) Quad rotor dynamics block

In this block, the state space analysis of the quad rotor model is done on the MATLAB/Simulink level. All states were feedback to the sensors block. However, only seven of them are affected by the sensor reading.

5) Sensor noise and plant noise block

In these blocks, the ***seven states*** readings like SONAR reading (altitude: z), gyroscopic reading (p, q, r) and accelerometer reading (a_x, a_y, a_z) are modelled on the MATLAB/Simulink and observed what is happened under the influence of high speed and high power noise of assumed white noise.

1.5. Scope and Limitation of the Thesis

From the fact that the model of the quad rotor in this thesis is basically from linearized model, it has its own scope and limitations.

Scope of the Thesis:

This thesis worked out only at a MATLAB/Simulink level. To work on this environment, the model is simplified as much as possible by assuming some legible assumptions such as small angle approximation and the advantages of body symmetry. While linearizing, the Tylor first order linearization technique is adopted. The controller used is of type LQG controller which considers both the state estimate and LQR for this linear type. The disturbance is tried to be minimized as much as possible using this controller. The action of path tracking is also observed on this environment.

Limitation of the Thesis:

The limitations of this thesis are as follows.

1) The first limitation is that the thesis is only at a MATLAB/Simulink level. Because of this, identification of any constants is not done. These constants are taken from the other thesis which are shown in the reference they are taken from.
2) The output diverges from the command as long as the angles are not kept small.
3) Due to small angle approximation, the model neglects: the effect of aerodynamics, carioles effect except a few, gyroscopic effect, frictional effect and hub effect and the effect of quaternion is not analyzed.
4) The noise is assumed as a white noise of small noise power.

1.6.Literature Review

In the last few years, the state of the art in Vertical Take-Off and Landing (VTOL) Unmanned Aerial Vehicle (UAV) has received several contributes. Moreover, most of the attention has been focused on, the quad rotor structure.

An iterative steady-state aero elastic simulator was used for holistic blade design. The aerodynamic load disturbances of the system in normal conditions were experimentally characterized to produce performance metrics for system sensitivity. A linear Single Input Single Output ***PID*** controller was designed to regulate a flyer attitude, and was demonstrated on a gimbal test rig. However, this achieved only a satisfactory closed-loop response time **[1]**.

Newton-Euler formalism was used to model the dynamic system. Particular attention was given to the group composed of the DC-motor, the gear box and the propeller which needed also the estimation of aerodynamic lift and torque to reach better accuracy. PID control algorithms were used. The electronics was composed of a Micro Control Unit (MCU) interfaced with several devices. Thanks to these devices and the MCU software, both guided and autonomous flights were possible. However, high approximation of the nonlinear model to the linear model was done for which all external forces are neglected and hence of not full scape **[2]**.

"Nonlinear control using feedback-linearization" control theory was chosen and implemented on the quad rotor to test the results. At the first time, PD controller was used which showed poor performance. To improve the performance, PD controller with partial differential was developed. This still gave a poor performance. At the end, the new controllers was designed using PID control which gave a good result but still not robust **[3]**.

The simulation model of OS4 was developed through several successive steps. The major tasks covered were the inclusion of hub forces (*H*), rolling moments (*Rm*) and variable aero dynamical coefficients. This makes the model realistic particularly in forward flight. With the preliminary versions of the model, it was often necessary to slightly adjust the control parameters for successful experiments with Integral Back stepping (IB) controller. However, estimation problem was there **[4]**.

The platform saw a great improvement after the simplified steady-state Kalman filter was designed and included to merge data from the gyroscopes and accelerometers. This filter was able to account for the long-term drift present in the roll and pitch axes of the gyroscope by pairing the short term accelerometer data, and the noise of the accelerometer was smoothed out by the steady weighted

inclusion of the gyroscope. Quaternion principles to avoid gimbal lock was applied. However, the long term drift of the yaw axis was unaccounted **[5]**.

Angle, height and position controllers were designed and presented using LQR, LQ-servo, pole placement, non-linear input-output feedback and feedback linearization techniques. Firstly, controllers were designed in continuous time, and then tried to give an estimation for such a maximal sampling time that maintains the stability properties of the original closed loop system **[8]**.

The combination of three gyroscopes, three accelerometers and three magnetometers of an IMU with infrared sensors were used to achieve accurate state estimation. This tightly coupled sensor integration allows estimation of all the states of the system and the biases of both the gyroscopes and accelerometers and estimation of all states of the quad rotor. The relationships between the measurements and states were described by non-linear equations, therefore the Extended Kalman Filter (EKF) was used to perform state estimation for the PID controller. Due to the linearization in the EKF the result is not optimal, but results of the filtering was improved by tuning the initial settings of the process and measurement noises **[10]**.

The control of quad rotor was done by using both PID and LQR for the case of both mathematical approach and Engineering approach by feed backing different states at a different time. Trajectory following was observed on the MATLAB/Simulink. However, as intuitively foreseen, stability and robustness mainly depend on the upper limit of the rotor voltages **[11]**.

The mathematical modeling, experimental identification and control design of a small unmanned indoors quad rotor aircraft, at low translational speeds around the hovering condition, where the aerodynamic forces on the airframe are disregarded was done. A Kalman filter was implemented for state estimation and noise filtering. Linear control techniques such as PID, LQ as well as modern robust mixed-sensitivity $H\infty$ and μ-synthesis with *DK*-iteration were employed and compared with each other in terms of flight trajectory reference tracking and parametric and model uncertainty. However, this is a problem when the platform is in the real world **[14]**.

Given the outputs provided by the available sensors, a linearized Kalman filter was developed which is capable of estimating the angular velocities and Euler angles of the Quad rotor, being

therefore possible to have a knowledge of its attitude. This filter was combined with a LQR controller in order to make the system follow a given attitude reference. However, the Quad rotor prototype was not observed in a harsh condition as already all external forces are neglected **[19]**.

1.7.Thesis Lay Out

This thesis work is arranged in five chapters. The first chapter introduces the introductory part of the thesis. The quad rotor back ground like why we select quad rotor from the many aerial vehicle is highlighted. This comparison is made w.r.t. the cost of Power and Control, Payload/volume, Maneuverability, Mechanics Simplicity, Survivability, Stationary Flight and the like. The second chapter revises about the UAV quad rotor model derivation and model simplification. The model of quad rotor includes the translational (forces) and rotational behavior (torque and moments), carioles effect, aerodynamics effect, hub effect and frictional effect only at the equation level. For the translation of one frame to the other frame, the rotational matrix is derived. For the simplification of the equation, first order Tylor series is used to erase the coupling of Euler Angeles and body velocities. The sensors dynamics and the motor dynamics are also discussed and simplified by taking some legible assumptions. The third chapter presents about the selected controller which can make the control robust and disturbance reject in nature. The fourth chapter discusses about the simulation results**.** In the last conclusion and recommendation are given.

CHAPTER TWO

2. MATHEMATICAL MODELLING OF UAV QUAD ROTOR

2.1. Introduction

The UAV quad rotor is very well modeled with a four rotors in a cross configuration. This cross structure is quite thin and light, however it shows robustness by linking mechanically the motors (which are heavier than the structure). Each propeller is connected to the motor through the reduction gears. All the propellers axes of rotation are fixed and parallel. Furthermore, they have fixed-pitch blades and their air flow points downwards (to get an upward lift). These considerations point out that the structure is quite rigid and the only things that can vary are the ***propellers speeds***. This propellers speeds may be affected by ***Aerodynamic forces*** and others. These are:

1) Thrust Force: This force is the resultant of the vertical forces acting on all the blade elements of the propeller
2) Hub Force: The hub force is the resultant of the horizontal forces acting on all the blade elements of the propeller.
3) Drag Moment: This moment about the rotor shaft is caused by the aerodynamic forces acting on the blade elements. It is the horizontal forces acting on the rotor multiplied by the moment arm and integrated over the rotor. It is important as it determines the power required to spin the rotor.
4) Rolling Moment: The rolling moment of a propeller exists in forward flight when the advancing blade is producing more lift than the retreating one. It is the integration over the entire rotor of the lift of each section acting at a given radius where the overall rolling moment is caused by a number of other effects.
5) Ground Effect: Helicopters operating near the ground experience thrust augmentation due to better rotor efficiency. It is related to a reduction of the induced airflow velocity.

The control of UAV quad rotor mainly depend on the Euler angles (roll, pitch and yaw angles) and the positions (x, y and z). The roll, pitch and yaw angles are controlled by ***differential thrust***. Differential thrust between opposite motors provides roll and pitch torques. Differential thrust

between the two pairs of counter-rotating motors provides yaw torque. Position control, with respect to the North-East-Down (NED) coordinate frame is accomplished by controlling the magnitude and direction of the total thrust. A drag force, which will be omitted in this thesis, also acts on the vehicle opposite to the velocity direction **[5], [19]**.

The variation of rotors speed in a different mode changes the orientation of quad rotor as below where '***om***' is the small changes to rotors speed due to control:

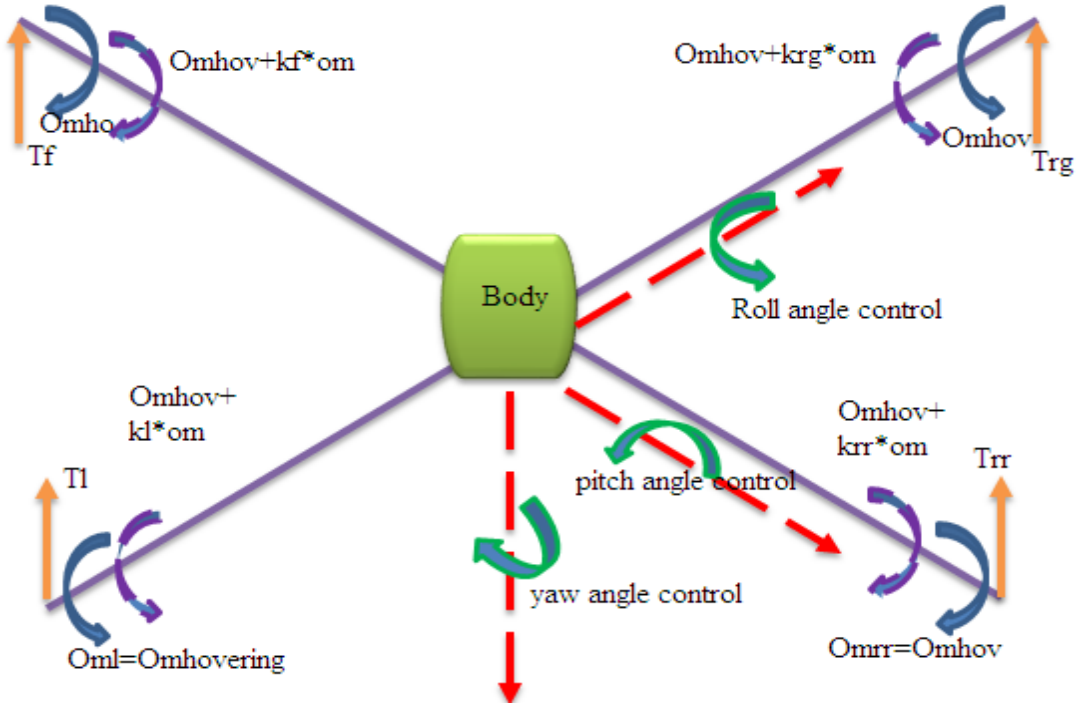

Figure 2_1: Orientation of the Quad rotor

1. *Yawing Right*: If the CW or the CCW spinning actuators are decreased or increased, a net torque will be induced on the craft resulting in a yaw angle. This happens when '***kf***' is taken as small positive value and '***krr***' taken as small negative and the two other constants as zero. Yawing left is analogous.

2. *Hovering / altitude change:* When all actuators are at equal thrust, the craft will either hold in steady hover (assuming no disturbance) or increases/decreases altitude depending on the actual thrust value. This happens when all the four constants are taken as positive (this makes the quad rotor moving up) or small negative (this makes the quad rotor moving down).

3. *Rolling Right*: If one of the actuators is decreased or increased on the roll axis as compared to the other actuator on the same axis, a roll motion will occur. This happens when '***krg***' is taken as small positive and ***'kl'*** taken as small negative and the other two constants as zero. The opposite is for rolling left.

4. *Pitching up:* Similar to the roll axis, if either actuator is changed on the pitch axis, the axis will rotate in the direction of the smaller thrust. This happens when '***kf***' is taken as small positive value and ***'krr'*** is taken as small negative and the other two constants as zero. The opposite is for pitching down.

2.1. Nonlinear Model of UAV Quad rotor

2.1.1. Derivation of Transformational and Rotation matrix for UAV Quad rotor

To control the UAV quad rotor, it is necessary to model it using all possible constrains. One of them is using different Reference frames. It is necessary to use several different coordinate systems for the following reasons:

1) Newton's equations of motion are given in the ***coordinate frame*** attached to the quad rotor.
2) Aerodynamics forces and torques are applied in the ***body frame***.
3) On-board sensors like accelerometers and rate gyros measure information with respect to the ***body frame***. Alternatively, GPS measures position, ground speed, and of course angles with respect to the ***inertial frame***.
4) Most mission requirements like ***flight trajectories*** are specified in the ***inertial frames***.

The expression of these coordinate frames to each other is transformed through two basic operations known as **rotational motion** and **translational motion**. One of them is transformed to the other through a transformation matrix and rotation matrix.

The quad rotor analysis can be done using a ***body reference frame*** i.e. the three angles and their rates and an ***earth inertial fixed frame*** i.e. the three velocity components and their rates. As the state variables, they should be transformed to the same reference frame through the generalized transformed matrix called ***rotational matrix*** if the ***hybrid control structure*** is intended to be used.

Therefore, its importance has no simple boundary which shall not be pushed to the corner without analysis. The following figure shows the transformation on ***z-axis*** for which z-axis is taken as a boundary or it is a point of rotation by an angle ***β***. The objective is to transform the quad rotor from some point to the other. This is a control of yaw angle or yaw movement. The body is assumed to be fixed with respect to the inertial frame.

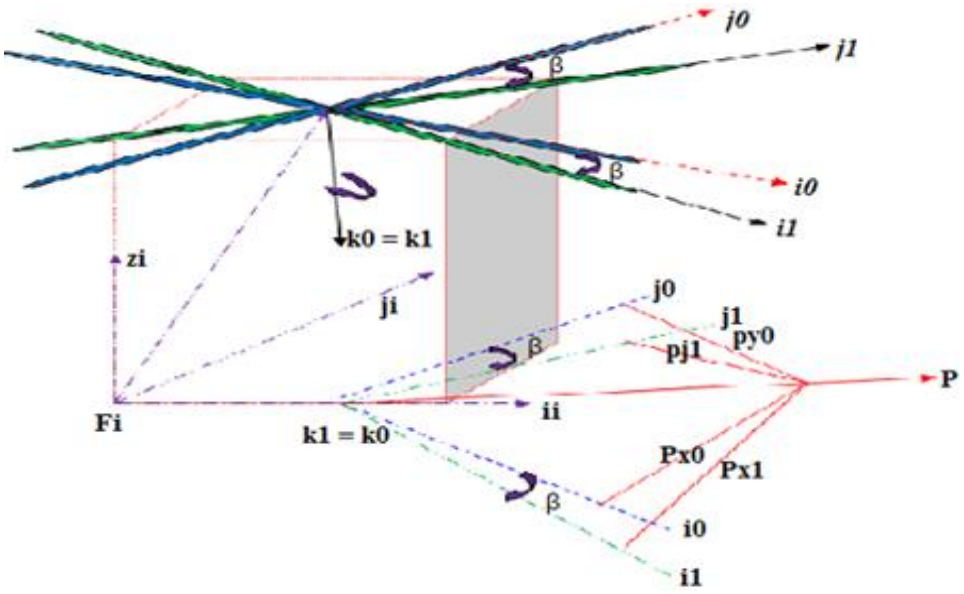

Figure 2_ 1: Transformation of vector p from point $\boldsymbol{p^0}$ to point $\boldsymbol{p^1}$

$$\boldsymbol{P = P_x^0 i^0 + P_y^0 j^0 + P_z^0 k^0} \qquad \textbf{(2.1)}$$

$$P = P_x^1 i^1 + P_y^1 j^0 + P_z^1 k^1 \qquad \textbf{(2.2)}$$

Since **(2.1)** and **(2.2)** are the same, they can be equated and dot production of the two sides by $V^1 = i^1 + j^1 + k^1$ yields the relation of p^1 with respect to p^0.

$$\left(P_x^1 i^1 + P_y^1 j^1 + P_z^1 k^1\right)(i^1 + j^1 + k) = \left(P_x^0 i^0 + P_y^0 j^0 + P_z^0 k^0\right)(i^1 + j^1 + k^1) \qquad \textbf{(2.3)}$$

In matrix form, it can be represented as:

$$\begin{bmatrix} P_x^1 \\ P_y^1 \\ P_z^1 \end{bmatrix} = \begin{bmatrix} i^0\, i^1 & j^0\, i^1 & k^0\, i^1 \\ i^0\, j^1 & j^0\, j^1 & k^0\, j^1 \\ i^0\, k^1 & j^0\, k^1 & k^0\, k^1 \end{bmatrix} \begin{bmatrix} P_x^0 \\ P_y^0 \\ P_z^0 \end{bmatrix} \tag{2.4}$$

This equation shows that the position of **P** can be transformed from the point $\boldsymbol{p_0}$ to $\boldsymbol{p_1}$ through the matrix transformation shown above as a dot product. This transformation matrix is expressed as a rotational matrices from the rotational elements which can be derived as follows. On the following figure, let vector ***p*** is transformed from point **P** to **N** by the projection of vector **n** through angle ∅ and assuming that the angle formed from a rotation of **P** to **Q** with respect to point **N** is **μ**, we can find the expression of vector **q** using vectors **P** and **n** and angle **μ** assuming some facts. The logic is that point **N** is observed from $\mathbf{F^i}$ or **O** through angles ∅ from the point of view of the two vectors.

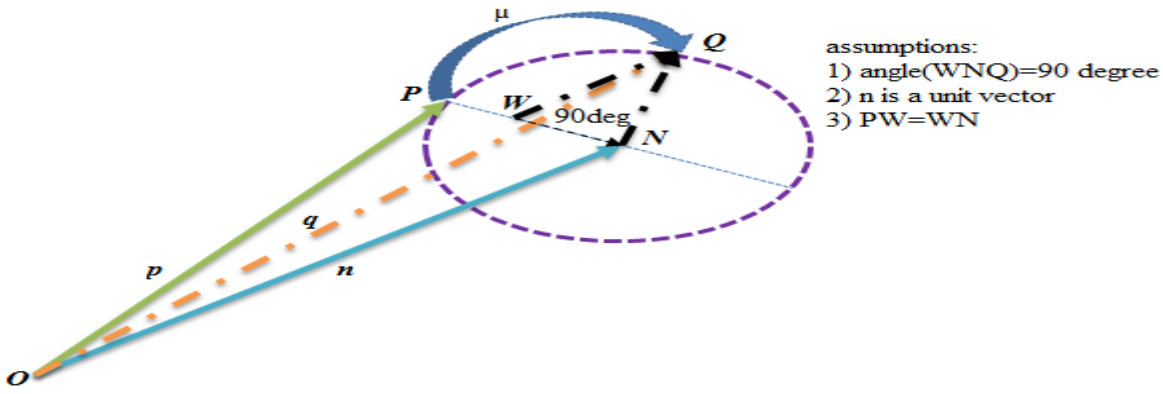

Figure 2_ 2: LHR of a vector p about the unit vector $\widehat{\boldsymbol{n}}$ to obtain the vector q

The rotation matrix **Q** for the above figure is derived as **[7]**:

$$\begin{aligned} ON &= (P.n)n \xrightarrow{and\ also} PN = -NP = OP - ON = P - (P.n)\,n \\ NW &= (|NQ| = |NP|)cos\,\mu \xrightarrow{implies} NW = (P - (P.n)n)cos\,\mu \\ NQ &= -QN = -n\,xP \xrightarrow{implies} WQ = -\,n\,xP\,sin\,\mu \\ q &= ON + NQ = OP + NW + NQ \\ &\xrightarrow{gives!} \\ Q &= (1 - cos\,\mu)(P.n)n + cos\,\mu\,P - n\,x\,P\,sin\mu \end{aligned} \tag{2.5}$$

Using the above derived equation, the three rotational matrices of vehicle frame can be derived easily.

2.1.1.1. Rotation about z-axis (yaw angle,φ) or vehicle frame- one; F v1

The origin of the vehicle-1 frame is the center of gravity. However, **F** ^{v1} is positively rotated about **k** ^{v} by the yaw angle φ so that if the airframe is not rolling or pitching, then **I** ^{v1} would point out the nose of the airframe, **j** ^{v1} points out the right wing, and **k** ^{v1} is aligned with **k** ^{v} and points into the earth.

$$\text{Here: } \gamma = \varphi \,;\, n = k^0 \,;\, P = P_x^0 i^0 + P_y^0 j^0 + P_z^0 k^0 \tag{2.6}$$

$$\mathbf{Q} = (1 - \cos\varphi)\left(\left(P_x^0 i^0 + P_y^0 j^0 + P_z^0 k^0\right).k^0\right)k^0 + \cos\varphi\,(P) - \begin{Bmatrix} i^0 & j^0 & k^0 \\ 0 & 0 & 1 \\ P_x^0 & P_y^0 & P_z^0 \end{Bmatrix} \tag{2.7}$$

For z-axis rotation, the projection of **k** on its self is = 1 where it is = 0 towards others. So, the rotation matrix has a form of:

$$R_{0,z}^1 = \begin{bmatrix} i^0\, i^1 & j^0\, i^1 & 0 \\ i^0\, j^1 & j^0\, j^1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ Which gives } R_{0,z}^1 = \begin{bmatrix} c\varphi & s\varphi & 0 \\ -s\varphi & c\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

2.1.1.2. Rotation about y-axis (pitch angle, θ) or vehicle frame- two; F v2

The origin of the vehicle-2 frame is again the center of gravity and is obtained by rotating the vehicle-1 frame in a right-handed rotation about the **j** ^{v1} axis by the pitch angle θ. If the roll angle is zero, then **i**^{v2} points out the nose of the airframe, **j** ^{v2} points out the right wing, and **k** ^{v2} points out the belly.

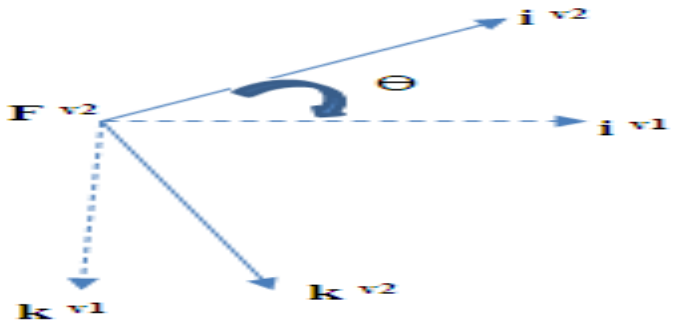

Figure 2_ 3: Vehicle frame- two; **F** ^{v2}

Here: $\gamma = \theta ; n = j^0 ; P = P_x^0 i^0 + P_y^0 j^0 + P_z^0 k^0$ **(2.9)**

For y-axis rotation, the projection of **j** on its self is = 1 where it is = 0 towards others. So, the rotation matrix has a form of, where: $c\theta \; is \cos(\theta), s\theta \; is \sin(\theta)$ and hence, the rotational matrix around the y-axis is given as follows.

$$R_{0,y}^1 = \begin{bmatrix} i^0 i^1 & 0 & k^0 i^1 \\ 0 & 1 & 0 \\ i^0 k^1 & 0 & k^0 k^1 \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \quad \textbf{(2.10)}$$

2.1.1.3. Rotation about x-axis (roll angle,ϑ)

The body frame is obtained by rotating the vehicle-2 frame in a right handed rotation about $\mathbf{i}^{v2}$ by the roll angle ϑ. Therefore, the origin is the center-of-gravity, $\mathbf{i}^{b}$ points out the nose of the airframe, $\mathbf{j}^{b}$ points out the right wing, and $\mathbf{k}^{b}$ points out the belly.

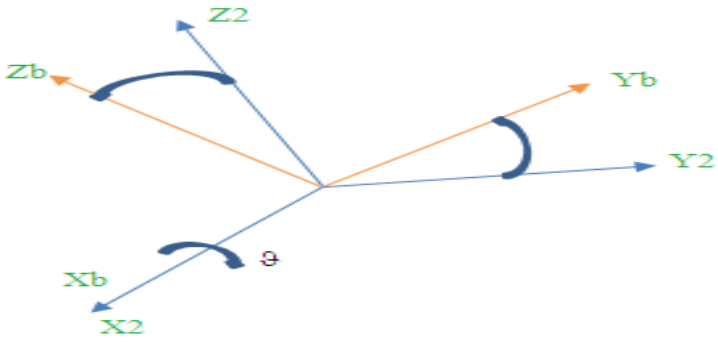

Figure 2_ 4: Rotation about the $\boldsymbol{x_2}$ axis of the angle $\boldsymbol{\vartheta}$ (roll) through R ($\boldsymbol{\vartheta}$, x)= $\boldsymbol{R_x^0}$

Here: $\gamma = \vartheta ; n = i^0 ; P = P_x^0 i^0 + P_y^0 j^0 + P_z^0 k^0$ **(2.11)**

For x-axis rotation, the projection of **I** on its self is = 1 where it is = 0 towards others. So, the rotation matrix has a form of:

$$R_{0,x}^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & j^0 j^1 & k^0 j^1 \\ 0 & j^0 k^1 & k^0 k^1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\vartheta & s\vartheta \\ 0 & -s\vartheta & c\vartheta \end{bmatrix} \quad \textbf{(2.12)}$$

2.1.1.4. Transformation from the vehicle frame to the body frame, $R_v^b(\vartheta, \theta, \varphi)$

Due to many reasons, the quad rotor rotation is needed to be expressed by a ***fixed body frame***. In this case, the origin is the center of gravity and it is the transformation of a rotation matrix which is given as:

$$R_v^b = R_{0,z}^1 * R_{0,y}^1 * R_{0,x}^1 = \begin{bmatrix} c\varphi c\theta & s\varphi c\theta & -s\theta \\ s\vartheta s\theta c\varphi - s\varphi c\vartheta & s\vartheta s\theta s\vartheta + c\varphi c\theta & s\vartheta c\theta \\ c\vartheta s\theta c\varphi + s\varphi s\vartheta & c\vartheta s\theta s\varphi - s\varphi c\vartheta & c\vartheta c\theta \end{bmatrix} \quad \textbf{(2.13)}$$

2.1.2. Dynamics and kinematics of Quad rotor

The analysis of dynamics of the quad rotor always needs a simple way that minimizes the complexity of the calculation with the best minimum error. The carioles' effect is the effect that translates the ***body fixed frame*** with respect to the ***earth inertial fixed frame*** and of course sometimes neglected from the controller design by a means of small angle approximation techniques for simplicity.

2.1.2.1. Equation of Carioles'

The carioles' effect is traditionally derived by a ***coordinate transformation***. Suppose that we are given two coordinate frames F^i and F^b where, F^i might represent the inertial frame and F^b might represent the body frame of a quad rotor. Suppose that the vector p is moving in F^b and that F^b is rotating and translating with respect to F^i . Our objective is to find the time derivative of P as seen from frame F^i.

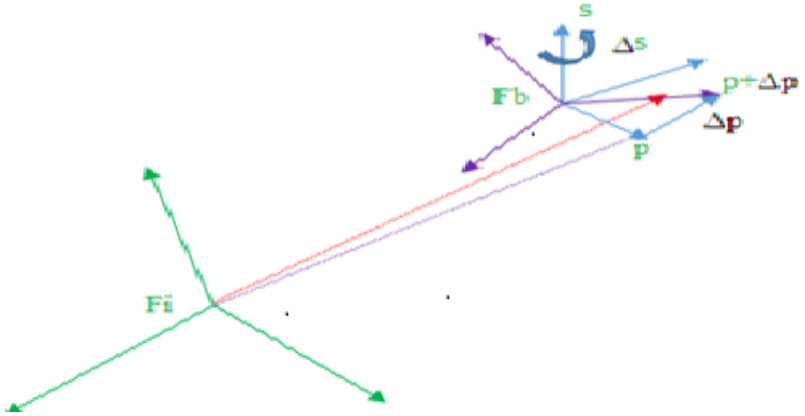

Figure 2_ 5: The projection of vector P on the rotating vector S by angle Δs.

From the above figure, we can drive the following formula depending on the following two assumptions:

1) F^b is not rotating with respect to F^i i.e. their time derivative of P are the same.
2) P is fixed in F^b but rotating with respect to F^i.

$$\frac{\partial P}{\partial t_I} = \frac{\partial P}{\partial t_b} + \delta P$$

$$\left(P = \frac{\partial P}{\partial t_b}\right) + \delta P = \left(1 - \cos(-\mathrm{ds})\right)(\mathrm{S.P})\mathrm{S} + \cos(-\mathrm{ds})\mathrm{P} - \sin(-\mathrm{ds})\mathrm{SxP}$$

$$lim_{ds\to 0}\frac{\sin(-\mathrm{ds})}{\mathrm{dt}} = -\frac{\mathrm{ds}}{\mathrm{dt}}\text{, we get } \frac{\partial P}{\partial t_I} \cong \frac{\mathrm{ds}}{\mathrm{dt}}\ \mathrm{S\ x\ P} \cong \Omega_{\mathrm{b/I}}\ \mathrm{x\ P}\text{;} \qquad \text{we find:} \tag{2.14}$$

$$\frac{\partial P}{\partial t_I} = \frac{\mathrm{dP}}{\mathrm{dt_b}} + \Omega_{\mathrm{b/I}}\ \mathrm{x\ P}$$

Equation **(2.14)** is an equation of Carioles' which is used to calculate forces and moments of the quad rotor here after. The quad rotor model is described in a form of state variables using the following states below:

1) Attitude ($\vartheta, \theta, \varphi$)'
2) Altitude (h) and
3) Position (x, y)

2.1.2.2. State Variables

The model of any system can be described through transfer function model, state space model or any other means. The model of a modern plant is described through the closed loop system. However, there is the system which is classical in nature. The classical (conventional) control theory concerned with single input and single output (SISO) is mainly based on Laplace Transforms theory. The modern control theory concerned with multiple inputs and multiple outputs (MIMO) is based on ***state variable representation*** in terms of a set of first order differential (or difference) equations. Here, the system (plant) is characterized by state variables, say, in linear, time invariant form as **[18]**:

$$\begin{cases} \dot{X}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{2.15}$$

Where: dot denotes differentiation with respect to (WRT) **t**, x(t), u(t), and y(t) are **n**, **r**, and **m** dimensional ***state***, ***control***, and ***output*** vectors respectively, and **A** is **nxn** state, **B** is **nxr** input, **C** is **mxn** output, and **D** is **mxr** transfer matrices. Similarly, a nonlinear system is characterized by:

$$\dot{X}(t) = f(x(t), u(t), t)$$
$$y(t) = g(x(t), u(t), t) \qquad \textbf{(2.16)}$$

The modern theory dictates that all the state variables should be fed back after suitable ***weighting***. We see from fig. 2_6 that in modern control configuration:

1) the input u(t) is determined by the controller (consisting of error detector and compensator) driven by system states x(t) and reference signal r (t) ,
2) all or most of the state variables are available for control, and
3) It depends on well-established matrix theory, which is amenable for large scale computer simulation.

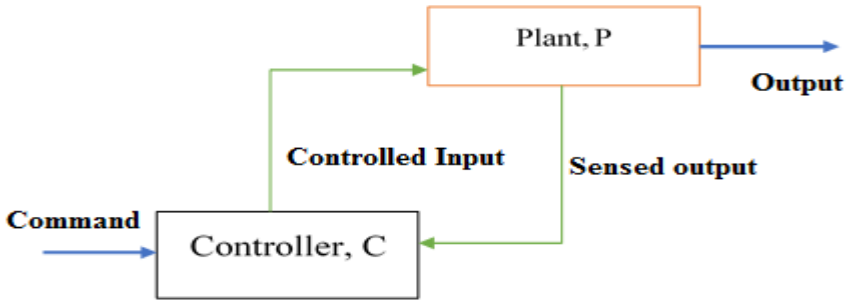

Figure 2_ 6: Modern Control Configuration

The fact that the state variable representation uniquely specifies the transfer function while there are a number of state variable representations for a given transfer function, reveals the state variable representation is a more complete description of a system. For this thesis, since the model is for MIMO systems, the modern control approach is used forth. The quad rotor is controlled from two Reference frames if the state equation has twelve (12) state variables **[5]**. These are:

1) Inertial (earth) frame and
2) Body frame

The state variable of UAV quad rotor is given by:

$$X = \left[P_e \;\; P_n \;\; P_{(h)} \;\; u_x \;\; u_y \;\; u_z \;\; \vartheta \;\; \theta \;\; \varphi \;\; p \;\; q \;\; r \right]^T \quad \textbf{(2.17)}$$

Where:

1) P_n *or y* is the inertial (north) position of the quad rotor along j^I in F^I
2) P_e *or x* is the inertial (east) position of the quad rotor along i^I in F^I
3) *h or z* is the altitude of the aircraft measured along k^I in F^I
4) $u_x = u$ is the body frame velocity measured along i^b in F^b
5) $u_y = v$ is the body frame velocity measured along j^bin F^b
6) $u_z = \omega$ is the body frame velocity measured along k^b in F^b
7) ϑ is the roll angle defined with respect to F^{v2}
8) θ is the pitch angle defined with respect to F^{v1}
9) φ is the yaw angle defined with respect to F^v
10) p is the roll rate measured along i^b in F^b
11) q is the pitch rate measured along j^b in F^b
12) r is the yaw rate measured along k^b in F^b

The objective of the control is analyzing the velocity, acceleration, force and torque of the quad rotor. The time derivative of equation **(2.17)** gives equation **(2.18)** which is the desired equation of state as below where '***T***' is the symbol of transpose.

$$\dot{X} = \left[\dot{P}_e \;\; \dot{P}_n \;\; \dot{P}_{(h)} \;\; \dot{u}_x \;\; \dot{u}_y \;\; \dot{u}_z \;\; \dot{\vartheta} \;\; \dot{\theta} \;\; \dot{\varphi} \;\; \dot{p} \;\; \dot{q} \;\; \dot{r} \right]^T \quad \textbf{(2.18)}$$

2.1.2.3. Kinematics of Quad rotors

This part of writing has a mathematical model of the state variables under:

1) Rate of inertial frame position with respect to body frame velocity
2) Rate of vehicle frame angles ***w***ith ***r***espect ***t***o angular acceleration in body frame
3) Force and Moments under the influence of Aerodynamic effects with respect to a body reference frame

The equations of motion are more conveniently formulated in the ***body-fixed frame*** because of the following reasons **[2]**:

1) The inertia matrix is time-invariant
2) Advantage of body symmetry can be taken to simplify the equations
3) Measurements taken on-board are easily converted to body-fixed frame
4) Control forces are almost always given in body-fixed frame

2.1.2.4. Rate of inertial frame position with respect to body frame velocity

The inertial frame position is transformed to the body frame velocity through a rotational matrix about a body frame as below using Euler kinematics formula where: $c\theta$ is a short hand writing of cosine (θ), $s\vartheta$ is a short hand writing of sine (ϑ) and $t\varphi$ is a short hand writing of $\tan(\varphi)$

$$\frac{\partial}{\partial t}\begin{bmatrix} P_n = X \\ P_y = Y \\ P_{(-h)} = -Z \end{bmatrix} = (R_v^b)^T \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} = \begin{bmatrix} c\theta c\varphi & s\vartheta s\theta s\varphi - c\vartheta s\varphi & c\vartheta s\theta c\varphi + s\vartheta s\varphi \\ c\theta s\varphi & s\theta s\vartheta s\varphi + c\vartheta c\varphi & c\vartheta s\theta s\varphi - s\vartheta c\varphi \\ -s\theta & s\vartheta c\theta & c\vartheta c\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} \quad \textbf{(2.19)}$$

2.1.2.5. Rate of vehicle frame angles w.r.t. angular acceleration in body frame

It is interesting even though not sufficient to assume the following. This assumption significantly reduces the complexity of the equations.

$$assuming \begin{bmatrix} \dot{\vartheta} & \dot{\theta} & \dot{\varphi} \end{bmatrix}^T are\ small, we\ find \quad R_{v2}^b(\dot{\vartheta}) = R_{v1}^{v2}(\dot{\theta}) = R_{v2}^{v1}(\dot{\varphi}) = I \quad \textbf{(2.20)}$$

Using the property,

$$\begin{gathered} (R_a^b)^{-1} = (R_a^b)^T = R_a^b \\ detR_a^b = 1 \\ R_b^c R_a^b = R_a^c \end{gathered} \quad \textbf{(2.21)}$$

We find equation **(2.22)** which is a transformation from body Euler angels rate to earth frame Euler angels rates. These properties are taken from the derivations above derived for rotational matrix under small angle approximation.

$$p = R_{v2}^{b}(\dot{\vartheta})\dot{\vartheta}$$
$$q = R_{v2}^{b}(\vartheta)R_{v1}^{v2}(\dot{\theta})\dot{\theta} \qquad \text{Which gives } \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\vartheta & s\vartheta c\theta \\ 0 & -s\vartheta & c\vartheta c\theta \end{bmatrix} \begin{bmatrix} \dot{\vartheta} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix}$$
$$r = R_{v2}^{b}(\vartheta)R_{v1}^{v2}(\theta)R_{v1\to v2}\,(\dot{\varphi})\dot{\varphi}$$

$$in\ inverse\ form, it\ gives: \qquad \textbf{(2.22)}$$

$$\begin{bmatrix} \dot{\vartheta} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 1 & s\vartheta t\theta & c\vartheta t\theta \\ 0 & c\vartheta & -s\vartheta \\ 0 & s\vartheta sec\theta & c\vartheta sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \xrightarrow[small\ angle\ approximation]{} = \begin{bmatrix} 1 & 0 & \Delta\theta \\ 0 & 1 & -\Delta\vartheta \\ 0 & \Delta\vartheta & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

2.1.2.6. Forces and Moments

There are a number forces applied on the propeller such as forces due to gyroscopic effect, forces due to aero dynamics, inertia due to actuation and external disturbances. The gyroscopic force is used to produce the gyroscopic torques and up lift forces so that the balanced torque is maintained. The blade faces the frictional force and air pressure which add even complexity on the design. The motion of the quad rotor is under the influence of both translational and rotational effects. However, gyroscopic torque is affected only by the rotational effect. There are several techniques which can be used to derive the equations of a rigid body with 6 DOF. The ***Newton-Euler formulation*** has been adopted in this work. The decision to describe the equations of motion in the body-fixed frame trades off complexity in the acceleration terms for relative simplicity in the force terms.

Two assumptions has been done in this approach which states that '***the origin of the body-fixed frame is coincident with the center of mass (COM) of the body'***. Otherwise, another point (COM) should have been taken into account and it would have considerably complicated the body equations. The second one specifies that '***the axes of the B-frame coincide with the body principal axes of inertia'***. In this case the inertia matrix I is diagonal and, once again, the body equations become easier.

2.1.2.6.1. Gravitational Force

The gravitational force which is assumed as constant is taken as if exerted on the center of gravity. The gravitational force is given as below where $F_{g,E}$ is the gravitational force exerted doe to earth and $F_{g,B}$ is the gravitational force exerted due to body of quad rotor.

$$\mathrm{F_{g,E}} = \begin{bmatrix} \mathrm{F_{g,Ex}} \\ \mathrm{F_{g,Ey}} \\ \mathrm{F_{g,Ez}} \end{bmatrix} = -m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \xrightarrow{implies} \mathrm{F_{g,B}} = \begin{bmatrix} \mathrm{F_{g,Bx}} \\ \mathrm{F_{g,By}} \\ \mathrm{F_{g,Bz}} \end{bmatrix} = -m(R_v^b)^T \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad \textbf{(2.23)}$$

2.1.2.6.2. Force generated by propeller system

The total force generated by the four propeller system is given as the first control input U_1 which affects only the trust. It is the force exerted doe to body of the propeller.

$$\mathrm{F_{g,p}} = \begin{bmatrix} \mathrm{F_{g,px}} \\ \mathrm{F_{g,py}} \\ \mathrm{F_{g,pz}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} \quad \textbf{(2.24)}$$

2.1.2.6.3. Aerodynamic drag forces

The effect of Aerodynamic drag force is considered for translational dynamics only. The basic knowledge is that it is proportional to the square of translational velocity and **d** is the aerodynamics drag force constant.

$$F_{aero,B} = \begin{bmatrix} d & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & d \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \textbf{(2.25)}$$

Combining all the above, the nonlinear translational model is given as below. This equation is derived using the carioles principle as a base. While combining, also the effect of frictional force from different parts of the UAV quad rotor is added.

$$F_{net} = m\left(dv/dt + \Omega_{b/i}\ x\ v\right) \quad implies \dots\dots$$

$$mdv/dt = \begin{bmatrix} F_{xnet} \\ F_{ynet} \\ F_{znet} \end{bmatrix} - m\begin{Bmatrix} i & j & k \\ p & q & r \\ u & v & \omega \end{Bmatrix}.(n = i + j + k)$$

$$\left(\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{\omega} \end{bmatrix}; F^B\right) = \left(\frac{1}{m}\begin{bmatrix} F_{xnet} \\ F_{ynet} \\ F_{znet} \end{bmatrix}; F^I\right) + \left(\begin{bmatrix} -q\omega + vr \\ -ur + p\omega \\ -pv + uq \end{bmatrix}; F^B\right) \quad \textbf{(2.26)}$$

$$where:$$

$$R_b^v\left(\begin{bmatrix} F_{xnet} \\ F_{ynet} \\ F_{znet} \end{bmatrix}; F^I\right) = \begin{bmatrix} F_{x,fric+hub,grav} = -\sum_{i=1}^{4} H_{xi} - 0.5c_x A_c \rho \dot{x}\,|\dot{x}| + mgs\theta \\ F_{y,hub+fric,gravitati} = -\sum_{i=1}^{4} H_{yi} - 0.5c_x A_c \rho \dot{y}\,|\dot{y}| - mgc\theta s\vartheta \\ F_z = -mgc\theta c\vartheta \end{bmatrix} + F_{aero,B}$$

The quad rotor dynamic system in equation **(2.26)** is written in the body fixed and Earth fixed frame. As stated before, this reference is rarely used in 6 DOF rigid body equations. However in this case it can be useful to express the dynamics with respect to a ***hybrid*** system composed of linear ***equations WRT E-frame*** and ***angular equations WTR B-frame***. The thrust from each rotor exists only in the ***z-direction*** with respect to the body frame **[21].**

2.1.2.6.4. Moment generated by propeller system

The moment generated by propellers depends on the propeller arm 'l' and control inputs given as below.

$$M_{prpo} = \begin{bmatrix} l & 0 & 0 \\ 0 & l & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad \textbf{(2.27)}$$

Due to Aerodynamics, the quad rotor faces ***the body and propeller gyro effects***, components of ***hub forces***, and ***forces due to friction*** which disturbs the vehicle from not to truck in the correct orientation. The hub forces are exerted on the hub of the vehicle where aerodynamics forces are exerted on the ***air frame part*** that covers the vehicle body for protection purpose. The ***gyroscopic*** effect is the effect due to the gyro sensor reading. These is due to the noise introduced in the sensor that protects the angular velocities of the vehicle no to exactly follow the desired value without noise introduced. Combining all of the above, the overall rotational moment is given as below.

$$J\,{d\Omega^I}/{dt^I} = \tau - l * \omega_{b/i}\ x\ L$$

$$\begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} \begin{bmatrix} \alpha_x = \dot{p} \\ \alpha_y = \dot{q} \\ \alpha_z = \dot{r} \end{bmatrix} = \begin{bmatrix} \tau_{x,net} \\ \tau_{y,net} \\ \tau_{z,net} \end{bmatrix} - l \begin{bmatrix} qL_z - rL_y \\ rL_x - pL_z \\ pL_y - qL_x \end{bmatrix}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} {\tau_\vartheta}/{J_{xx}} \\ {\tau_\theta}/{J_{yy}} \\ {\tau_\varphi}/{J_{zz}} \end{bmatrix} + \begin{bmatrix} \frac{-J_r q \Omega_r + (J_{yy} - J_{zz})qr}{J_{xx}} \\ \frac{J_r p \Omega_r + (J_{zz} - J_{xx})pr}{J_{yy}} \\ \frac{(J_{xx} - J_{yy})pq}{J_{zz}} \end{bmatrix} + \begin{bmatrix} \frac{\tau_{x,aerody:fric\ and\ hub}}{Jxx} \\ \frac{\tau_{y,aerody:fric\ and\ hub}}{Jyy} \\ \frac{\tau_{z,aerody:fric\ and\ hub}}{Jzz} \end{bmatrix}$$

$J_r \ldots\ldots\ldots\ldots\ldots\ldots rotor\ inertia \qquad L \ldots\ldots\ldots\ldots\ldots\ldots angular\ momentum$

$\tau_{x,hub} \ldots\ldots\ldots\ldots - l\left(\sum_{i=1}^{4} H_{yi}\right); \qquad \tau_{x,fric} \ldots\ldots\ldots\ldots \left(\sum_{i=1}^{4} (-1)^{i+1} R_{mxi}\right)$

$\tau_{y,hub} \ldots\ldots\ldots\ldots l\left(\sum_{i=1}^{4} H_{xi}\right); \qquad \tau_{y,fric} \ldots\ldots\ldots\ldots \left(\sum_{i=1}^{4} (-1)^{i+1} R_{myi}\right)$

$\tau_{z,hub} \ldots\ldots\ldots\ldots l\left(H_{xl} - H_{xrg}\right) + l\left(H_{yrr} - H_{yf}\right); \qquad \tau_{z,fric} \ldots\ldots\ldots\ldots \left(\sum_{i=1}^{4} (-1)^{i} Q_i\right)$

(2.28)

Equation **(2.28)** shows the model of the angular acceleration components along the body frame.

Table 2_ 1: Symbols and their corresponding name

Name	***symbol***	***Name***	***Symbol***
Inflow ratio	λ	Lift slope	a
Induced velocity	v	Rotor advance ratio	μ
Twist pitch	θ_{tw}	Rotor radius	R
Average drag coefficient	c_d	Rotor speed	Ω
Air density	ρ	Rotor area	A
Solidity ratio	σ	Pitch of incidence	θ_0

The symbols above are used in the formula representing thrust force, hub force, drag moments, ground effects and torques and forces along the three components.

The general formula that is used in equations **(2.26)** and **(2.28)** are given below **[4].**

a) The thrust force without ground effect is given by

$$T_i = \left(\sigma a \left(\frac{1}{6} + \frac{1}{4}\mu^2\right)\theta_0 - (1 + \mu^2)\frac{\theta_{tw}}{8} - \frac{1}{4}\lambda\right)\rho A(\Omega_i R) \qquad \textbf{(2.29)}$$

b) Hub force is given by

$$H_i = \left(\sigma a\left(\frac{1}{4a}\mu c_d + \frac{1}{4}\lambda\mu\left(\theta_0 - \frac{\theta_{tw}}{2}\right)\right)\right)\rho A(\Omega_i R_{rad})^2 \quad \textbf{(2.30)}$$

The hub force is the resultant of the horizontal forces acting on all the blade elements.

c) $$Q_i = \left(\sigma a\left(\lambda\left(\frac{1}{6}\theta_0 - \frac{\theta_{tw}}{8} - \frac{1}{4}\lambda\right) + \frac{1}{4a}(1+\mu^2)c_d\right)\right)\rho A(\Omega_i R_{rad})^2 R_{rad} \quad \textbf{(2.31)}$$

This moment about the rotor shaft is caused by the aerodynamic forces acting on the blade elements. The horizontal forces acting on the rotor multiplied by the moment arm and integrated over the rotor. Drag moment, $\boldsymbol{Q_i}$, is important as it determines the power required to spin the rotor.

d) Ground effect = $$T_{IGE} = \left(\sigma a\left(\frac{C_T^{OGE}}{\sigma a} + \frac{v_i}{(4\Omega_i R_{rad}(\frac{4z}{R_{rad}})^2)}\right)\right)\rho A(\Omega_i R_{rad})^2 \quad \textbf{(2.32)}$$

e) Rolling moment= $$R_{mi} = \left(-\mu\sigma a\left(-\frac{1}{8}\theta_{tw} + \frac{1}{6}\theta_0 - \frac{1}{8}\lambda\right)\right)\rho A(\Omega_i R_{rad})^2 R_{rad} \quad \textbf{(2.33)}$$

The rolling moment of a propeller exists in forward flight when the advancing blade is producing more lift than the retreating one. It is the integration over the entire rotor of the lift of each section acting at a given radius. This is not to be confused with the overall rolling moment which is caused by a number of other effects.

2.1.2.7. Simplification of the equations by taking legible assumptions

To minimize the mathematical complexity in the controller design, it is better to assume as if aerodynamics effect is Negligible, disturbance is negligible and hub forces are also neglected and the quad rotor is assumed symmetric around its axes **[8]:**

$$\begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} = \begin{bmatrix} \iiint(y^2+z^2)dm & 0 & 0 \\ 0 & \iiint(x^2+z^2)dm & 0 \\ 0 & 0 & \iiint(x^2+y^2)dm \end{bmatrix} \; symmetric$$

(2.34)

The model can be simplified to the following and expressed in state space form of hybrid system with 6 body frame equations and 6 inertial frame equations: the ***three translational velocities*** and the ***three rotational velocities*** respectively**.**

$$\begin{bmatrix} 12\ states: \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\vartheta} \\ \dot{\theta} \\ \dot{\varphi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} twelve\ states: \\ \dot{x}(u,v,\omega) \\ \dot{y}(u,v,\omega) \\ \dot{z}(u,v,\omega) \\ vr - wq + gs\theta \\ wp - ur - gc\theta s\vartheta \\ uq - vp - gc\theta c\vartheta + \frac{U_1}{m} \\ p + qs\vartheta t\theta + rc\vartheta t\theta \\ qc\vartheta - rs\vartheta \\ qs\vartheta sec\theta + rc\vartheta sec\theta \\ \tau_\vartheta / J_{xx} + \frac{(J_{yy}-J_{zz})qr - J_r q\Omega_r}{J_{xx}} \\ \tau_\theta / J_{yy} + \frac{J_r p\Omega_r + (J_{zz}-J_{xx})pr}{J_{yy}} \\ \tau_\varphi / J_{zz} + \frac{(J_{xx}-J_{yy})pq}{J_{zz}} \end{bmatrix} \boldsymbol{where} \begin{bmatrix} \tau_\vartheta \\ \tau_\theta \\ \tau_\varphi \\ F_x \\ F_y \\ F_z \\ T_t \\ J_r \\ T_i \\ \tau_i \end{bmatrix} = \begin{bmatrix} l(T_{rg} - T_f) = U_2 = J_{xx}\ddot{\vartheta} \\ l(T_l - T_{rr}) = U_3 = J_{yy}\ddot{\theta} \\ J_r\ \dot{\Omega}_r + d\left(\sum_{i=l}^{rg}(-1)^i {\Omega^2}_i\right) = U_4 = J_{zz}\ddot{\varphi} \\ gms\theta \\ -gmc\theta s\vartheta \\ -gmc\theta c\vartheta + U_1 \\ U_1 = b({\Omega_{rr}}^2 + {\Omega_f}^2 + {\Omega_{rg}}^2 + {\Omega_l}^2) \\ mR^2 \\ b{\Omega^2}_i \\ d{\Omega^2}_i \end{bmatrix}$$

$$and \quad \boldsymbol{Y} = c\boldsymbol{X}; \qquad \boldsymbol{where}: U_1 \ldots\ldots Vertical\ thrust; U_2 \ldots\ldots Rolling\ moment$$

(2.35)

Equation **(2.35)** is used to model the controller for the three control parameters. In this equation, the control parameters are the four U's from which we can control the ***three*** translational and rotational body velocities and the ***three*** translational and rotational inertial frame accelerations. This is 6 DOF since 6 variables can be controlled. This control parameters are used to control the four rotors speed which in turn used to control the voltage feed in to the motor. Equation ***(2.36)*** describes how the four rotors speeds are obtained from the four control parameters.

The system described by Equation ***(2.36)*** is an under-actuated system, because the number of inputs are four, but the degree of freedom is six (three angles and three coordinates) and the positions depend on the angles, but not on each other where the angles only depend from each other but not from the positions. The Simulink schematic is shown as below for both ***Simulink schematic representation for rotor speed calculation from the four input parameters*** and ***Simulink schematic representation for input control parameters calculation*.**

$$U_3 \ldots\ldots Pitching\ moment;\ \ U_4 \ldots\ldots Yawing\ moment$$

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -l*b & 0 & l*b \\ l*b & 0 & -l*b & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} \Omega_l^2 \\ \Omega_{rg}^2 \\ \Omega_{rr}^2 \\ \Omega_f^2 \end{bmatrix}$$

$for\ the\ motor\ control\ porpuse, we\ use\ an\ inverse\ input\ mehod\ as:$ **(2.36)**

$$\begin{bmatrix} \Omega_l^2 \\ \Omega_{rg}^2 \\ \Omega_{rr}^2 \\ \Omega_f^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4*b} & 0 & \frac{1}{2*l*b} & -\frac{1}{4*d} \\ \frac{1}{4*b} & -\frac{1}{2*l*b} & 0 & \frac{1}{4*d} \\ \frac{1}{4*b} & 0 & -\frac{1}{2*l*b} & -\frac{1}{4*d} \\ \frac{1}{4*b} & \frac{1}{2*l*b} & 0 & \frac{1}{4*d} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}$$

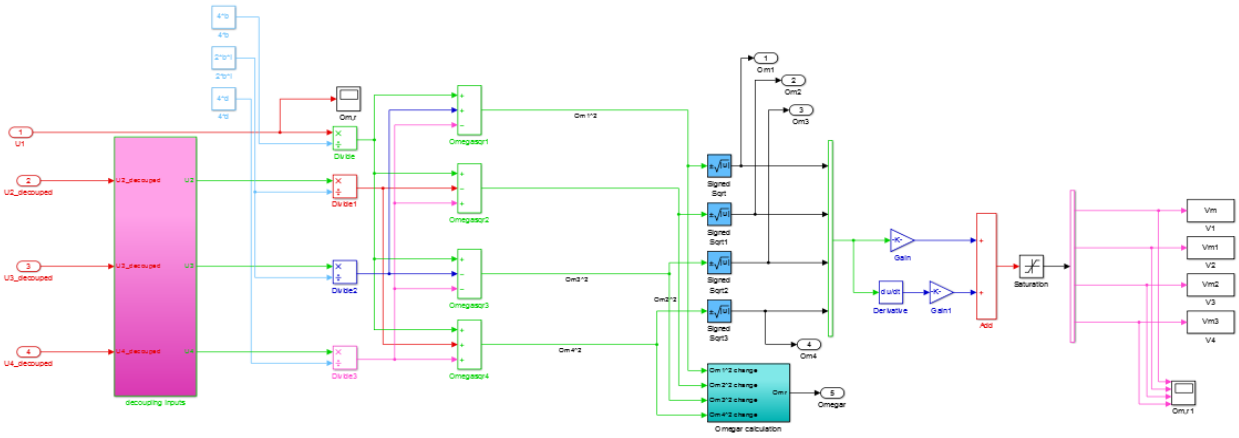

Figure 2_ 7: Simulink schematic representation for rotor Trust Force calculation

To regulate the value of omega in the control process, the following schematic can be observed. It is just used to show inly how the frequency is affected in the control.

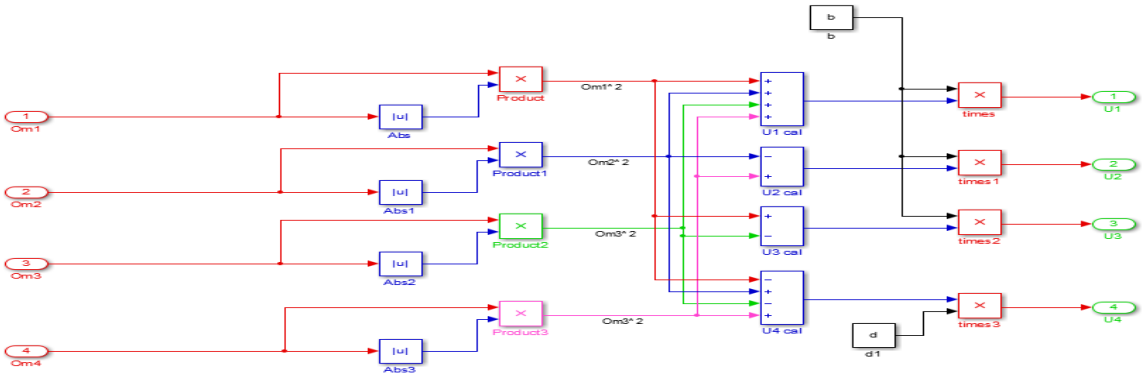

Figure 2_ 8: Simulink schematic representation for input control parameters calculation

2.1.3. Motor dynamics

The DC-motor is an actuator which converts electrical energy into mechanical energy. By applying a DC-current flow into the windings, the rotor turns because of the force generated by the electrical and magnetic interaction. Thanks to the rotor and the commutator geometries, the motor keeps turning while supplied by a DC-voltage on its terminals. The DC-motor has a well-known model which binds electrical and mechanical quantities. This model is composed of the series of a resistor R [Ω], an inductor L [H] and a voltage generator e [V]. The generator **e** (called also ***BEMF***) supplies a voltage proportional to the motor speed. The circuit of the DC-motor is controlled by a real voltage generator v [V] which gives the control input. This may be PWM voltage for many reasons. **[2], [4], [9], [10],** and **[14]**.

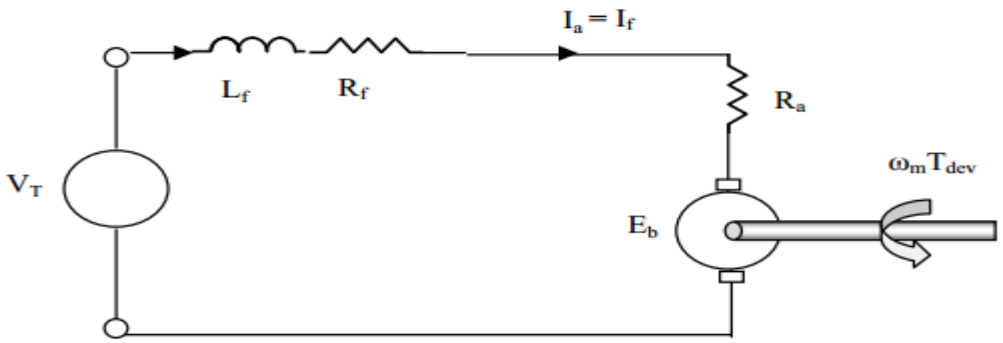

Figure 2_ 9: Electrical equivalent circuit of the Dc motor [**9, 14**]

The KVL analysis yields the following formula **[2]**

$$V_T = V_{Lf} + V_{Rf} + V_{Ra} + E_b \cong V_R + e = R * i + K_m * \omega_m$$
$$and$$
$$J_{Tm}\dot{\omega_m} = \tau_m - \tau_L \xrightarrow{simplified\ to} J_{Tm}\dot{\omega_m} = K_m * i - \tau_L \quad \textbf{(2.37)}$$

However, for better approximation, the generator losses and field effects are kept low and therefore, it is possible to neglect them in the model because of three main aspects:

1) Most of the motors used in robotics show small inductance thanks to construction optimization.

2) The pole (response time) of the electrical part is always much faster than the mechanic one, therefore the speed of the overall system will be determined just by the slowest contribution.
3) It's much easier to solve a first order differential equation rather than a second order one

Combining the two equations of equation **(2.37)** after solving for the current in the first**, t**he model can be given by first order where R is armature constant, K_m is torque constant, μ is a motor efficiency, u is the motor input voltage, K_E is electrical constant and d is drag coefficient. In the equation, the input voltage is fed into the motor so that we can find the rotor speed which in turn gives the propeller speed.

According to **[21]**, the relationship between PWM voltage feed to the motor and the Trust forces produced due to these PWM voltages are given as below in milliseconds.

$$PWM = 0.004 * T + 0.001 \quad \textbf{(2.38)}$$

2.2. Linear model of the UAV Quad rotor about the equilibrium points

The solutions of linear systems may be found explicitly. The problem is that in general, real life problems may only be modeled by nonlinear systems. In this case, we only know how to describe the solutions globally (via null clines). What happens around an equilibrium point remains a mystery so far. The main idea is to approximate a nonlinear system by a linear one (around the equilibrium point). This is the case most of the time (not all the time!) **[12]**.The linearization of state space equation can be given as below in equation ***(2.44)*** around the equilibrium. The equilibrium points are the followings.

$$\{\vartheta_0\,, \theta_0\,, \varphi_0\} rad\,; \{\dot{\vartheta}_0, \dot{\varphi}_0, \dot{\theta}_0\} = 0\,\frac{rad}{sec};\; U_{1\_0} = mg;\;\; N;\; U_{2\_0} = U_{3\_0} = U_{4\_0} = \; 0\; N.m \quad \textbf{(2. 39)}$$

$$for\;\; \dot{x} = f(x, U, w) = Ax + Bu + G\omega \;\; and \;\; z = cx + v$$
$$liniarized\;\; as:$$
$$f(x, U, w) = f(x_0, U_0, w_0) + \frac{\partial f}{\partial x}(x_0, U_0, w_0)(x - x_0) + \frac{\partial f}{\partial U}(x_0, U_0, w_0)(U - U_0) + \frac{\partial f}{\partial w}(x_0, U_0, w_0)(w - w_0) \quad \textbf{(2.40)}$$

$$\dot{\vartheta} \cong \Delta p + q_0\theta_0\Delta\vartheta + \theta_0\Delta r + r_0\Delta\theta \approx \Delta p + \theta_0\Delta r + r_0\Delta\theta \quad (\boldsymbol{from}\;\; \mathbf{2.29})$$

At the hovering condition, all angles and the velocity components are zero and, the equilibrium is maintained.

2.2.1. Coupling between $\vartheta, \varphi, U_2, U_3, U_4$

The linearization of equation **(2.35)** around $\vartheta_0\, rad;\ \varphi_0\, rad;\ \ U_{2\_0} = U_{3\_0} = U_{4\_0} = 0\ N.m$ gives:

$$\ddot{\theta} = \frac{c\varphi_0}{J_{xx}} U_2 - \frac{s\varphi_0}{J_{yy}} U_3;\quad \ddot{\varphi} = \frac{t\vartheta_0 c\varphi_0}{J_{xx}} U_2 + \frac{t\vartheta_0 c\varphi_0}{J_{yy}} U_3 + \frac{1}{J_{zz}} U_4$$
$$\ddot{\vartheta} = \frac{s\varphi_0}{c\vartheta_0\, J_{xx}} U_2 + \frac{c\varphi_0}{c\vartheta_0\, J_{yy}} U_3; \qquad \textbf{(2.41)}$$
$$\ddot{\varphi} = \frac{t\vartheta_0 c\varphi_0}{J_{xx}} U_2 + \frac{t\vartheta_0 c\varphi_0}{J_{yy}} U_3 + \frac{1}{J_{zz}} U_4$$

$for\ which\ \vartheta_0\, rad;\ \varphi_0\, rad\ are\ psi\ and\ phi\ angles\ at\ operating\ point$

2.2.2. Decoupling Inputs

It is important to decouple the inputs since the coupling between $\boldsymbol{x, y, z, U_1, \varphi\ and\ \theta}$ is not enough for the Quadrotor not to swift around the hovering position. Since, at this stage we feed back the position, the velocity, the attitude and rate of change of attitude, controlling is good. Therefore, we consider only the coupling between $\boldsymbol{\theta, \vartheta,\ U_2, U_3, U_4}$.

$using\quad U_{2-decoupled} = \ddot{\theta} J_{xx};\ U_{3-decoupled} = \ddot{\vartheta} J_{yy};\ U_{4-decoupled} = \ddot{\varphi} J_{zz}$:

$$\begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} c\varphi_0 & \frac{J_{xx}}{J_{yy}} c\vartheta_0 s\varphi_0 & 0 \\ -\frac{J_{yy} s\varphi_0}{J_{xx}} & c\vartheta_0 c\varphi_0 & 0 \\ 0 & -s\vartheta_0 \frac{J_{zz}}{J_{yy}} & 1 \end{bmatrix} \begin{bmatrix} U_{2-decoupled} = \ddot{\theta} J_{xx} \\ U_{3-decoupled} = \ddot{\vartheta} J_{yy} \\ U_{4-decoupled} = \ddot{\varphi} J_{zz} \end{bmatrix} \qquad \textbf{(2.42)}$$

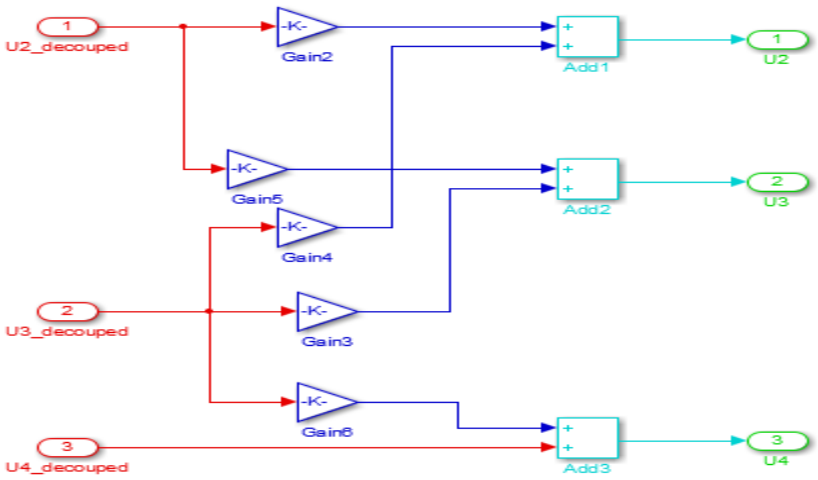

Figure 2_ 10: decoupling the control parameters and Euler angels

2.2.3. Obtaining matrices A and B

Using the state equation of equation **(2.17)** and input vectors from equation **(2.42)** and by introducing the forth input vector due to lifting forcing of Quad rotor, $\boldsymbol{U_1}$, we obtain the state space form as below (***small angle approximation of equation (2.43) will be used with equation (2.46)***).

2.2.3.1. Position Decoupling

Position control keeps the helicopter over the desired point which means the (x, y) horizontal position with regard to a starting point. Horizontal motion is achieved by orienting the thrust vector towards the desired direction of motion. This is done by rotating the vehicle itself. Thus, the position controller outputs the attitude references which are tracked by the attitude controller. The thrust vector orientation in the earth fixed frame is given by $\boldsymbol{R}$, the rotation matrix. Using small angle approximation and the property below in equation **(2.43)**, we obtain equation **(2.45)** with the elimination of constants due to the fact that they are very small.

$$\lim_{\alpha\to 0} c\alpha = 1\,,\ \lim_{\alpha\to 0} s\alpha = \alpha\,,\ \lim_{\alpha\to 0} s\alpha * \lim_{\alpha\to 0} c\alpha = \alpha \text{ and } \lim_{\alpha\to 0} s\alpha * \lim_{\alpha\to 0} s\alpha = 0 \qquad \textbf{(2.43)}$$

$$R_v^b = \begin{bmatrix} c\varphi c\theta & s\varphi c\theta & -s\theta \\ s\vartheta s\theta c\varphi - s\varphi c\vartheta & s\vartheta s\theta s\varphi + c\varphi c\theta & s\vartheta c\theta \\ c\vartheta s\theta c\varphi + s\varphi s\vartheta & c\vartheta s\theta s\varphi - c\varphi s\vartheta & c\vartheta c\theta \end{bmatrix} \qquad \textbf{(2.44)}$$

$$R_b^v = \lim_{(\vartheta,\theta,\varphi)\to 0} (R_v^b)^T = \begin{bmatrix} 1 & \Delta\varphi & \Delta\theta \\ \Delta\varphi & 1 & -\Delta\vartheta \\ -\Delta\theta & \Delta\vartheta & 1 \end{bmatrix} \quad \textbf{(2.45)}$$

From equation **(2.35)**, the acceleration components of x, y and z axis are given below as:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 1 & \Delta\varphi & \Delta\theta \\ \Delta\varphi & 1 & -\Delta\vartheta \\ -\Delta\theta & \Delta\vartheta & 1 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 1 & \Delta\varphi & \Delta\theta \\ \Delta\varphi & 1 & -\Delta\vartheta \\ -\Delta\theta & \Delta\vartheta & 1 \end{bmatrix} * \left(\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} + \begin{bmatrix} vr - wq \\ wp - ur \\ uq - vp \end{bmatrix} \right) \quad \textbf{(2.46)}$$

Using this equation and all the points about operating points, ***TAYLOR SERIES*** linearization yields matrix A and B as below. The detail of this linearization process is given on ***appendix A***. the operating points are: $\vartheta_0, \theta_0, \varphi_0, w0, v_0, u_0, r_0, q_0 \; and \; p_0$.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\vartheta} \\ \dot{\theta} \\ \dot{\varphi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & \vartheta_0 & \theta_0 & 0 & w0 & v_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varphi_0 & 1 & -\vartheta_0 & -w0 & 0 & u_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\vartheta_0 & \vartheta_0 & 1 & v_0 - u0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_0 & -q_0 & 0 & \boldsymbol{k_{11}} & 0 & 0 & -w_0 & v_0 \\ 0 & 0 & 0 & -r_0 & 0 & p_0 & \boldsymbol{k_1} & \boldsymbol{k_2} & 0 & w_0 & 0 & -u_0 \\ 0 & 0 & 0 & q_0 & -p_0 & 0 & \boldsymbol{k_3} & \boldsymbol{k_4} & 0 & -v_0 & u_0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_0 & 0 & 1 & 0 & \theta_0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -r_0 & 0 & 0 & 0 & 1 & -\vartheta_0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q0 & 0 & 0 & 0 & \vartheta_0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{k_6} & \boldsymbol{k_5} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{k_8} & 0 & \boldsymbol{k_7} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{k_{10}} & \boldsymbol{k_9} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ u \\ v \\ w \\ \vartheta \\ \theta \\ \varphi \\ p \\ q \\ r \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{J_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix} \begin{bmatrix} U_1 = T_t \\ U_2 = J_{xx}\ddot{\vartheta} \\ U_3 = J_{yy}\ddot{\theta} \\ U_4 = J_{zz}\ddot{\varphi} \end{bmatrix}$$

(2.47)

Table 2_ 2 : values and description of constants used for the matrix

Constant	*Representation*	*Description*
$\boldsymbol{k_1}$	$\mathrm{g} * \cos(ph_0) * \cos(th_0)$	Gravitational effect due angle phi on y-axis body velocity from rotation
$\boldsymbol{k_2}$	$-\mathrm{g} * \cos(ph_0) * \mathrm{sine}(th_0)$	Gravitational effect due angle theta on y-axis body velocity from rotation
$\boldsymbol{k_3}$	$-\mathrm{g} * \cos(th_0) * \mathrm{sine}(ph_0)$	Gravitational effect due angle phi on z-axis body velocity from rotation
$\boldsymbol{k_4}$	$-\mathrm{g} * \mathrm{sine}(ph_0) * \mathrm{sine}(pth_0)$	Gravitational effect due angle theta on z-axis body velocity from rotation
$\boldsymbol{k_5}$	$\left(\frac{\mathrm{Jyy} - \mathrm{Jzz}}{\mathrm{Jxx}}\right) * q_0$	Carioles' effect due to body rates of yaw angle
$\boldsymbol{k_6}$	$\left(\frac{\mathrm{Jyy} - \mathrm{Jzz}}{\mathrm{Jxx}}\right) * r_0$	Carioles' effect due to body rates of theta angle
$\boldsymbol{k_{11}}$	$-9.81cos(\theta_0)$	Gravitational effect due angle theta on y-axis body velocity from rotation
$\boldsymbol{k_7}$	$\left(\frac{\mathrm{Jzz} - \mathrm{Jxx}}{\mathrm{Jyy}}\right) * p_0$	Carioles' effect due to body rates of yaw angle
$\boldsymbol{k_8}$	$\left(\frac{\mathrm{Jzz} - \mathrm{Jxx}}{\mathrm{Jyy}}\right) * r_0$	Carioles' effect due to body rates of phi angle
$\boldsymbol{k_9}$	$\left(\frac{\mathrm{Jxx} - \mathrm{Jyy}}{\mathrm{Jzz}}\right) * p_0$	Carioles' effect due to body rates of theta angle
$\boldsymbol{k_{10}}$	$\left(\frac{\mathrm{Jxx} - \mathrm{Jyy}}{\mathrm{Jzz}}\right) * r_0$	Carioles' effect due to body rates of phi angle

2.3. Sensors and Measurement Variables

Complementary Filter uses **three** main components to estimate orientation:

1) Gyroscope measurements
2) Accelerometer measurements and
3) SONAR measurements.

Since data's of the first component from the second and third components are different from the others, two methods has been used to estimate angles **[20]**. Under this topic, two kinds of sensors will be introduced **[10]**.

2.3.1. 3-Axis gyroscope

The gyroscopes used onboard of Quad rotor are Micro-machined Electro Mechanical Systems (***MEMS***) sensors, which have compared to mechanical and optical gyroscopes and are relatively cheap to manufacturer. At present, MEMS sensors cannot match the accuracy of optical devices, however they are expected to do so in the future. Some advantages of MEMS sensors are: small size, low weight rugged construction and low power consumption. MEMS gyroscopes make use of the ***Carioles effect***, which describes the effect that in a frame of reference rotating at an angular velocity ω, a mass m moving with velocity v experiences a force **[10]**:

$$F = -2 * m(wxV) \quad \textbf{(2.48)}$$

a) MEMS error characteristics

As mentioned, MEMS gyroscopes are far less accurate than optical gyroscopes. To model them correctly, it is necessary to first describe the errors that arise in them. The ability to compensate for these inaccuracies is an important factor to determine the quality of the device such as:

1) Constant bias
2) Thermo-Mechanical White Noise or Angular Random Walk
   Flicker noise / Bias drift
3) Temperature Effects
4) Calibration Errors

b) Assumptions

Assumptions for the gyroscope model:

1) Noise with zero mean and variance σ^2
2) Bias with random walk model

c) Mathematical model

The measured angular rates are assumed to have a bias and white noise:

$$\begin{bmatrix} p_m \\ q_m \\ r_m \end{bmatrix} = \begin{bmatrix} p + \lambda_{p_m} + w_{p_m} \\ q + \lambda_{q_m} + w_{q_m} \\ r + \lambda_{r_m} + w_{r_m} \end{bmatrix} \quad \textbf{(2.49)}$$

The derivatives of the gyroscope biases are modelled as white noise:

$$\begin{bmatrix} \dot{\lambda_{p_m}} \\ \dot{\lambda_{p_m}} \\ \dot{\lambda_{p_m}} \end{bmatrix} = \begin{bmatrix} w_{\dot{\lambda_{p_m}}} \\ w_{\dot{\lambda_{p_m}}} \\ w_{\dot{\lambda_{p_m}}} \end{bmatrix} \quad \textbf{(2.50)}$$

2.3.2. 3-Axis accelerometer

The ADXL330 sensor is a monolithic, three-axis, force-balanced, capacitive accelerometer. The proof mass, supported by the four springs, is movable along any axis (X-Y-Z) in response to an inertial force. The proof mass is attached to the common electrode of sets of differential capacitors, with the other electrodes of each differential capacitor pair being fixed. Each sense axis requires a separate set of differential capacitors oriented to the appropriate direction. The differential capacitance is measured by applying one of two complementary (180 out-of-phase) square-waves to each of the fixed plates of the differential capacitors **[10]**.

a) Specific Accelerometer Error characteristic

Additional to the errors mentioned for the MEMS gyroscopes, the accelerometer measurement is influences by the position of the sensor with respect to the center of gravity of the vehicle. Because the sensor is placed close to the center of gravity, this effect is ignored.

b) Assumption

Assumptions for the accelerometer model:

1) Noise with zero mean and variance σ^2
2) Bias with random walk model
3) Placed in the center of gravity of the vehicle

c) Mathematical model

In reality, also a biases and noise are measured like that of accelerometer. They can be modelled as follows.

$$\begin{bmatrix} A_{x_m} \\ A_{y_m} \\ A_{z_m} \end{bmatrix} = \begin{bmatrix} A_x + \lambda_{A_{x_m}} + w_{A_{x_m}} \\ A_y + \lambda_{A_{y_m}} + w_{A_{y_m}} \\ A_z + \lambda_{A_{z_m}} + w_{A_{z_m}} \end{bmatrix} \quad \textbf{(2.51)}$$

2.4. System Identification

This thesis is done only at a simulation level. In other words, to identify the components used in this thesis, the open loop analysis from the point of view of closed loop response must be done. This is called system identification. This is the task to be done in in the lab. However, there is no identification done since what is done in this thesis is only simulation analysis. All identified components are taken from some related materials and sited as a reference from where they were taken. Theses identified components are the three major axis components of UAV quad rotor and the two control input coefficients like drag force constant and moment constants including the gyro bias. These references are reference [2] and reference [10].

CHAPTER THREE

3. LQG CONTROL OF UAV QUAD ROTOR

3.1. Introduction

The Unmanned Aerial Vehicle quad rotor is a great platform for control systems research as its nonlinear nature and under-actuated configuration make it ideal to synthesize and analyze control algorithms. The choice of the controller depends on the goal of the control. The most obviously used controllers in this field of study are: PID, Linear Quadratic Regulator (LQR), Sliding mode, Back Stepping, Feedback Linearization, Adaptive, Robust, Optimal, $L\infty$, $H\infty$, Fuzzy logic and Artificial neutral networks. The following table summarizes their characteristics.

Table 3_ 1: Controllers Characteristics

	Characteristics														**Type**
Control algorithm	***Robustness: Rob***	***Adaptiveness: Ad***	***Optimality: Opt***	***Intelligence: Int***	***Tracking Ability: TA***	***Fast Convergence: FC***	***Precision: Prc***	***Simplicity: Sim***	***Disturbance Reject: DR***	***Unmolded Parameters***	***Handling: UPH***	***Manual Tuning: MT***	***Noise Signal: NS***	***Chattering or energy Loss: EL***	
PID	1	0	0	0	1	1	1	2	0	0		2	2	0	Linear
Intelligent PID	1	0	0	2	1	1	1	1	0	0		0	1	0	Linear
LQR	0	2	1	0	1	1	0	1	1	0		1	1	0	Linear
LQG	***0***	***2***	***2***	***0***	***1***	***1***	***0***	***0***	***2***	***0***		***1***	***0***	***0***	***Linear***
L_∞	0	2	2	0	1	2	2	0	1	0		0	0	0	Linear
H_∞	2	1	2	0	2	0	1	0	1	1		0	0	0	Non Linear

Control algorithm	Rob	Ad	Opt	Int	TA	FC	Prc	Sim	DR	UPH	MT	NS	EL	Type
FBL	1	1	0	0	2	2	2	1	1	1	0	1	0	Non Linear
SMC	1	2	1	0	2	2	2	1	2	1	0	0	2	Non Linear
Back stepping	0	2	0	0	2	0	1	0	2	1	0	0	0	Non Linear
Fuzzy Logic	1	1	1	2	1	1	1	1	1	0	1	0	0	Non Linear
Neural Networks	1	2	2	2	1	1	1	0	1	1	0	0	0	Non Linear
Genetic	1	2	2	2	1	1	1	0	1	2	0	0	0	Non Linear

Legend: 0 - low to none; 1 - average; 2 – high.

The main objective of this thesis is to design the controller which controls the quad rotor so that it can move and hover ***optimally*** with a good ***disturbance rejection***. This is done by using **Linear Quadratic Gaussian control *(LQG*)** controller.

3.2. LQG Control

LQG is a combination of ***LQR*** and ***Kalman Filter***. It is a relevant control methods that may benefited from ***state estimators***. The parameters that are needed to be controlled through this controller are:

a) Attitude control $(\vartheta, \theta, \varphi)'$
b) Altitude control (z) and
c) Position control (x, y)

3.2.1. Design of Kalman Filter

An observer is an algorithm for estimating the values of state variables of a dynamic system. Why can such state estimates be useful? The following points are brief answers.

1) ***Supervision***: State estimates can provide valuable information about important variables in a physical process, for example feed composition to a reactor, environmental forces acting on a ship, load torques acting on a motor, etc.

2) ***Control***: In general, the more information the controller has about the process it controls, the better (more accurate) it can control it. In particular, some control methods assumes that the state of the process to be controlled are known. If the state variables are not measured, they may be estimated, and the estimates can be used by the controller as if they were measurement. Note that even process disturbances and process parameters can be estimated. The clue is to model the disturbances or parameters as ordinary state variables.

Observers are calculated from specified estimator error dynamics, or in other words: how fast and stable we want the estimates to converge to the real values (assuming we could measure them). An alternative to observers is the ***Kalman Filter*** which is an estimation algorithm based on ***stochastic theory***. The Kalman Filter produces state estimates that contain a minimum amount of noise in the assumed presence of random process disturbances and random measurement noise. One particular drawback about observers is that they are not straightforward to design for systems having more than one measurement, while this is straightforward for Kalman Filters **[17]**.

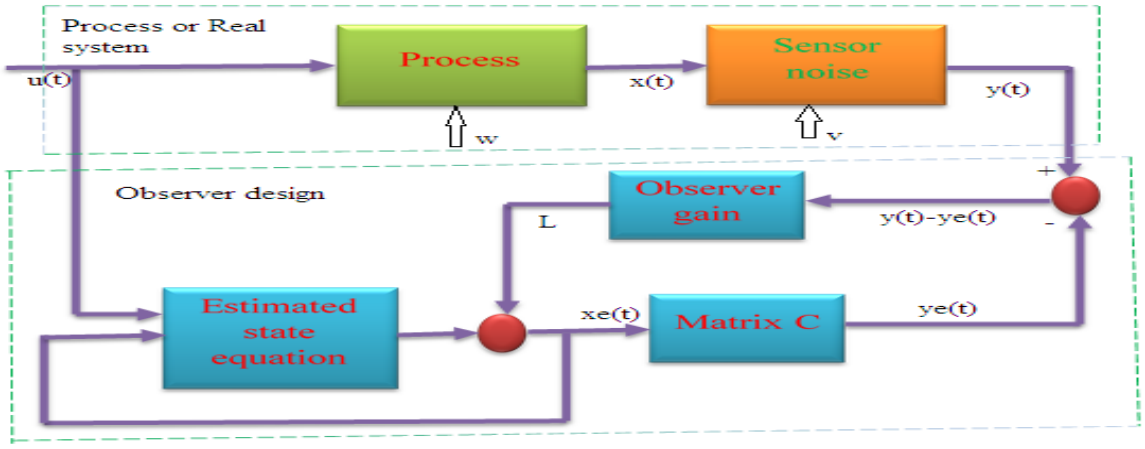

Figure 3_ 1: Process with observer

For a non-linear process, the Kalman Filter **(KF)** is used which linearizes the nonlinear system about the current ***mean*** and ***covariance***. Assuming the sensor noise exists, a direct Kalman filter can be derived as follows. Assuming the process has a state vector $x \in R_e^n$ described by the non-linear stochastic difference equation with the measurement $z \in R_e^m$:

$$\dot{x} \approx f(x,u,w) \;\; and \;\; z \approx h(x,v) \dots\dots\dots\dots\dots\dots\dots from\ measurement$$
$$using \quad p(v) \sim N(0,Q)\ and\ p(w) \sim N(0,R)$$
$$\dot{\hat{x}} = f(\hat{x},\hat{u},0)\ and\ \hat{z} = f(\hat{x},\ 0)$$

$$\dot{x} \approx \dot{\hat{x}} + A(x-\hat{x}) + w_{k-1} \rightarrow \dot{x} = Ax + Bu + Gw$$
$$z_k \approx \bar{z}_k + H_k(x_k - \hat{x}_k) + v \rightarrow y = c_y x + D_y w\ and\ z = C_z x + D_z v \qquad \textbf{(3.1)}$$
$$where:$$

$$\begin{bmatrix} A \\ [B\ G] \\ [C_x\ C_y] \\ [D_y\ D_z] \end{bmatrix} = \begin{bmatrix} \frac{df}{dx}|_{\hat{x},\hat{u},0} \\ \frac{df}{dw}|_{\hat{x},\hat{u},0} \\ \frac{dh}{dx}|_{\hat{x},\ 0} \\ \frac{dh}{dv}|_{\hat{x},\ 0} \end{bmatrix} \qquad J = \begin{bmatrix} A & [B\ G] \\ [C_x\ C_y] & [D_y\ D_z] \end{bmatrix} = \begin{bmatrix} \frac{df}{dx}|_{\hat{x},\hat{u},0} & \frac{df}{dw}|_{\hat{x},\hat{u},0} \\ \frac{dh}{dx}|_{\hat{x},\ 0} & \frac{dh}{dv}|_{\hat{x},\ 0} \end{bmatrix}$$

In the above equation, w and v are the process and measurement noise respectively where Q and R are a process noise covariance and a measurement noise covariance. The operation of the Kalman filter is divided into two steps:

1) prediction step and
2) correction step

3.2.1.1. Prediction step: (Between measurements)

The state prediction for the above non-linear stochastic difference equation is given by:

$$\dot{\hat{x}} = f(\hat{x},\hat{u},0) \xrightarrow{by\ simple\ Euler\ Integration} x + \dot{x}\Delta t \qquad \textbf{(3.2)}$$

The best way to analyze this equation is using the Covariance matrix of the state prediction error as below from the fact that this stage predicts state between the measurements. It accounts for the noise which introduced previously and intended to affect the next stage measurement from the sensor point of view. The evolution matrix, matrix P is needed because of the fact that the mean

and covariance of the state determines how to select the control inputs so that the error will be minimized as much as possible. From the following derivation, it shows that the evolution matrix is selected from the expectation of estimates. The logic is that the evolution matrix is the expectation of the product of the estimates and transpose. The following procedure briefly describes how to find the evolution matrix.

$$\begin{gathered}
for\ \breve{x}\ isestimate\ error: \\
\breve{x} = e_{k-1} = x_{k-1} - \hat{x}_{k-1}\ and\ \dot{\breve{x}} = e_k = x_k - \bar{x}_k : \\
\dot{\breve{x}} = A_k\breve{x} + w_{k-1} \xrightarrow{implies,\ in\ case\ contineous,} \breve{x} = e^{At}\breve{x}_0 + \int_0^t e^{A(t-\tau)}(w(t) = G\omega(t))dt \\
P = E\{\breve{x}\breve{x}^T\} \xrightarrow{implies} \dot{P} = \frac{d}{dt}(E\{\breve{x}\breve{x}^T\}) = E\left(\dot{\breve{x}}\breve{x}^T + \breve{x}\dot{\breve{x}}^T\right) = E\{A\breve{x}\breve{x}^T + \breve{x}\breve{x}^T A^T + G\omega\breve{x}^T + \breve{x}\omega^T G^T\} \\
\dot{P} = AP + PA^T + GE\{\omega\breve{x}^T\}^T + E\{\breve{x}\omega^T\}G^T = AP + PA^T + GQG^T \\
where: \\
E\{\breve{x}\omega^T\} = 0.5Q \qquad and\ Q = Q^T
\end{gathered} \tag{3.3}$$

3.2.1.2. Correction step: (At measurements)

This stage corrects the measurement from the feedback gained from the sensor measurements using magnetometers and IR sensors. Magnetometer measures the magnetic field of z-axis magnetic field from which the other axes magnetic field are measured. Accelerometer and gyroscope measure the noisy body acceleration and noisy body rates of angles.

$$\begin{gathered}
\breve{x}^+ = {e_{k-1}}^+ = x - \hat{x}^+ \\
Kest * (y - \hat{y}^-) = x - (\breve{x}^+ + \hat{x}^-) \\
where\ \ \breve{x}^+\ is\ current\ value\ and\ \hat{x}^-\ previous\ stimate \\
\breve{x}^+ = \left((x - \hat{x}^-) = \breve{x}^-\right)(I - Kest * H) - Kest * \omega \\
P^+ = E\left\{\breve{x}^+{\breve{x}^+}^T\right\} = E\left((I - Kest * H)\hat{x}^- - Kest * \omega\right)(I - Kest * H)\hat{x}^- - Kest * \omega)^T) \\
Objective: \ to\ minimize\ \ P^+\ and\ say\ Kest = L: \\
\frac{d}{dL}P^+ = 0 = \frac{d}{dL}E\{(P^-(I - (LH)^T) - LHP^-(I - (LH)^T) + LRL^T)\} \\
where: \\
Expec\left\{\begin{bmatrix} \hat{x}^-\hat{x}^{-T} & \hat{x}^-\hat{x}^{-T}H^TL^T & \hat{x}^-\omega^TL^T \\ LH\hat{x}^-\hat{x}^{-T} & LH\hat{x}^-\hat{x}^{-T}H^TL^T & LH\hat{x}^-\omega^TL^T \\ L\omega\hat{x}^{-T} & L\omega\hat{x}^{-T}H^TL^T & L\omega\omega^TL^T \end{bmatrix}\right\} = \begin{bmatrix} P^- & P^-H^TL^T & 0 \\ LHP^- & LHP^-H^TL^T & 0 \\ 0 & 0 & LR^TL^T \end{bmatrix} \\
Kest = P^-H^T(R + HP^-H^T)^{-1} \\
P^+ = (I - Kest * H)P^-\ ;\ in\ descreate\ form\ \ P_k = (I - LH)P_{k-1}
\end{gathered} \tag{3.4}$$

However, in this thesis, this step is used by taking the constant value of $\boldsymbol{p}$ without no update, i.e. the value of $\boldsymbol{Kest}$ is kept constant for the design of LQG and calculated as follows. This is due to the fact that this gain is calculated ***offline*** once for the input commands. In the equation **(3.4),** $Expec$ is meant the expectation and P^{+} is meant the LQR gain currently where as P^{-} is for the previous case.

3.2.1.3. Calculation of Kest

using the values of the matrices A, B, C, D; $the\ Kalman\ gain\ \boldsymbol{p}$,
Kest $can\ be\ found\ as\ below$. **[13], [15]:**

$$\begin{gathered}\textbf{Kest} = p * C * inv(Rn) \ldots\ldots\ldots\ldots .where: \\ p = are(A, Cn, Q), \\ Cn = (C'/Rn) * C, \\ Rn = covariance * C, \\ Q = Bw * Qn * Bw', \\ Qn = covariance^{2} * C + B * B'\end{gathered} \tag{3.5}$$

The value of **Kest** is found in a manner that there will be no loop formed in the model. This is known as loop recovery.

3.2.2. Linear Quadratic Regulator (LQR)

Linear Quadratic Regulator (LQR) is the optimal theory of pole placement method. LQR algorithm defines the optimal pole location based on two cost function. To find the optimal gains, one should define the optimal performance index firstly and then solve ***Algebraic Riccati Equation*** *(ARE)*. LQR does not have any specific solution to define the cost function to obtain the optimal gains and the cost function should be defined in iterative manner. This technique is known as a very robust control strategy and in case of the angle controller, it is better to rely on. The idea of this technique is to find a **K** feedback, which minimizes the **J** function, given that the weighting matrices: **Q** and **R**. **Q** matrix and **R** matrix are positive definite symmetric matrix and semi positive definite symmetric matrix respectively.

$$J = 0.5\int_0^{\infty}(x^T Qx + u^T Ru)dt$$
$$subject\ to \quad \dot{x} = Ax + Bu$$
$$goal: finding\ the\ optimal\ control\ input\ u^*\ from\ the\ ARE\ below$$
$$A^T P_0 + P_0 A - P_0 B R^{-1} B^T P_0 + C^T QC = 0 \ldots\ldots\ldots P_0 = are(A, B^*, Q)\ \ where\ B^* = BR^{-1}B^T$$
$$u^* = -R^{-1}B^T P_0 x = -k_{LQR}x$$
$$and$$
$$\dot{x}^* = \left(A - Bk_{kalman\ gain}\right)x^* \ldots\ldots\ldots shows\ the\ poles\ are\ at: sI - A + Bk_{kalman\ gain} = \bar{0}$$
$$from\ this\ x(s) = \frac{1}{sI+(-A+Bk_{kalman\ gain})}\ implies\ \bar{x}(t) = \exp(-\left(-A + Bk_{kalman\ gain}\right) * t = \bar{0})$$
$$J^* = 0.5{x^*}^T p x^*$$

(3.6)

Coefficients of Q limit the amplitude of the state variables and coefficients of R limit the amplitude of the inputs. The larger these coefficients, the smaller the state/input variables will be. The Q matrix has unitary coefficients except for the controlled variables ($\boldsymbol{x, y, z\ and\ \varphi}$). Indeed, the priority is to make the UAV reach its desired position as fast as possible. In other terms, the main goal is to minimize the $L_{2-norm}\ of\ x, y, z\ and\ \varphi$. Multiplying the associated coefficients by say, ***10***, advanced these variables **[8], [13**]

$$R = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{0.1} \end{bmatrix}$$

$$Q = [k_1 \quad k_2 \quad k_3 \quad k_4 \quad k_5 \quad k_6 \quad k_7 \quad k_8 \quad k_9 \quad k_{10} \quad k_{11} \quad k_{12}]^T I_{12x12}$$

selection: as if the first one third of the ***k's*** and the **9th** are valued as below where as the two third of the last except the **9th**are taken as if **1**

$$Q = \begin{bmatrix} \mathbf{10000} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{10000} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{25000} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{10000} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(3.7)

The input amplitudes don't need to be minimized. With these weighting matrices, the controller is in fact designed to minimize the $L_{2-norm}\ of\ x, y, z\ and\ \varphi$, without considering the other variables too much. Since the weighting matrices are adjusted to correct the gain of LQR, it is by trial and error we can found the desired LQR gain. We may try more than a dozen. The best thing to do is starting from the unity matrix and travelling across the diagonal. We need to change only the states we need to compare with the command.

CHAPTER FOUR

4. SIMULATION STUDIES AND ANALYSIS OF RESULTS

4.1. Introduction

In this chapter, the quad rotor simulator is presented. This tool is very helpful to verify the correctness of the helicopter dynamic model and to test the control algorithms performance. It is also possible to evaluate the behavior of the quad rotor through a 3D view. This simulator has been modeled with the Mat lab tool Simulink.

The first section (***4.1. System structure***) introduces the simulation tools MATLAB/Simulink. Furthermore it provides an overview of the quad rotor system architecture and it gives a brief description of blocks and commands.

The second section (***4.2. output obtained, errors and commands***) shows the analysis of the previously introduced blocks. The analysis is done both at cruising and hovering position. The problem from noise introduced or problem of completely rejecting the disturbance is also observed compared to that of no noise assumption case.

The third section (***4.3: Trajectory Following***) underlines the quad rotor capacity to track the path. This tracking can be observed through 3D visualization. Thanks to this visualization, it is easy to analyze the position and the orientation of the quad rotor, hence verify its performance.

4.2. System Simulation Model

Mat lab is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis and numeric computation. It is used in engineering and science because of its easy interface and powerful commands. The main strengths of Mat lab are **[2]:**

1) It is relatively easy to learn.
2) Its optimized code is relatively quick when performing matrix operations.
3) It may behave like a calculator or as a programming language.

4) It is an interpreted language, errors are easier to be fixed.

The main weakness of MATLAB/ Simulink software is, its ***slowness***: it is almost always much slower than a compiled language such as C (since it is an interpreted language).

Simulink is an environment for multi domain simulation and Model-Based Design for dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries which allow to design, simulate, implement, and test a variety of time-varying systems. Simulink has been chosen in this work for its easy and clear graphic interface **[2].**

The model of the whole system is composed of several interconnected blocks in a classic feedback structure. "***Dynamics***" represents the physics of the quad rotor and provides the position, velocity and acceleration of both linear and angular quantities. The input of this block is the voltage given to the motor drivers (V control) from the "***control***". "***Sensors***" models the ACCELEROMETRE, GYROSCOPE and SONAR. It gives its data to the "control", while it receives information about the quad rotor motion from "dynamics". "***Inputs***" acquires the task references directly from the helicopter remote controller or from a signal builder. As for the "GYROSCOPE AND SONAR", the output of "inputs" is directly connected with "control".

a) The disturbance block

This block enable or disable the accelerometer measurement as the input of white noise. The effect of white noise is observed if the gain is 1.

b) The sensor noise block

This block shows the effect of gyro for the body rates of angels. The gyro measures the rates of body angles and sum up with the body rates of angels so the effect of gyroscopic torque is analyzed.

c) The ***'LQG Controller'*** block

This block is the main part of this thesis. It consists the observer design or Kalman filter '**Kf**' and ***LQR*** feedback gain '**K**' for the modeled state space equation. On this block, the left side block named as FROM WORKSPACE is needed to declare the command to test whether the quad rotor

can follow the trajectory or not. The switch is simply used to see the effect of declared trajectory and step input for the commands Xcom, Ycom, Zcom and Psicom. On this block, **KF** observer is used to filter the output we need further since it is the gain of the error from Yout and Yestimates. The LQR gain is needed to minimize the error from the command and the output. The following figures shows the difference between the errors of LQR gain and Kalman gain. From these two combination, the magnitude of errors on the state error is higher than that of output errors. This shows the errors are minimized and filtered further using Kalman Filters. However, the desired output is obtained if the covariance is almost 1 or the noise is a white noise. The two gains are obtained using Mat lab and given on the appendix.

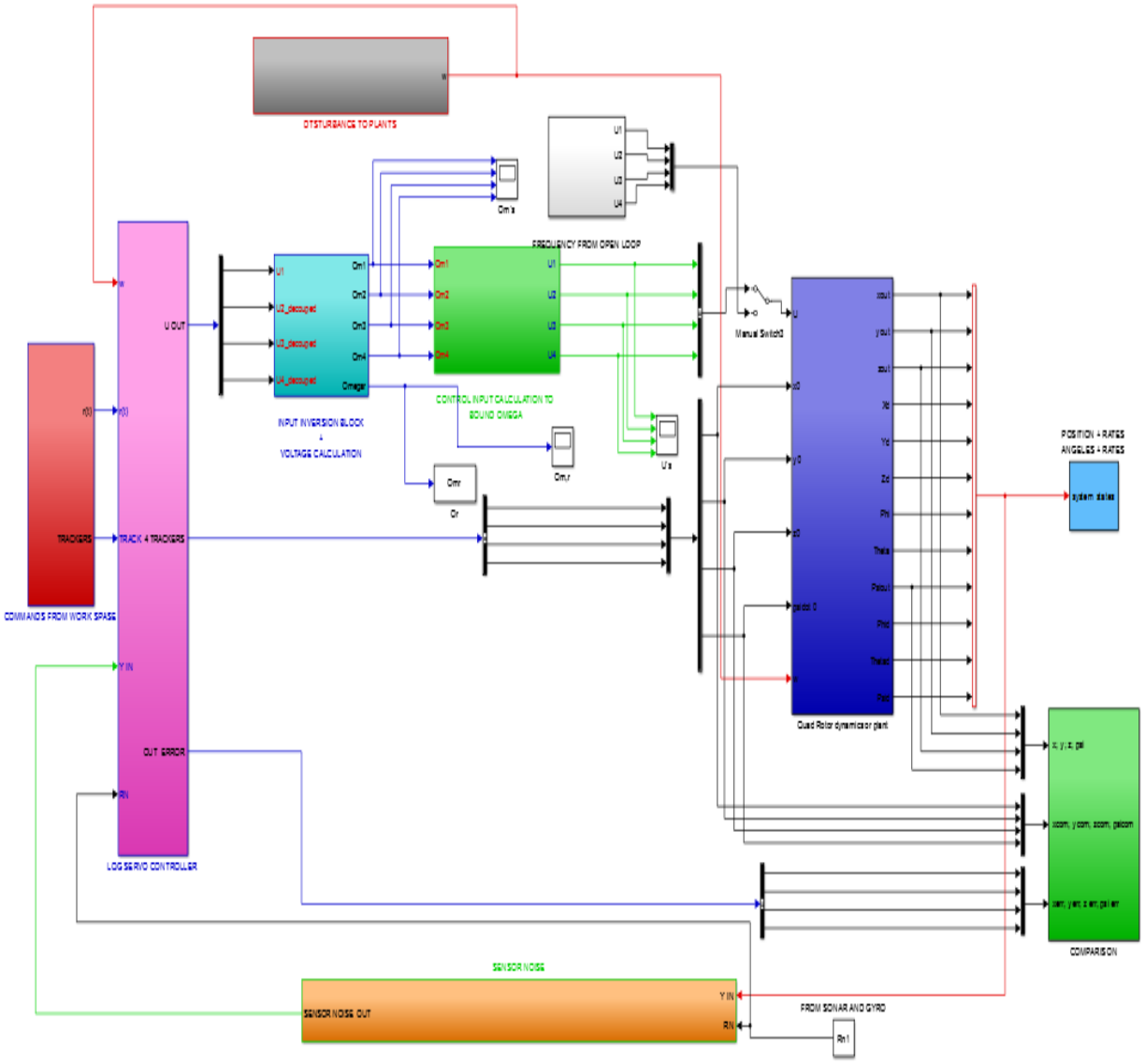

Figure 4_ 1: Simulink representation for system structure of linearized model

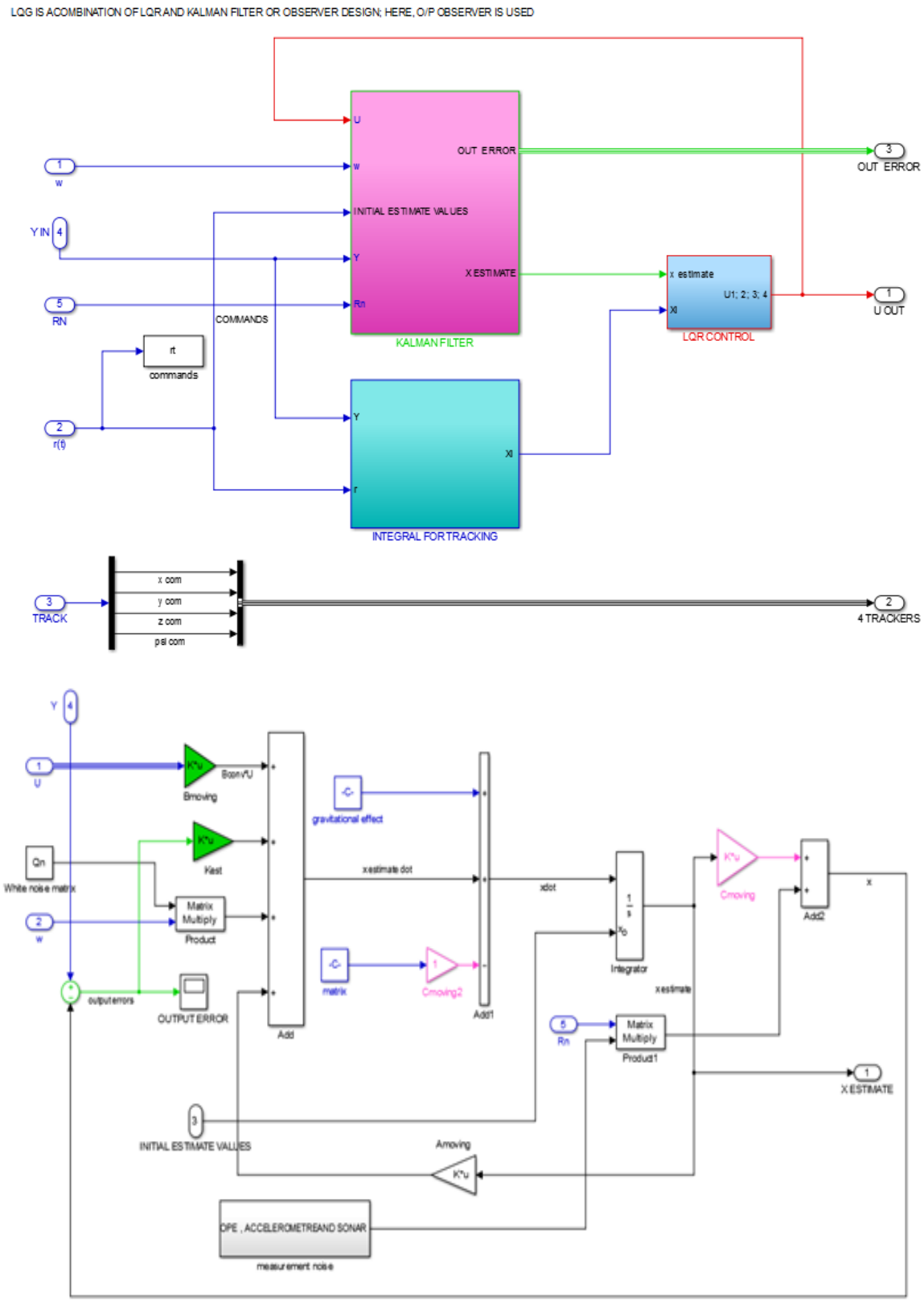

Figure 4_ 2: LQG Track Controller for linearized dynamics

d) The '***Quad rotor dynamics or plant'*** block

This block has the state space model and the output is obtained following an ordinary exponential function of homogeneous equation in case of Linearized model. For the case of linearized model of LQG control, the initial conditions are provided through the integral. In this block, the matrices A and B are obtained from the linearized model obtained by Taylor series. The problem observed here is that the Euler angles rates for both body and earth reference frames are the same. To truck the states with a command, we need to directly connect the command so as to keep the initial point with the command. The initial points are given to the integral initial states connected externally. The block named as plant noise and sensor noise is the acceleration component noise from accelerometer. The three body component accelerometer is assumed to be a white noise. The sonar is also tried to use a white noise for the altitude variation. In this block, the plant disturbance from accelerometers are introduced as body acceleration.

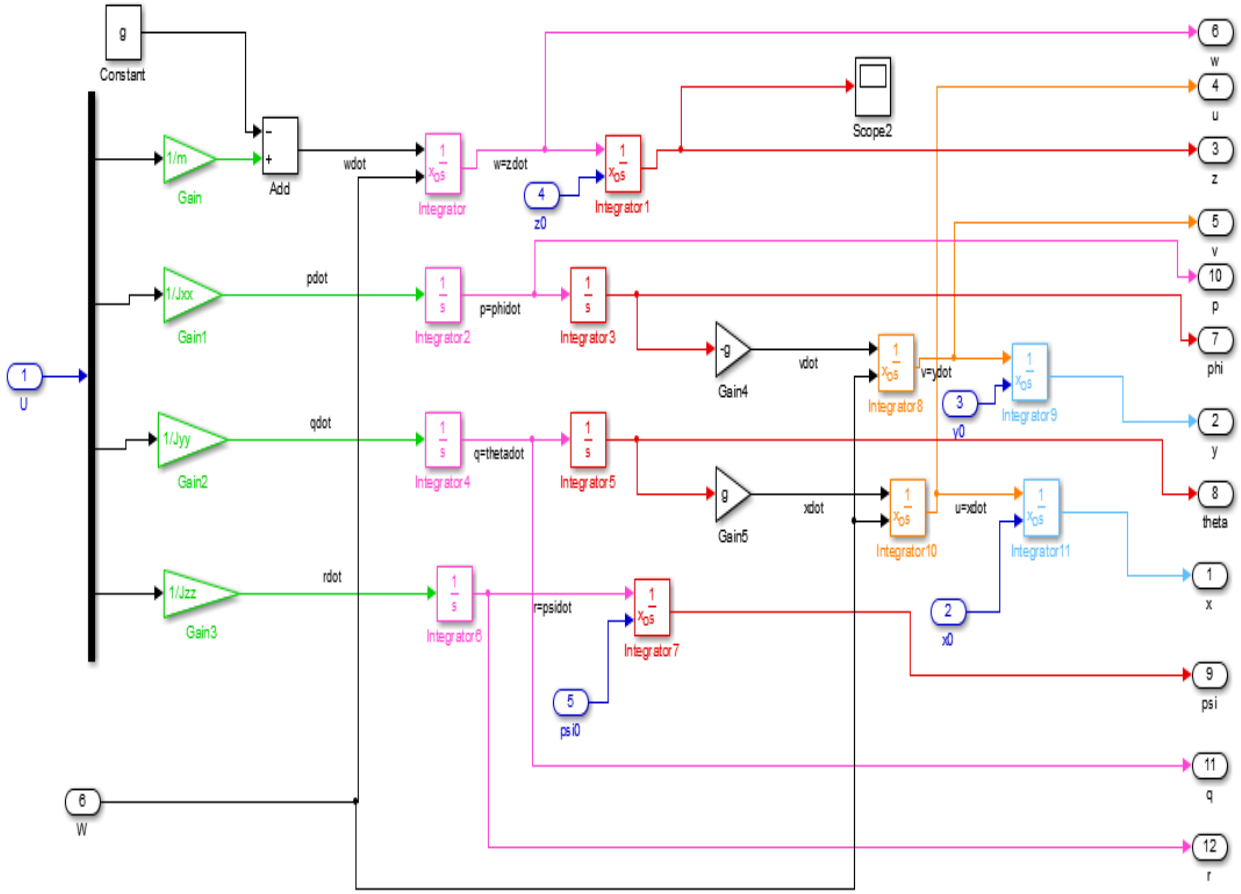

Figure 4_ 3: Linear modelling of UAV quad rotor

4.3. Simulation Results and Analysis

Under this topic, the output obtained is analyzed. The objective is to fond the output which is as much as possible equal to the command we declare for the quad rotor to follow. The four command specified in this thesis are:

1) X-axis command
2) Y-axis command
3) Z-axis command
4) Psi Euler angle command

The following figures shows the command, errors and output for each of the above commands both for the case of:

a) No noise and
b) With noise introduced

A) Model verification using open loop response

The verification of model is done through the open loop control. The control inputs are varied by hand to see the effect of the four control inputs on the roll angle, yaw angle, psi angle and the positions x, y and z.

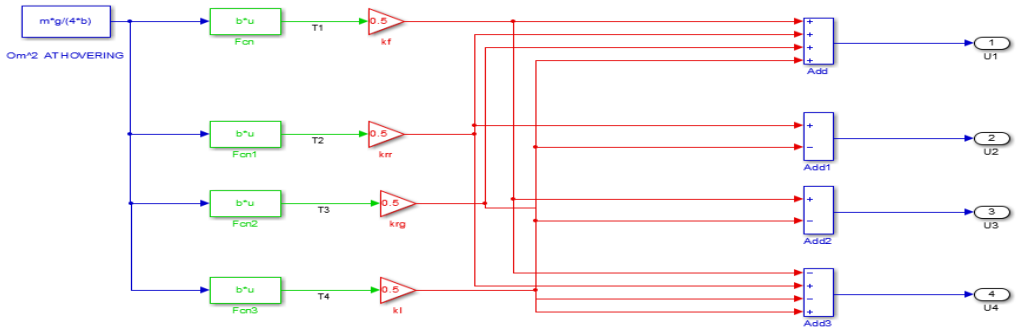

Figure 4_ 4: Open loop input control verification

a) The first one is the state in which the UAV quad rotor is stationary. At this case, the UAV quad rotor stays where it is due to the fact that the rotor frequency is equal to that of hovering frequency, ***224.5rad/seconds***. This is done by taking each gain as one valued. The open loop response gives the following.

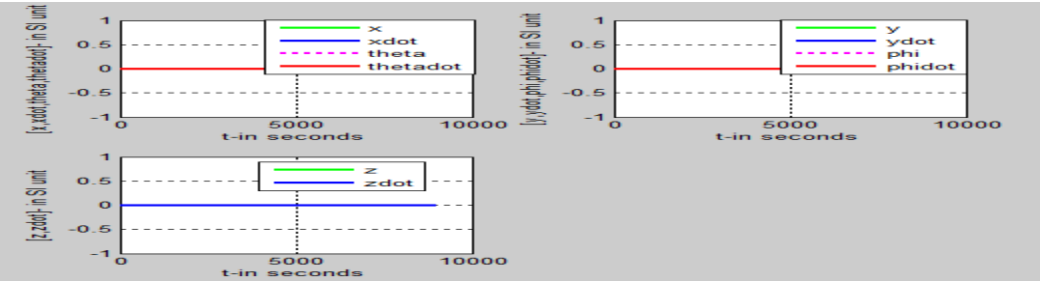

Figure 4_ 5: open loop hovering position response

b) The second one is the state of climbing. In this case, the rotors frequencies are greater than that of ***224.5rad/seconds***. In another words, the trust increases due to the increase of the rotors frequency by ***the same amount, may be 112.25 rad/seconds as below***. Here is the open loop response.

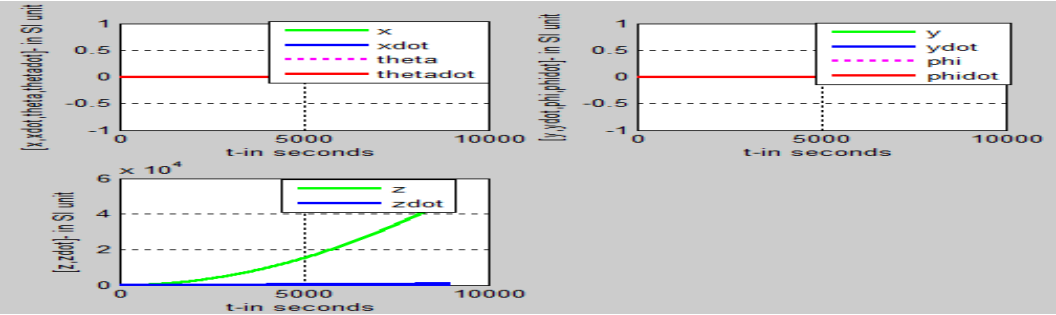

Figure 4_ 6: open loop climbing response

c) The third open loop response is the case where the UAV quad rotor is descending. In this case, the frequencies of each rotors are less than that of hovering frequencies. This may be done by decreasing the hovering frequency by ***112.25 rad/seconds as below***. The following is open loop response.

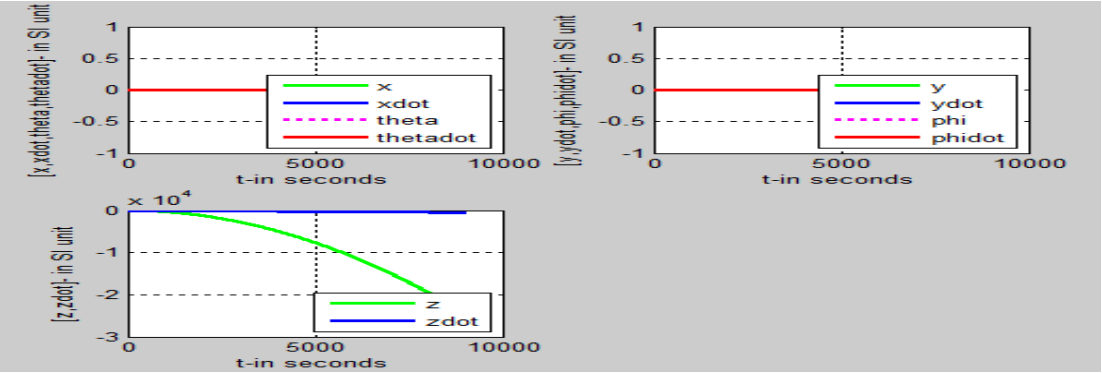

Figure 4_ 7: open loop descending response

d) Roll change can be done by taking ***krg*** is as positive i.e. ***(224.5+112.25)rad/seconds*** and ***kl*** as negative i.e. ***(224.5-112.25)rad/seconds*** or vice versa by keeping the others two zero. If the ***krg*** increases from ***224.5 rad/seconds and kl decreases from 224.5 rad/seconds***, the quadrotor moves along the y-axis or to the right position. Otherwise, it moves to the negative y-axis if the change is negative.

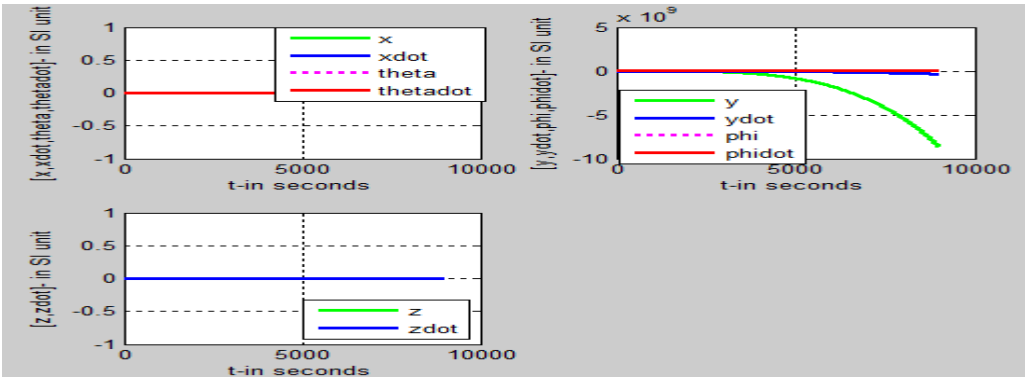

Figure 4_ 8: roll input control manually

e) Pitch change is done by varying the kf is taken as small positive or krr as small negative or vice versa by keeping the others zero. If kf increase from 224.5 ***rad/seconds*** and krr decreases from 224.5 ***rad/seconds***, the quadrotor moves along the y-axis or to the right position. Otherwise, it moves to the negative y-axis if the change is negative.

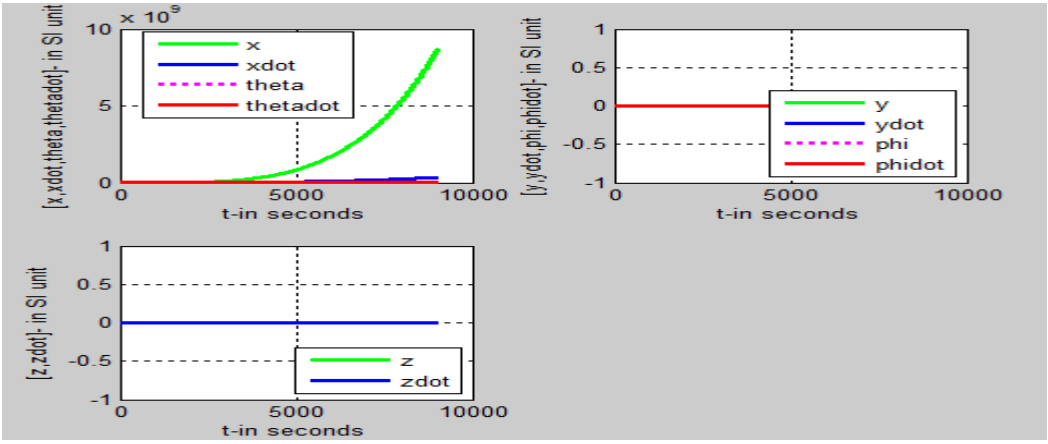

Figure 4_ 9: pitch input control

B) Appling the LQG controller

4.3.1. Hovering state

a) By neglecting the effect of noise

The output of different model structures on this thesis are given as below with some of its highlights. The controller stabilizes all states at zero after some time. At hovering, all rotational speeds are equal since all the sates outputs are zero and the supply voltages to the quad rotor are also equal. Due to derivative action, the error spike at the discontinuity point of step input with a given step time is very high. To limit this, the saturation of the voltages are given. The rotational speed at this stage can be found using the formula given below.

i) VTOL

VTOL is the potential of the UAV quad rotor to take off and landing. In this case three process can be observed. These process are hovering, climbing and descending. On the figure below, the pilot needs the UAV quad rotor to take off vertically at 1 second to 40m and hovering for **29** seconds. At 30 second, the pilot order the UAV to descend. On this figure, it is observed that the output follows the input strictly after about **8** seconds. This is done by giving the command to z-axis as $z_c = 40 * u(t-1) - 40 * u(t-30)$. As seen from the figure, climbing is more difficult than retreating one for the Quad rotor. But, it is faster.

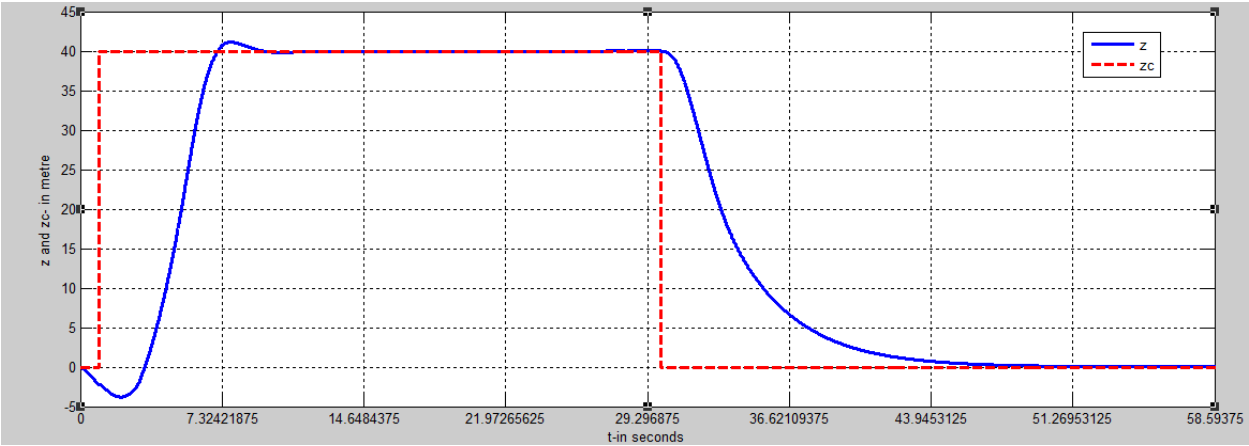

Figure 4_ 10: VTOL of UAV

ii) Omegas

As seen on figure 4_11 above, the UAV hovers from 11seconds to 30 seconds after climbing for 40 meters. This hovering condition is controlled by 6.38N trust force. The rotational speed for all rotors at this condition is 224.5 rad/sec. The control input U_1 changes by a fraction for a high altitude change from the fact that the control input is scaled by the trust constant **b**.

$$m\dot{w} = U_1 - mg = 0 \xrightarrow[implies]{} U_1 = 6.38\,N \;\; and\; the\; rest\; zero$$
$$from\; this: \qquad \textbf{(4.1)}$$
$$b * 4 * (om)^2 = mg \overset{implies}{\longleftrightarrow} om = 224.5 rad\,/sec$$

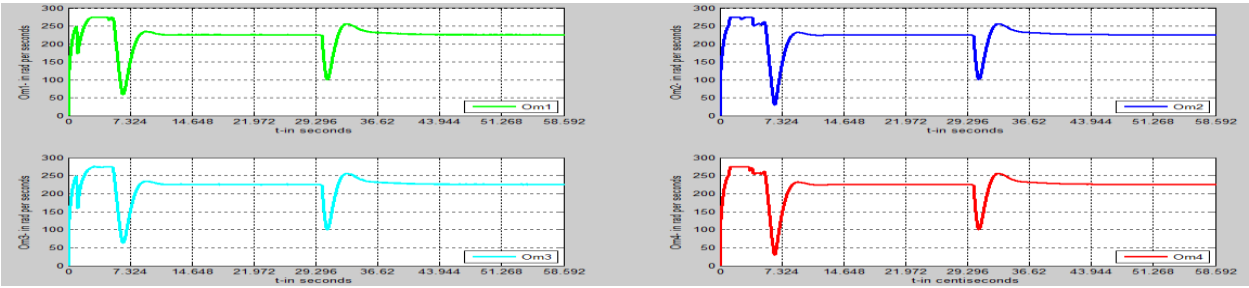

Figure 4_ 11: Omegas at hovering at 40m along z-axis

iii) Overall Omega

From these four rotor frequencies, the overall quad rotor frequency can be found by using the formula below. Since the position is at the hovering position, this frequency must be ***zero*** in order to stabilize the system at this position. However, due to inefficiency of estimation and choose of correct weighting matrices Q and R, there is still small amount of disturbance which inhibit Omr not to be zero or the oscillation exists.

$$Om_r = -Om1 + Om2 - Om3 + Om4 \qquad \textbf{(4.2)}$$

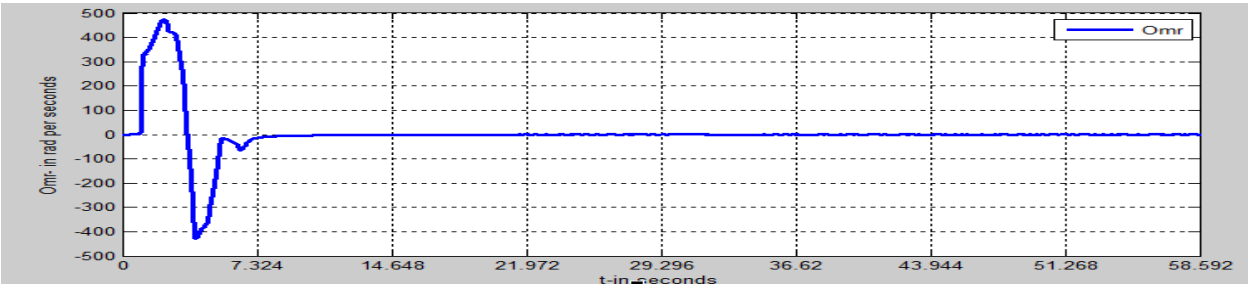

Figure 4_ 12: Omr at hovering at 40m along z-axis

iv) Trust Force at Hovering

The trust force at hovering is given by the formula in equation (**2.36)** after scaling by ***b.*** The trust force is given by $T_l = T_f = T_{rr} = T_{rg} = 0.25\sum_{i=1}^{4} T_i = m * \frac{g}{4} = 2.4525 * m = 1.594125\ N$

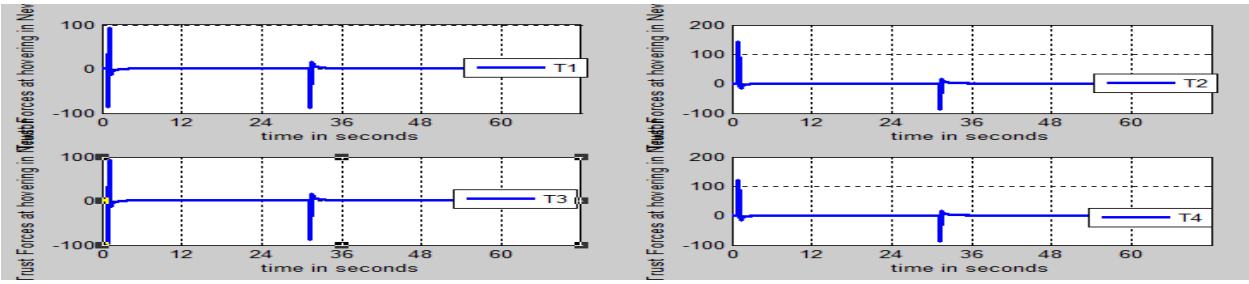

Figure 4_ 13: Trust Forces used to lift quad rotor to 40m along z-axis

v) Control inputs at hovering

The input controls at hovering are zero except U_1 which results : $U_1 = m * g = 6.38N$. Due to transition from 40m to 0m of the quad rotor at 30 seconds, there is small spike formed as shown below. This is formed due to the derivative part exist in the system design.

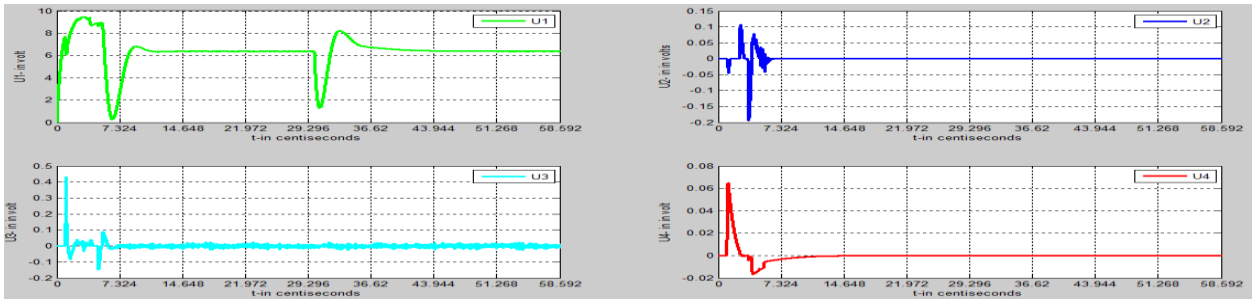

Figure 4_ 14: Input controls at hovering

vi) Position of quad rotor at VTOL from the ground

At the hovering, all states are istabilized at the hovering position or there will be no change from the hovering position as long as the controller arders the system not to follow some specifec path. In case of tracking, the quad rotor stays the place where it is ordered to be for this ***30 seconds span at (x, y, z1) position and then go to 80m and stay there for 27 seconds at (x, y, z2) as shown on the figure below***. The control principle employed in this sytem is error stabilizing. ***This hovering***

position is the position of Quad rotor from one second to 30 seconds and from 30 seconds to 55 seconds which is in reality imposible at 30 seconds. The quad rotor is intended to be landed at 30 seconds.

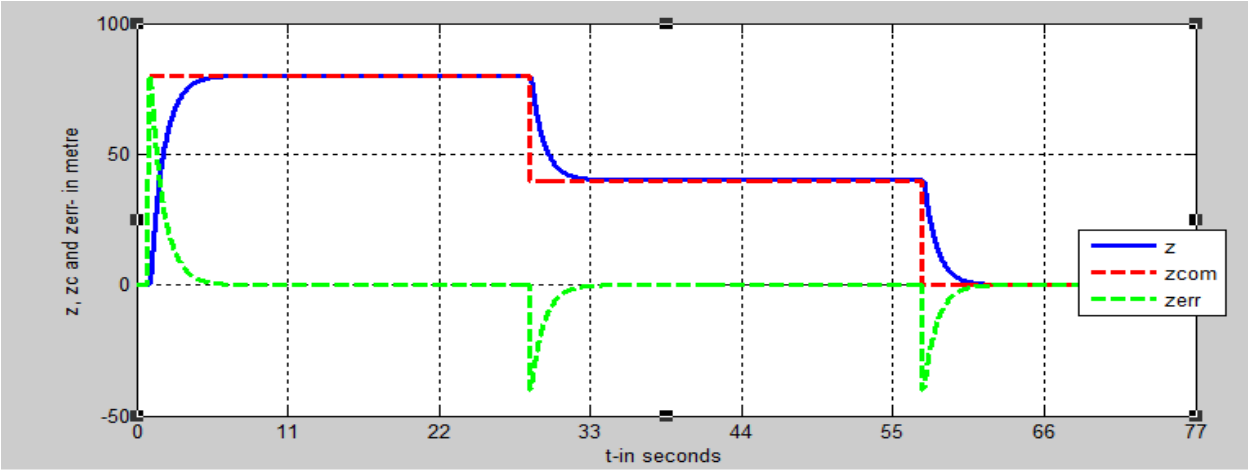

Figure 4_ 15: x_out, y_command and z_out

vii) The eight states at hovering

Except the four control variables, all the states are exactly stabilized as zero value. The two control part employed are the Kalman filter and LQR controlkers.

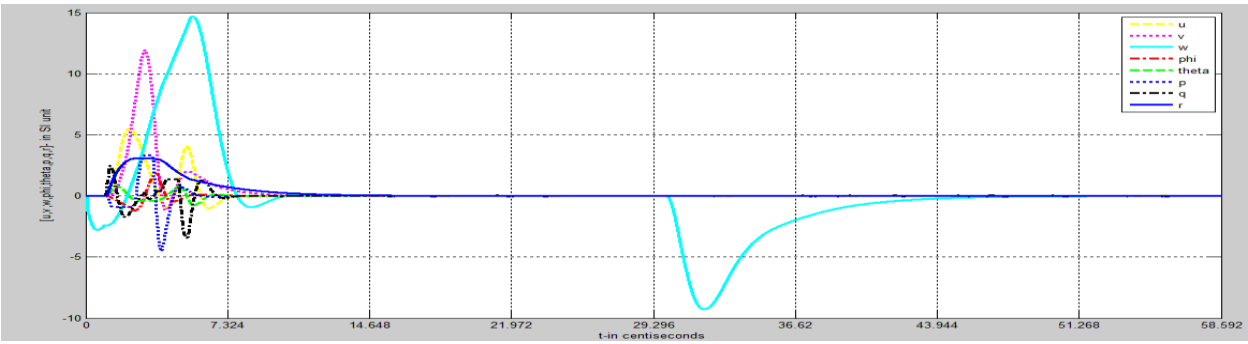

Figure 4_ 16: Output errors: u; v; w; phi; theta; p; q; r

From the state estimators, the errors from both state and out put must be approximated to zero in order to have a good out put. Moreover, the LQR out put is stabilized at the hovering. Becaues the C-matrix is identity matrix, output errors and state outputs are the same. The spike formed at 30 seconds is the output vertical speed due to derivative action from z axis posinon instanteneous change.

b) By introducing the effect of noise

1) Overall frequency

As much as possible, the system is tried to be stabilized for the given noise power and frequencies. The accelerometer and gyroscopes bias are assumed to be zero and only the noise introduced is considered. The introduced noise in each of them is taken to be a white noise of noise power about ***0.2w*** and its frequency about ***23341*** rad per second. As it can be seen from the result, the quad rotor is disturbed a little from the operating point. This is of course a minimum effect due to the problem of completely erasing the noise effect. The second spike is formed due to the transition from 40m to 0m from z axis position. ***The spike starts exactly at 30 seconds and stabilized fast.***

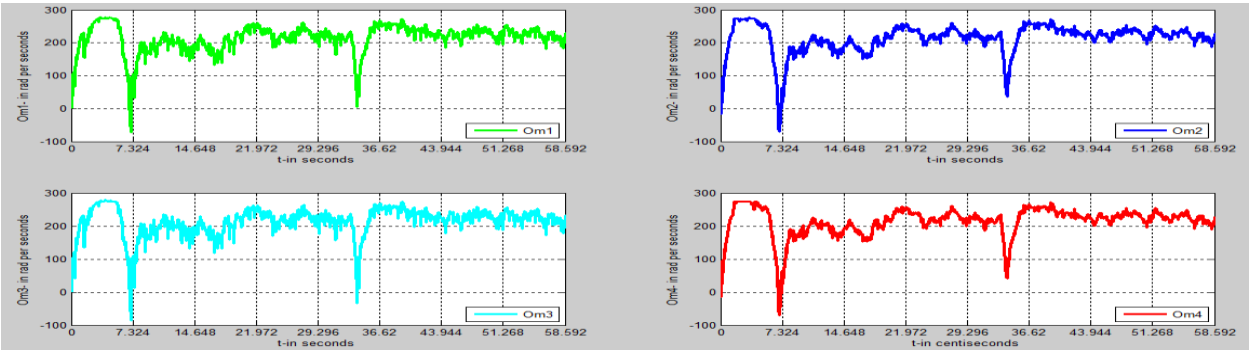

Figure 4_ 17: Rotors frequency under the existence of noise

2) Hovering position

With noise introduced in the system there is still desirable output. The noise is assumed to be of moderate power like 0.1. The following figure shows how accurate the VTOL is given that the command from z-axis as $z_c = 60 * u(t-1) - 60 * u(t-40) - 60 * u(t-60)$. This means the pilot needs the UAV quad rotor to take off vertically at 1 second to 60m and moves down to 30m at **35** seconds and then going down at 60 seconds. But, there is a little oscillation at the position as shown below on figure 4_20.

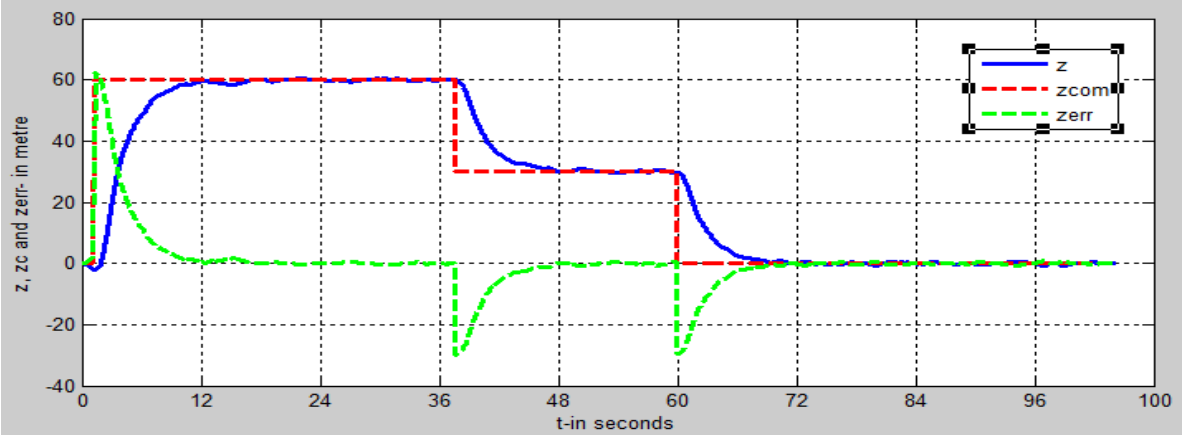

Figure 4_ 18: hovering Position with noise Introduced

4.3.2. Cruising mode

A) By neglecting the effect of noise

4.3.2.1. Altitude control and up and down translational velocity and acceleration.

The altitude control is done by controlling the Z-axis position and its corresponding translational velocity and acceleration. The control of the altitude is done by varying the control input U_1 by some value. From the derived formula of the system in this thesis, the body velocity and acceleration are given as:

$$\ddot{z} + g = \dot{w} + g = \frac{1}{m} U_1 \; and \; \dot{z} = w = -\int_0^t g + \int_0^t U_1(\tau) d\tau \qquad \textbf{(4.3)}$$

For the z-axis command, its velocity $w = \frac{d}{dt} z$ which means for a constant altitude or at a hovering position, the velocity $w = 0$. Otherwise, it changes accordingly. For example for the step input of $z_{command} = 40 * u(t-1) - 40 * u(t-50)$ given along z-axis, the speed of the vertical axis $w = 0 \; rad \; per \; seconds \;\; except \; at \; 50 \; seconds$ and its corresponding acceleration is: $\dot{w} = \Delta w = 0$. Down movement is analogous to up movement which is determined by LQG controller.

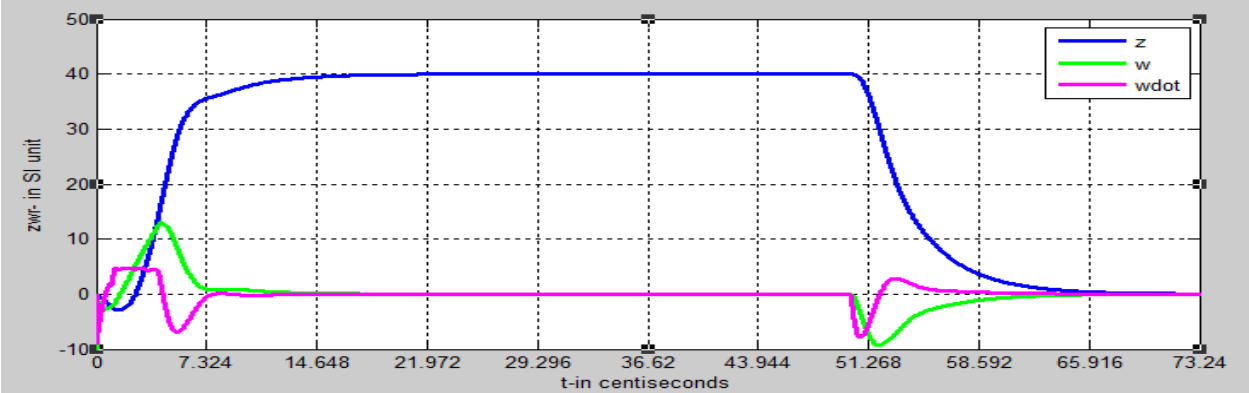

Figure 4_ 19: Altitude control

4.3.2.2. Pitch control and its sideways movements

Back movement is the effect of translation along the negative x-axis where Forth movement is the motion along the positive x-axis. These two effects can be controlled by the control of roll control previously shown on the model. The control of the roll is done by varying the control input U_2 by some value.

$$\ddot{x} = g * \theta \xrightarrow{implies} x^{(4)} = g * \ddot{\theta} \qquad \textbf{(4.4)}$$

Since the control input is kept as small value, the roll angle is also kept at small value. Due to this effect only a little roll drift is allowable. The control input U_2 is stabilized to the hovering position which is numerically about zero value.

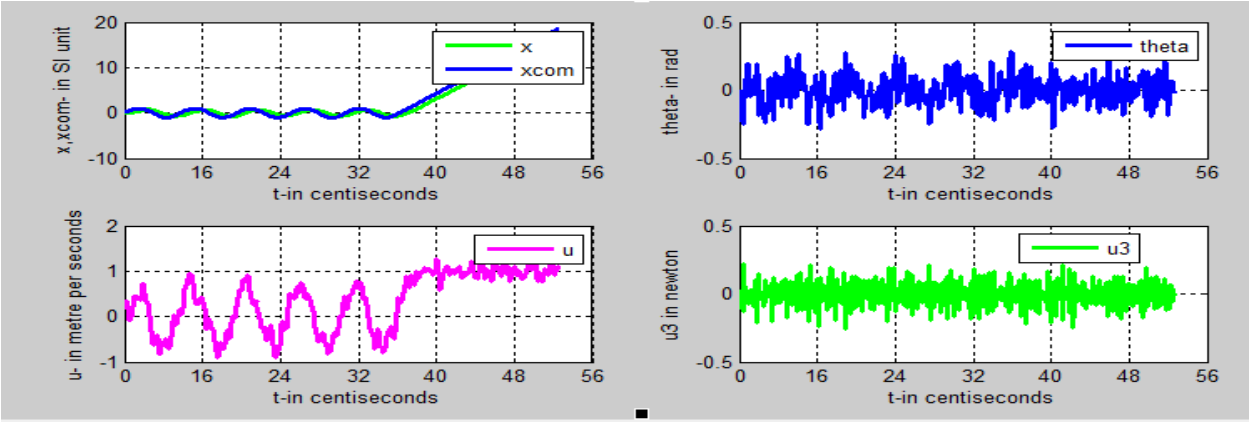

Figure 4_ 20: Roll control and it's forth and back word control

4.3.2.3. Roll control and its forth and back movements control

The pitch control is done by varying the rotational speeds of sideways by some amount which must not to be a high value in order to prevent high speed change that can disturb the quad rotor. The control of the roll is done by varying the control input U_3 by some value. The y-axis body velocity and the pitch control are related as follows after small angle approximation.

$$\ddot{y} = -g * \vartheta \xrightarrow{implies} y^{(4)} = -g * \ddot{\vartheta} \tag{4.5}$$

For small angle approximation, the output for those angles are stabilized at the hovering position and due to this effect, the U2 is zero after some time. The phi angle is stabilized after 3 seconds with a constant oscillation of sinusoidal nature from the command. This is due to the fact that small angle approximation is used while modeling.

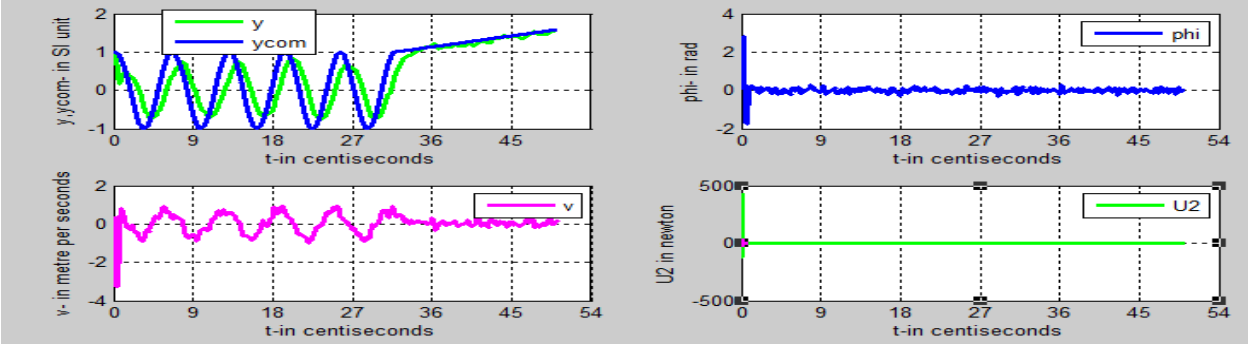

Figure 4_ 21: y-axis control

4.3.2.4. Psi control and its movement around the yaw axis

The psi angle movement is controlled by the fourth input parameter. This input can be commanded by the yaw angle and made to rotate around the desired axis by the set point already specified.

4.3.3. Trajectory following or path tracking

For the tracking purpose, the command is introduced in the Riccati equation and what is to be controlled is not the states as above. In this case, the error due to output and command or tracker is minimized as much as possible. The following formulas are used for the control of the path following. The detail is shown on the Appendix

$$J_{minimized} = 0.5\int(e^T(\tau)Qe(\tau) + U^T(\tau)RU(\tau))d\tau$$
$$e(t) = r(t) - y(t)$$
$$where:$$
$$e(t)\ is\ the\ error\ introduced;$$

(4.6)

The path following behavior of the quad rotor both for noisy and no noise both for declared command from the workspace and for step input is shown as follows. The declared path is written on the Mat lab editor window and passed to the Simulink block with the same name. The command plot and animation for ***spiral*** both for output and command is observed for the comparison purpose. $r(t)$ is the command. The methods employed for this purpose may be of the form:

a) LQT (Linear Quadratic Tracker)

This type of controller is used for time variant system or digital system **[24]**. The system used in this thesis is LTI system. This method is given on appendix both for derivation and coding.

$$g(t) = (pBR^{-1}B^T - A^T)^{-1}(C^TQr(t) - C^TV - pQ_nw) \quad \textbf{(4.7)}$$

b) LQI (Linear Quadratic Integrator)

This type of controller introduces the integral action for ***each state errors separately***. It brings the system output to the desired state as fast as possible. Because it doesn't concern about the system matrix, this is the simple task to be done.

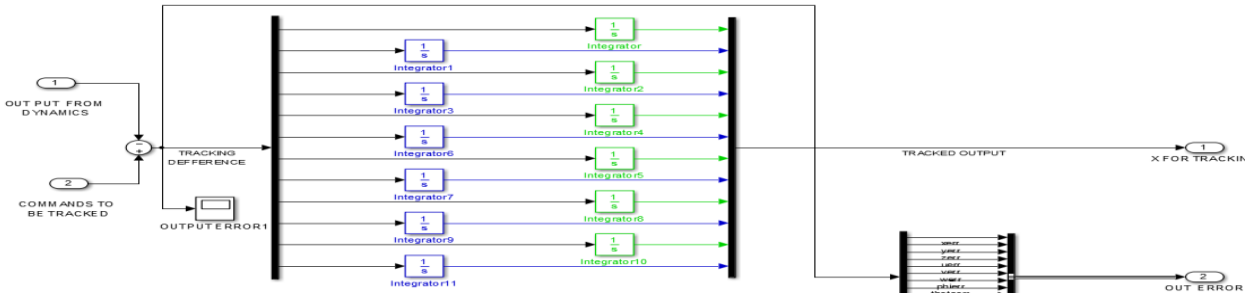

Figure 4_ 22: Integral in LQI tracking

Like it said before under LQR controller part, the efficiency of this task is done due to choose of weighting matrices $R\ and\ Q$. This can be achieved through trial and error.

$$x_i = \int_0^t e(\tau)\,d\tau = \int_0^t (r(\tau) - y(\tau))d\tau \quad \textbf{(4.8)}$$

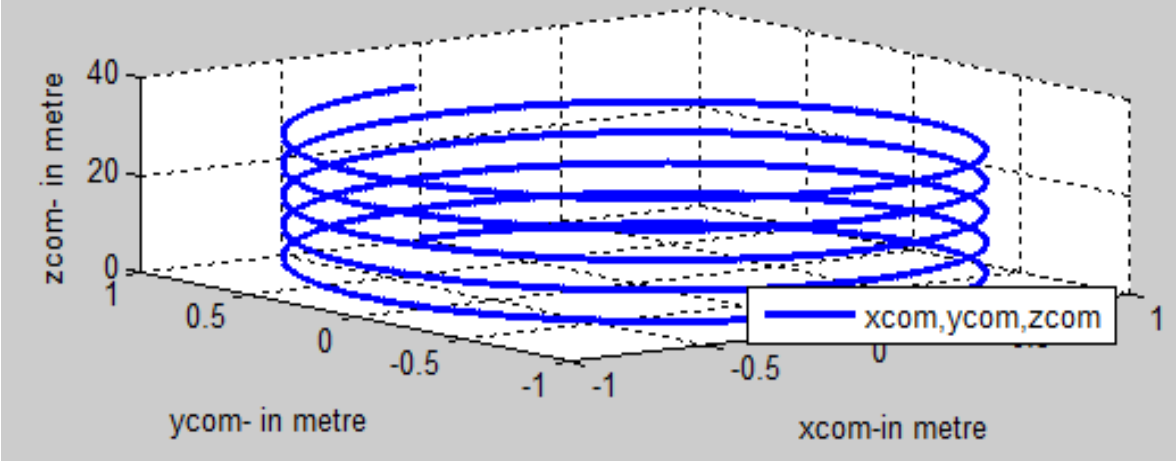

Figure 4_ 23: 3D Plot for spiral command

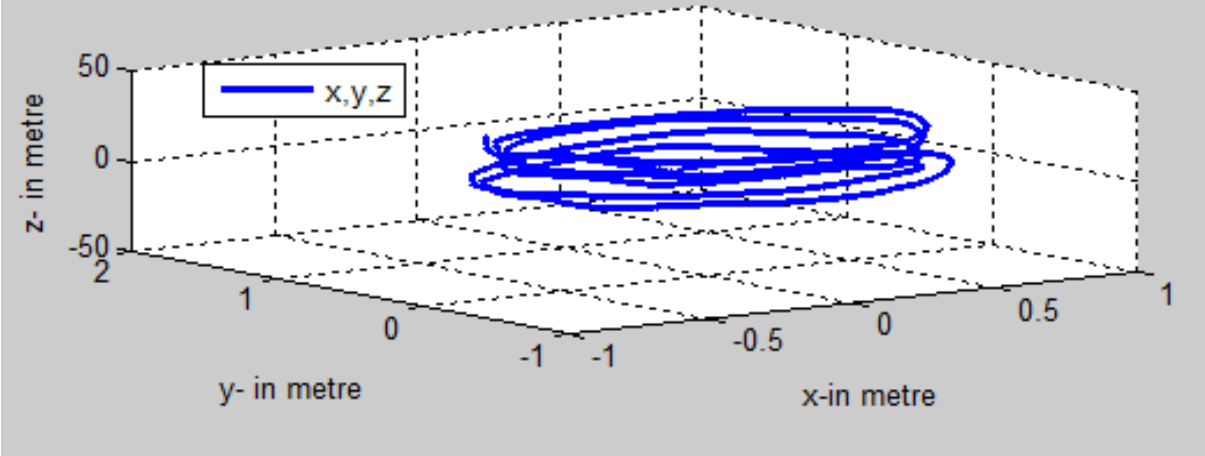

Figure 4_ 24: 3D animation output for path tracking with high power noise

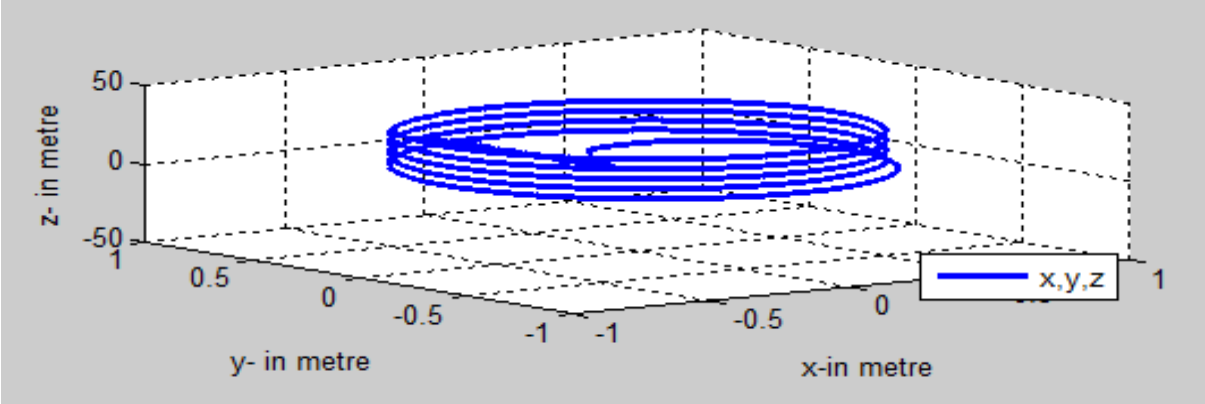

Figure 4_ 25: 3D animation output for path tracking with no noise introduce

Summary of simulation analysis and results:

Serial Number	***Control Action***	***Open loop Response: Commands are frequencies of the rotors***	***LQG controller Response: Commands are x_{com}, y_{com}, z_{com} and psi $_{com}$***
1.	***Hovering Control***	a) If $Om_{l, rg, f, rr} = Om_{hovering}$, Quad rotor hovers b) If $Om_{l, rg, f, rr} > Om_{hovering}$, Quad rotor climbs c) If $Om_{l, rg, f, rr} < Om_{hovering}$, Quad rotor descends	a) If Zcom constant and others zero, each rotors frequencies constant and U1 constant. Therefore, quad rotor hovers b) If Zcom increases and others zero, each rotors frequencies increase and U1 increase. Therefore, quad rotor climbs c) If Zcom decreases and others zero, each rotors frequencies decreases and U1 decreases. Therefore, quad rotor descends
2.	***Cruising Control or Tracking***	***a) Pitch control:*** Except U3 all zero. This happens when Om_{l} increases and Om_{rg} decreases or vice versa. ***b) Roll control:*** Except U2 all zero. This happens when Om_{f} increases and Om_{rr} decreases or vice versa. ***c) Yaw control:*** Except U4 all zero. This happens when Om_{l} and Om_{rg} increases and Om_{f} and Om_{rr} decreases or vice versa.	***a) Pitch control:*** Only y_{com} given and hence U3 is to be stabilized which in turn stabilizes angle phi. ***b) Roll angle:*** Only x_{com} given and hence U2 is to be stabilized which in turn stabilizes angle theta. ***c) Yaw control:*** Only psi $_{com}$ given and hence U4 is to be stabilized which in turn stabilizes angle psi.

Table 5_ 1: simulation analysis Summary

CHAPTER FIVE

5. CONCLUSIONS AND RECOMMENDATIONS AND SUGGESTION FOR FUTURE WORKS

5.1. Introduction

In this chapter three things are discussed. These are conclusion, recommendations and suggestion for future works. Conclusion is the short and precise discussion of the thesis work. Recommendation is the alternative way of the work done in the thesis. Suggestion for future works is about the works that are not included in the thesis work.

5.2. Conclusions

In this thesis, the control of Vertical take Off Unmanned Arial Vehicle was presented. The UAV quad rotor model was derived and linearized using first order Tylor series. Model verification was done to verify the correctness of the model through open loop response. From this open loop response, it was observed that the hovering position of the UAV quad rotor was stabilized depending on the four control inputs. The z-axis position of quad rotor was controlled by the first control input which depends on the rotors frequency values. To move the quad rotor up, the overall trust must overcome the weight of the quad rotor. To balance the quad rotor at the position, the net z-axis force must be balanced. Otherwise, the quad rotors descends.

By using the LQG controller, both tracking and hovering position were regulated to the desired position. These positions were checked both for the case where there is no disturbance and there is the influence of disturbance. The Kalman filter was used to minimize the effect of this disturbance introduced both in the plant and sensors. The followings were the main results obtained.

- The quad rotor was stabilized at hovering position at 8 seconds in the case disturbance is disregarded and where as it is about 9 seconds in the case disturbance is considered both in plant and sensors.

- The tracking was achieved in about 6seconds both for x-axis and y-axis but took about 9 seconds for the z-axis in the case disturbance is disregarded. In the case of disturbance, it was observed that the time taken was about 7 seconds for x-axis and y-axis where it was about 10 seconds for the z-axis.
- The control input U1 varied slightly for a large number of position change which shows the power consumed is minimum which depends on the mass of the quad rotor and arm length of the propellers.
- The other three control inputs were obtained zero due to the fact that the small angle approximation was used. For any constant commands, the second derivative gives zero. Therefore, the three control input are zero.
- Angles are stabilized fast depending on the commands given to the three axis distances

5.3.Recommendations

In the thesis work, the two weighting matrices were randomly selected. Due to this, the inputs and states were not fast to be converged as seen from simulation results. Hence it is a good practice if system optimization tool is used to search for the optimum value of LQR gain. The second one is that while Kalman filter was calculated using loop recovery technique in the model, the covariance of the disturbance was taken near to one. As far as this covariance drops, the estimate errors becomes large. Therefore, other loop recovery method that does not depend on the covariance value shall be used.

5.4. Suggestion for future works

As the model used is of linear type, the full model is not used. This means, the effects of the hub, aerodynamics and friction were neglected. The drag coefficient (d) and the lift coefficient (b) which are normally varying in reality were taken as if constants. However, the real system may not be as simple as the system designed at a simulation level. Due to this, it is a good practice if the model is somewhat advanced. The other thing to be included is that the control translator is not designed except some of them were observed under open loop system design. It is good if this control translators will be used so that there will be a clear communication between ground operator and quad rotor.

REFERENCES

[1]. Design, Construction and Control of a Large Quad rotor Micro Air Vehicle; Paul Edward Ian Pounds; Thesis, September 2007

[2]. Modelling, Identification and Control of a Quad rotor Helicopter; Tommaso Bresciani; Thesis, October 2008

[3]. Development and Implementation of a Control System for a quad rotor UAV; By Yiting Wu; Thesis, March 2009

[4]. Design and Control of a Miniature Quad rotor; Samir Boubdallah and Roland Siegwart; Autonomous Systems Lab, ETH Zurich, Switzerland, Thesis.

[5]. SIMULATION AND CONTROL OF A QUADROTOR UNMANNED AERIAL VEHICLE Michael David Schmidt; University of Kentucky, mdschm2@uky .edu; Thesis, 2011

[6]. Modelling of the Flight Dynamics of a Quad rotor Helicopter; VICENTE MARTÍNEZ, Thesis, September 2007.

[7]. Quad rotor Dynamics and Control Randal W. Beard Brigham Young University, Thesis, October 3, 2008

[8]. Analysis and model based control of a quad rotor helicopter; Ilona Sonnevend; Budapest, Thesis, 2010.

[9]. Modelling and simulation of a quad-rotor helicopter a. bousbaine*, m. h. wu*, g. t.poyi*, Gwangtim Timothy, article, 14 June 2016 15:01:18.

[10]. Design, implementation and flight test of indoor navigation and control system for a quad rotor UAV; Menno Wierema B.Sc., Thesis, 11 December 2008.

[11]. MODELLING AND LINEAR CONTROL OF A QUADROTOR; C BALLAS; 2006 2007

[12]. www.linearization techniques

[13]. www.LQR,LQG and H_infinity controllers

[14]. Modeling, Identication and Control of a Quad rotor Aircraft Marcelo De Lellis Costa de Oliveira, Thesis, 20 may 2011.

[15]. www.Optimal Linear Quadratic Gaussian (LQG) Control

[16]. Conventional Quad rotor control and simulation Sérgio Eduardo Aurélio Pereira da Costa Instituto Superior Técnico Portugal, Thesis, October 2008

[17]. http://www. State estimation with observers

[18]. OPTIMAL CONTROL SYSTEMS; Desineni Subbaram; Naidu Idaho State University; Pocatello. Idaho. USA, Book.

[19]. Quad rotor prototype; Jorge Miguel Brito Dominguez; Instituto Superior Técnico Portugal Thesis, October 2009

[20]. REPORT OF THE COURSE "PROGETTO DI MICROELETTRONICA," English reduced version; D. Comotti, M. Ermidoro, Thesis.

[21]. Design Optimization of a Quad-Rotor Capable of Autonomous Flight; Mark Dupuis_mdupuis@wpi.edu; Jonathan Gibbons_jonored@WPI.EDU; Maximillian Hobson-Dupont ltdan@wpi.edu; Alex Knight koohash@wpi.edu ;Artem Lepilov: artem@wpi.edu ;Michael Monfreda monfreda@wpi.edu ;George Mungai gmungai@wpi.edu; Project.

[22]. Flight Control System of an Experimental Unmanned Quad-Rotor Helicopter; Antal Turóczi, a.turoczi@t-online.hu, Thesis, June 25, 2009.

[23]. Brushless_motor.png, control.bmp, filter.png from quadrotorismV2.link, by Balaji Shankar Kumar, Thesis, 2013.

[24]. Trajectory Tracking of UAV via attitude and position control, by EMRE CAN SU˙ IÇMEZ, Thesis, JULY 2014

APPENDICES

Appendix A: Linearization of UAV quad rotor Nonlinear Model

As explained under the topic of Linearization near the equilibrium points, the linearization process is shown below. The linearization of earth frame velocity in terms of body frame is as follow. Before this procedure, the rotational matrix R_v^b must be approximated using ***small angle approximation*** for: the sake of simplicity and the fact that the product of two or more small numbers are zero.

$$R_b^v = \begin{bmatrix} 1 & -\Delta\varphi & \Delta\theta \\ \Delta\varphi & 1 & -\Delta\vartheta \\ -\Delta\theta & \Delta\vartheta & 1 \end{bmatrix}$$

Using this approximated Rotational matrix and transfer matrix:

a) $\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 1 & -\Delta\varphi & \Delta\theta \\ \Delta\varphi & 1 & -\Delta\vartheta \\ -\Delta\theta & \Delta\vartheta & 1 \end{bmatrix} * \begin{bmatrix} u \\ v \\ w \end{bmatrix} \overset{implies}{\longleftrightarrow} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} u - \Delta\varphi * v + \Delta\theta * w \\ \Delta\varphi * u + v - \Delta\vartheta * w \\ -\Delta\theta * u + \Delta\vartheta * v + w \end{bmatrix} where: \Delta\theta = \theta - \theta_0 \cong \theta$

b) $\begin{bmatrix} \dot{\vartheta} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta\theta \\ 0 & 1 & -\Delta\vartheta \\ 0 & \Delta\vartheta & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \overset{implies}{\longleftrightarrow} \begin{bmatrix} \dot{\vartheta} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} p + \Delta\theta * r \\ q - \Delta\vartheta * r \\ \Delta\vartheta * q + r \end{bmatrix}$

c) $\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} v * r - w * q - g * \sin(\theta) \\ u * r - w * p + g * \sin(\vartheta) * \cos(\theta) \\ u * q - v * p - g * \cos(\vartheta) * \cos(\theta) + \frac{1}{m} U_1 \end{bmatrix}$

d) $\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{U_2}{J_{xx}} + \left(\frac{J_{yy} - J_{zz}}{J_{xx}}\right) q * r_0 \\ \frac{U_3}{J_{yy}} + \left(\frac{J_{zz} - J_{xx}}{J_{yy}}\right) p * r_0 \\ \frac{U_4}{J_{zz}} + \left(\frac{J_{xx} - J_{yy}}{J_{zz}}\right) q * p_0 \end{bmatrix}$ *gyro and inertial countr torque are neglected*

Using all these simplifications and logics, the final answers for the linear expression of earth frame velocities in terms of body frame velocities in matrix form is:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = \begin{bmatrix} u + \varphi * v + \theta * w \\ \varphi * u + v - \vartheta * w \\ -\theta * u + \vartheta * v + w \end{bmatrix}$$

$$\begin{bmatrix}\dot{\vartheta}\\ \dot{\theta}\\ \dot{\varphi}\end{bmatrix} = \begin{bmatrix} p + \Delta\theta * r \\ q - \Delta\vartheta * r \\ \Delta\vartheta * q + r \end{bmatrix}$$

Using first order Taylor series at $\{\theta_0; \vartheta_0; \varphi_0; u_0; v_0; w_0\}$ and

$$f(u, v, w, \vartheta, \theta, \varphi) = f(u_0, v_0,\ w_0, \vartheta_0, \theta_0, \varphi_0) + \frac{\partial f}{\partial u}(u_0, v_0,\ w_0, \vartheta_0, \theta_0, \varphi_0)\Delta u +$$

$$\frac{\partial f}{\partial v}(u_0, v_0,\ w_0, \vartheta_0, \theta_0, \varphi_0)\Delta v + \frac{\partial f}{\partial w}(u_0, v_0,\ w_0, \vartheta_0, \theta_0, \varphi_0)\Delta w + \frac{\partial f}{\partial \vartheta}(u_0, v_0,\ w_0, \vartheta_0, \theta_0, \varphi_0)\Delta\vartheta +$$

$$\frac{\partial f}{\partial \theta}(u_0, v_0,\ w_0, \vartheta_0, \theta_0, \varphi_0)\big((\theta - \theta_0) = \Delta\theta\big) + \frac{\partial f}{\partial \varphi}(u_0, v_0,\ w_0, \vartheta_0, \theta_0, \varphi_0)(\varphi - \varphi_0)$$

And taking $f(u_0, v_0,\ w_0, \vartheta_0, \theta_0, \varphi_0) = 0$, it yields:

a) $$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = \begin{bmatrix} \Delta u + \Delta\varphi * v_0 + \Delta v * \varphi_0 + \Delta\theta * w_0 + \Delta w * \theta_0 \\ \Delta\varphi * u_0 + \Delta u * \varphi_0 + \Delta v - \Delta\vartheta * w_0 - \Delta w * \vartheta_0 \\ -\Delta\theta * u_0 - \Delta u * \theta_0 + \Delta\vartheta * v_0 + \Delta v * \vartheta_0 + \Delta w \end{bmatrix}$$

b) $$\begin{bmatrix}\dot{\vartheta}\\ \dot{\theta}\\ \dot{\varphi}\end{bmatrix} = \begin{bmatrix} \Delta p + \Delta\theta * r_0 + \Delta r * \theta_0 \\ \Delta q - \Delta\vartheta * r_0 - \Delta r * \vartheta_0 \\ \Delta\vartheta * q_0 + \Delta q * \vartheta_0 + \Delta r \end{bmatrix}$$

c) $$\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{bmatrix} =$$

$$\begin{bmatrix} \Delta v * r_0 + \Delta r * v_0 - \Delta q * w_0 + \Delta w * q_0 - g * \cos(\theta_0)\Delta\theta \\ \Delta w * p_0 + \Delta p * w_0 - \Delta r * u_0 + \Delta r * u_0 + g * \mathrm{c}\,os(\vartheta_0) * \cos(\theta_0)\Delta\vartheta - g * \mathrm{s}\,in(\vartheta_0) * \sin(\theta_0)\Delta\theta \\ \Delta u * q_0 + \Delta q * u_0 - \Delta p * v_0 + \Delta v * p_0 - g * \mathrm{s}\,in(\vartheta_0) * \cos(\theta_0)\,\Delta\vartheta - g * \mathrm{c}\,os(\vartheta_0) * \sin(\theta_0)\,\Delta\vartheta + \frac{1}{m}U_1 \end{bmatrix}$$

d) $$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}\frac{U_2}{J_{xx}} + \left(\frac{J_{yy}-J_{zz}}{J_{xx}}\right)(q_0 * \Delta r + r_0 * \Delta q)\\ \frac{U_3}{J_{yy}} + \left(\frac{J_{zz}-J_{xx}}{J_{yy}}\right)(p_0 * \Delta r + r_0 * \Delta p)\\ \frac{U_4}{J_{zz}} + \left(\frac{J_{xx}-J_{yy}}{J_{zz}}\right)(q_0 * \Delta p + p_0 * \Delta q)\end{bmatrix}$$

Appendix C: tracking modelling for LQT

In this thesis the tracking process introduced the effect of noise. However, LQG tracking has a problem of noise and most of the time ***LQT*** controller is used. This control used for time variant system. Nevertheless, it was tried to minimize the effect of noise as much as possible. The following derivation follows the principle of Hamiltonian formula. All the steps are taken from '***optimal control systems'*** by ***Richard C. Dorf*** except that the effect of noise was not considered.

The state and output equation under the influence of noise is given as follows for which the optimal control is also written following it. In the followings, the ***** symbol is simply used to show optimal state of the symbolized vector. In below, $r(t)$ is the command or path needed to be followed and Q_n is the noise covariance matrix where V is the measurement noise.

$$\dot{x}(t) = Ax(t) + BU(t) * Q_n w$$
$$y(t) = Cx(t) + V$$
$$e(t) = r(t) - y(t)$$
$$J = 0.5\int (e^T(\tau)Qe(\tau) + U^T(\tau)RU(\tau))d\tau$$

Step one: Hamiltonian formulation

$$H\big(x(t), U(t), \lambda(t)\big) = 0.5(r(t) - Cx(t) + V)^T Q(r(t) - Cx(t) + V) + 0.5U^T(t)RU(t) + \lambda^T(t)(x(t) + BU(t) * Q_n w)$$

Step two: Open loop optimal control equation

$$0 = \frac{\partial H}{\partial U} = RU(t) + B^T\lambda(t) \overset{implies}{\longleftrightarrow} U^*(t) = -R^{-1}B^T\lambda^*(t)$$

Step three: state and cost ate system Interms of Hamiltonian from step one

$$\dot{x}^*(t) = \frac{\partial H}{\partial \lambda} = Ax(t) + BU(t) * Q_n w \xrightarrow{substituting\ for\ U\ from\ step\ 2} \dot{x}^*(t) = Ax(t) - BR^{-1}B^T\lambda^*(t) + Q_n w$$

$$and$$

$$\dot{\lambda}^*(t) = -\frac{\partial H}{\partial x} = -C^TQCx^*(t) + C^TQr(t) - C^TV - A^T\lambda^*$$

Step four: Riccati and vector equation

In this derivation, the goal is to find the vector that can enable the system to follow the specified path. P is the solution from algebraic Riccati equation.

$$\lambda^*(t) = px^*(t) - g(t) \xrightarrow[implies]{} \dot{\lambda}^*(t) = p\dot{x}^*(t) - \dot{g}(t) \dots\dots substituting\ for\ \dot{\lambda}^*(t)\ and\ \dot{x}^*(t)$$

$$(-C^TQC - pA - A^Tp + pBR^{-1}B^Tp)x^*(t) - (pBR^{-1}B^T - A^T)g(t) + C^TQr(t) - C^TV - pQ_nw = \dot{g}(t)$$

$$if\ the\ commands\ are\ step\ inputs,\ \dot{g}(t) = 0.then,\quad it\ can\ be\ said\ that\ the\ followings\ are\ true$$

$$-C^TQC - pA - A^Tp + pBR^{-1}B^Tp = 0 \dots\dots\dots..ARE$$

$$g(t) = (pBR^{-1}B^T - A^T)^{-1}(C^TQr(t) - C^TV - pQ_nw) \dots\dots\dots.vector\ for\ path\ following$$

Step five: Solving for the optimal control in terms of tracker

Finally, the control input can be found as below for the desired trajectory tracking.

$$u^*(t) = K_{are}x^*(t) + inv(R) * B^Tg(t)$$

Appendix B: MATLAB code

1) ***Mat lab code for LQG design:***

```
close all
clear all
clc
%xdot=Amoving(12,12)*x(12,1)+Bmoving(12,4)*U+Gmoving(12,3)*w(3,1)
%y=Cmoving(12,12)*x(12,1)+Dmoving(12,4)*U(4,1)
%Z(12,1)=Hmoving(12,12)*x(12,1)+Rn(12)*Vmoving(12,1)
```

```
mu1=1;
Jxx = 7.5*10^(-3);  % Quadrotor moment of inertia around X axis
Jyy = 7.5*10^(-3);  % Quadrotor moment of inertia around Y axis
Jzz = 1.3*10^(-2);  % Quadrotor moment of inertia around Z axis
Jr = 6.5*10^(-5);  % Total rotational moment of inertia around the propeller axis
b = 3.13*10^(-5);  % Thrust factor
d = 7.5*10^(-6);  % Drag factor
l = 0.23;  % Distance to the center of the Quadrotor
m = 0.65;
% Mass of the Quadrotor in Kg
g = 9.81;
% tolaration to change from Operating Points
ph0=0.04;
th0=0.05;
ps0=0.03;
u0=0.01;
v0=0.02;
w0=0.03;
p0=0.04;
q0=0.01;
r0=0.01;
%constants used in Matrix Amoving
k1=g*cos(ph0)*cos(th0);
k2=-g*cos(ph0)*sin(th0);
k3=-g*cos(th0)*sin(ph0);
k4=-g*sin(ph0)*sin(th0);
k5=((Jyy-Jzz)/Jxx)*q0;
k6=((Jyy-Jzz)/Jxx)*r0;
k7=((Jzz-Jxx)/Jyy)*p0;
k8=((Jzz-Jxx)/Jyy)*r0;
k9=((Jxx-Jyy)/Jzz)*p0;
```

```
k10=((Jxx-Jyy)/Jzz)*r0;
xd0=u0-ps0*v0-th0*w0;
yd0=ps0*u0+v0-ph0*w0;
zd0=-th0*u0+w0+ph0*v0;
ud0=v0*r0-w0*q0-g*sin(th0);
vd0=u0*r0-w0*p0+g*sin(ph0)*cos(th0);
wd0=u0*q0-v0*p0-g*cos(ph0)*cos(th0)+g;
pd0=k5*r0;
qd0=k6*p0;
rd0=k9*q0;
phd0=p0+th0*r0;
thd0=q0-p0*r0;
psd0=r0+ph0*q0;
% constant of opearting point from first order tylor series
Xd0=[xd0;yd0;zd0;ud0;vd0;wd0;phd0;thd0;psd0;pd0;qd0;rd0];
%decoupling inputs and angles
decopmat=[1 0 0 0;0 cos(ps0) sin(ps0)*cos(ph0)*Jxx/Jyy 0;
        0 -sin(ps0)*Jyy/Jxx cos(ph0)*cos(ps0) 0;
        0 -sin(ph0)*Jzz/Jxx 0 1];
    invdec=inv(decopmat);
    % linearized matrices
Amoving=[0 0 0 1 ph0 th0 0 w0 v0 0 0 0 ;
   0 0 0 ps0 1 -ph0 -w0 0 u0 0 0 0 ;
   0 0 0 -ph0 ph0 1 v0-u0 0 0 0 0 0 ;
   0 0 0 0 r0 -q0 0 g*cos(th0) 0 0 -w0 v0 ;
   0 0 0 -r0 0 p0 -k1 k2 0 w0 0 -u0 ;
   0 0 0 q0 -p0 0 k3 k4 0 -v0 u0 0 ;
   0 0 0 0 0 0 0 r0 0 1 0 th0 ;
   0 0 0 0 0 0 -r0 0 0 0 1 -ph0 ;
   0 0 0 0 0 0 q0 0 0 0 ph0 1 ;
   0 0 0 0 0 0 0 0 0 0 k6 k5 ;
```

```
   0 0 0 0 0 0 0 0 0 k8 0 k7 ;
   0 0 0 0 0 0 0 0 0 k10 k9 0 ];
Bmoving1= [0 0 0 0 ; 0 0 0 0 ; 0 0 0 0 ; 0 0 0 0 ;
   0 0 0 0 ; -1/m 0 0 0 ; 0 0 0 0 ; 0 0 0 0 ;
   0 0 0 0; 0 1/Jxx 0 0 ; 0 0 1/Jyy 0 ; 0 0 0 1/Jzz ];
% gravitational effect
constant=[0;0;0;0;0;g;0;0;0;0;0;0];
% measure and output matrices
Cmoving=eye(12,12);
Dmoving=zeros(12,4);
Gmoving=zeros(12,4);
Hmoving=zeros(12,4);
R=[1 0 0 0 ; 0 1 0 0 ; 0 0 1 0 ; 0 0 0 0.1];
B=Bmoving1/(R)*Bmoving1';
Q=eye(12,12);
Q(1,1)=10000;
Q(2,2)=10000;
Q(3,3)=25000;
Q(9,9)=10000;
%Kalman gain
p=are(Amoving,B,Q);
Kare=(R)\Bmoving1'*p;
Rn1=mu1^2.*eye (12, 12); %Quad rotor noise
Bw =eye (12, 12); %Bmatrix due to noise
Qn1 =mu1^2.*eye (12, 12) + Bmoving1*Bmoving1';
Cn = (Cmoving'/ (Rn1))*Cmoving;
Qn=Bw*Qn1*Bw';
s=are (Amoving, Cn, Qn);
%observer gain
Kest=s*Cmoving'/ (Rn1);
%%%%%%%%%%END%%%%%%%%%%%%
```

2) *Mat lab code for path generation*

```
t = [0: pi/50:10*pi]';
x = sin (t);
y = cos (t);
z=t;
Wave. time = t;
wave.signals.values = [x, y, z];
wave.signals.dimensions =3;
```

3) *Mat lab code for 3D pilot*

```
t = 0: pi/50:10*pi;
plot3 (sin (t), cos (t), t, 'b+-')
Xlabel ('sin (t)')
Ylabel ('cos (t)')
Zlabel ('t')
Grid on
Axis square
```

4) *Tracking code*

```
%quad rotor model m. file. This is the alternative way expressing the model of quad rotor
% for purpose of LQT tracking
function [xdr,ydr,zdr,xdd,ydd,zdd,psid,thetd,phid,pd,qd,rd] =
fcn(x,y,z,xd,yd,zd,psi,thet,phi,p,q,r,U1,U2,U3,U4)
Jxx = 7.5*10^(-3);  % Quadrotor moment of inertia around X axis
Jyy = 7.5*10^(-3);  % Quadrotor moment of inertia around Y axis
Jzz = 1.3*10^(-2);  % Quadrotor moment of inertia around Z axis
Jr = 6.5*10^(-5);  % Total rotational moment of inertia around the propeller axis
b = 3.13*10^(-5);  % Thrust factor
%d = 7.5*10^(-6);  % Drag factor
l= 0.23;  % Distance to the center of the Quadrotor
m = 0.65;% Mass of the Quadrotor in Kg
```

```
g = 9.81;   % Gravitational acceleration
%omega_r=0;
Kt=0.001;
xdr=xd;
ydr=yd;
zdr=zd;
%R_b^v = [cos(thet)*cos(psi)  sin(phi)*sin(thet)*cos(psi)-cos(phi)*sin(psi)
cos(phi)*sin(thet)*cos(psi)+sin(phi)*sin(psi);cos(thet)*sin(psi)
sin(phi)*sin(thet)*sin(psi)+cos(phi)*cos(psi) cos(phi)*sin(thet)*sin(psi)-sin(phi)*cos(psi);-
sin(thet) sin(phi)*cos(thet) cos(phi)*cos(thet)];
%acc=  -[0;0;g]+ R_b^v *[0;0;U1/m]-(Kt/m)*[xd;yd;zd];
xdd=(U1/m)*(cos(phi)*sin(thet)*cos(psi)+sin(phi)*sin(psi)-(Kt/m)*xd);
ydd=(U1/m)*(cos(phi)*sin(thet)*sin(psi)+sin(phi)*cos(psi))-(Kt/m)*yd;
zdd=-g+(U1/m)*cos(phi)*cos(thet)-(Kt/m)*zd;
LRI=[1 sin(phi)*tan(thet) cos(phi)*tan(thet);0 cos(phi) -sin(phi);0 sin(phi)/cos(thet)
cos(phi)/cos(thet)];
eulerangles_der=LRI*[p;q;r];
phid=eulerangles_der(1);
thetd=eulerangles_der(2);
psid=eulerangles_der(3);
pd=(Jyy-Jzz)*q*r/Jxx+U2*l/Jxx;
qd=(Jzz-Jxx)*p*r/Jyy+U3*l/Jyy;
rd=(Jxx-Jyy)*p*q/Jzz+U4/Jzz;
%Tracker calculation where the final is loaded from workspace.
[K,S,E]=lqr(final,Q,R);
Gt=pinv(final.C*pinv(-final.A+final.B*K)*final.B);
```