PROJECT REPORT

ON

KINEMATIC, DYNAMIC MODELING AND SIMULATION OF QUADCOPTER

Submitted By

Shaikh Altamash(12ME48)
Syed Adnan (12ME62)
Padwekar Aamir(13ME133)

Under The Supervision Of
Mr.Rahul R Thavai

In partial fulfillment of the requirement for degree of
BACHELOR OF ENGINEERING

In

MECHANICAL ENGINEERING

ANJUMAN-I-ISLAM's
KALSEKAR TECHNICAL CAMPUS
SCHOOL OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING

UNIVERSITY OF MUMBAI
2015 – 2016

Project Report

On

"Kinematic, Dynamic Modeling and Simulation of Quadcopter"

Submitted in partial fulfillment of the requirements

Of the degree of

Bachelor of Engineering

by

Shaikh Altamash(12ME48)
Syed Adnan (12ME62)
Padwekar Aamir(13ME133)

Supervisor

Mr.Rahul R Thavai

ANJUMAN-I-ISLAM's
KALSEKAR TECHNICAL CAMPUS
SCHOOL OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING

UNIVERSITY OF MUMBAI
2015 – 2016

Project Approval for B.E

This dissertation report entitled **"Kinematic, Dynamic Modelling and Simulation of Quadcopter"** by **Shaikh Altamash(12ME48), Syed Adnan (12ME62), Padwekar Aamir(13ME133)** is approved for the degree of **Bachelor of Engineering.**

External Examiner

Internal Examiner

Supervisor

HoD-Mechanical Engineering

Date:

Place:

Anjuman-I-Islam's

Kalsekar Technical Campus

School of Engineering & Technology

Department of Mechanical Engineering

Certificate

This is to certify that **Shaikh Altamash(12ME48), Syed Adnan (12ME62), Padwekar Aamir(13ME133)** has satisfactorily carried out the project work entitled **"Kinematic, Dynamic Modelling and Simulation of Quadcopter"**, for the degree of **Bachelor of Engineering** in **Mechanical Engineering** of **University of Mumbai.**

Project Supervisor	Head of the Department
Prof. Rahul R Thavai	Prof. Zakir Ansari

Declaration

I declare that this written submission represents my ideas in my own words and where other's ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shaikh Altamash(12ME48)-----------------
Syed Adnan (12ME62)-----------------------
Padwekar Aamir(13ME133)----------------

Date:

Acknowledgement

I give my true and sincere gratitude to **Mr.Rahul R Thavai** for his dedication and encouragement. His passion and broad knowledge in robotics nourished my growth. I would like to thank him for giving me his ideas to do this work as well as for his guidance, support and inspiration throughout the course. I would like to thank our HoD, **Mr. Zakir Ansari** for his constant encouragement throughout the course. To sum up, I would like to thank my friend with whom I am with. Of course, I don't have words to describe the support I received from my parents.

Shaikh Altamash(12ME48)
Adnan Syed (12ME62)
Padwekar Aamir(13ME133)

List of Figures

List of Tables

List of Abbreviation

UAV	Unmanned air Vehicles
VTOL	Vertical Takeoff and Landing

Abstract

Project consists of plotting various outputs of kinematic and dynamic motion with respect to time. This output is obtained by making a quad copter to trace a prescribed trajectory in the MATLAB simulation platform the centre of the mass of the quad copter is maintained at the centre of the quadcopter and the following governing kinematic and dynamic equations are prepared. These equations derived are used for the inception of the kinematic and dynamic modeling of the quad copter made to trace the trajectory. The outputs of the kinematic and dynamic modeling are made the input for the correction of error input and tuning of the PID control system. This closed function PID control system is used for the stabilization of the quad copter while following a given trajectory and optimize the output of the kinematic and dynamic modeling process for tuning purpose of the control.

Table of Contents

Chapter 1

Introduction

1.1 Introduction

A helicopter is a flying vehicle which uses rapidly spinning rotors to push air downwards, thus creating a thrust force keeping the helicopter aloft. Conventional helicopters have two rotors. These can be arranged as two coplanar rotors both providing upwards thrust, but spinning in opposite directions (in order to balance the torques exerted upon the body of the helicopter). The two rotors can also be arranged with one main rotor providing thrust and a smaller side rotor oriented laterally and counteracting the torque produced by the main rotor. However, these configurations require complicated machinery to control the direction of motion; a swash plate is used to change the angle of attack on the main rotors. In order to produce a torque the angle of attack is modulated by the location of each rotor in each stroke, such that more thrust is produced on one side of the rotor plane than the other.

The complicated design of the rotor and swash plate mechanism presents some problems, increasing construction costs and design complexity. A quad rotor helicopter (quad copter) is a helicopter which has four equally spaced rotors, usually arranged at the corners of a square body. With four independent rotors, the need for a swash plate mechanism is alleviated. The swash plate mechanism was needed to allow the helicopter to utilize more degrees of freedom, but the same level of control can be obtained by adding two more rotors. The development of quad copters has stalled until very recently, because controlling four independent rotors has proven to be incredibly difficult and impossible without electronic assistance. The decreasing cost of modern microprocessors has made electronic and even completely autonomous control of quad copters feasible for commercial, military, and even hobbyist purposes. Quad copter control is a fundamentally difficult and interesting problem.

With six degrees of freedom (three translational and three rotational) and only four independent inputs (rotor speeds), quad copters are severely under actuated. In order to achieve six degrees of freedom, rotational and translational motions are coupled. The resulting dynamics are highly nonlinear, especially after accounting for the complicated aerodynamic effects. Finally, unlike ground vehicles, helicopters have very little friction to prevent their motion, so they must provide their own damping in order to stop moving and remain stable. Together, these factors create a very interesting control problem. We will present a very simplified model of quad copter dynamics and design controllers for our dynamics to follow a designated trajectory. We will then test our controllers with a numerical simulation.

Unmanned air vehicles (UAVs) are self-propelled aerial robots. They can be equipped with various instruments and payloads, making them capable of performing various civilian or military tasks. Among existing small UAVs, we find quad rotors which are Vertical Take-Off and Landing (VTOL) four rotor helicopters .They are controlled simply by changing the rotation speed of the four rotors. The front and rear rotors (2, 4) rotate in a clockwise direction while the left and right rotors (1, 3) rotate in a counter-clockwise direction to balance the torque created by the spinning rotors. The up/down motion is achieved by increasing/decreasing the rotors speed while maintaining an equal individual speed. The forward/backward, left/right motions are achieved through a differential control strategy of rotors speed. Thanks to this configuration, quad rotors are able to hover, takeoff, and land in small areas and enable them to perform tasks that fixed-wing craft are unable to do. [4]

1.2 PROBLRM DEFINITION

- To carry out mathematical analysis of a quad copter following various forms of path
- To derive generalized governing kinematic and dynamic equations of the quad copter assuming its centre of gravity to be at the centre of the quad copter
- To carry out body frame analysis on various co-ordinate scales:
- Earth-frame(E-frame) analysis:- in this form of analysis the earth surface is taken as the reference and the x coordinate is assumed to be the north pole , the y coordinate is assumed to be the east and the z coordinate is assumed towards the earth. This also known as absolute or generalized coordinates system.
- Body -frame analysis (B-frame):-in this form of analysis the x, y & z coordinate is set as per the orientation and is more convenient way of rigid body motion analysis.
- To carry rigid body motion analysis sung MATLAB ra13 for simulation of yaw motion.
- To carry trajectory analysis of quad copter tracing a path using MATLAB ra13 for four conditions:-
  - hovering
  - pitching

- rolling
- yawing

This all orientations are conditional for the simulation of positional error of quad copter in a with respect to a time frame in seconds in 1 dimensional and 3 dimensional mode of results.

- To evaluate the error with respect to time over the output error function from the kinematic and dynamic modeling for a PID control system.
- To set adequate gain parameters for a PID control system so as to optimize stabilization parameter of a quad copter and prevent overshoot of the system by adequate tuning of the PID gain parameters

1.3 OBJECTIVE OF WORK:-

- Design and develop kinematic and dynamic simulation models for a quad copter moving over a trajectory
- Study the position and time variables of the motion of quad copter in 1 dimensional and 3 dimensional time variant scale
- Study and obtain the change in position , velocity and acceleration of the quad copter motion with respect to the variant time scale
- Study and obtain the results of third and fourth order derivative motions such as jerk and jounce and its error cost function analysis (J).
- Enable PID control system with automatic tuning to avoid steady state error by setting adequate and proper gain parameters via: kip, kid, kid so as to avoid overshoot to large error function value.

Chapter 2

Literature Survey

Sr. No	Name of the paper	Author	Year of Publishing	Inferences taken from the paper
1.	Geometric tracking and control of UAV’s	Taeyong Lee, Melvick Leok, N. Harris	Dec 2010	Eucladian group of non-linear tracking controller.
2.	Design and implementation of a 6 DOF control of an autonomous quad copter	Alexender Lebedev	Dec 2013	6 DOF control and dynamics of quad copter using forward kinematics
3.	Modeling, identification and control of a quad copter aircraft.	Marcello Ne Lelis Costa De Oliveria	June 2011	Design and study of feedback controller
4.	“Quad copters”	Andrew Depriest	Jan 2011	Introduction to design and development of dynamic and control of quad copter
5.	Modeling, identification and control of a quad copter helicopter	Tommaso Bresciani	Oct 2008	Study the dynamic system, modeling and control algorithm evaluation for example PID
6.	Simulation and experimental works of quad copter model for simple maneuver	Rafiuddin, Syam, Muslary	June 2015	For study of theoretical methods consisting of aircraft testing
7.	Path planning using concentrated analytically-defined trajectories	Jamie Biggs, Jonathan Jamieson	June 2015	Study of methods for a semi-analytical trajectory planning of quad copter
8.	Aerial robotics	Anibal	Jan 2011	Evaluation for techniques

	cooperative assembly system	Ollero		for interaction of UAV's to the environment
9.	Quad copter video surveillance of UAV	Anton, Nakazua, Bai Xiang, Jin	June 2012	Study of image processing for a given trajectory.
10.	Modeling and linear control of quad copter	C. Balas	2006-07	Study of modeling the rotor dynamics, decoupling the inputs and designing the control law
11.	Quad copter flight dynamics	Mohammed Khan	August 2014	Study the maneuvering scheme at a given altitude for pitching, rolling and yawing
12.	The modeling and simulation of autonomous quad copter, micro-quad copter in a virtual outdoor scenario	Gyula Mester	August 2011	Dynamic modeling and simulation of a quad copter
13.	Simulation and control of a quad copter unmanned aerial vehicla	Michael David Schmedt	June 2011	Deep study and analysis of kinematic, dynamic modeling, feedback controller and SIMULINK method
14.	Trajectory planning for a quad copter	M. Haddad, Y. Boutkir	June 2008	A simple direct method able to generate time optimal trajectories and modeling of quad copter

Table 2.1 Literature survey

Chapter 3

Proposed System

3.1 METHODOLOGY:-

The following steps will be adopted to prepare kinematic and dynamic model and get output.

Results on MATLAB ra13:-

- Study various thrust equations and propelling conditions for:-
  - hovering
  - rolling
  - pitching
  - yawing
- Study, analyze and evaluate the governing kinematic and dynamic equations for computation of position and time variable of a moving quad copter over a trajectory.
- Computing the above results on MATLAB ra13 simulation platform and obtain respective results of position versus time analysis by writing algorithms and commands on the simulation platform
- Computing the inertia and body frame equations of rigid bogy dynamics on MATLAB ra13 simulation platform.
- Comparing the results and obtain the error function as result of deviation of quad copter displacement with the reference trajectory.
- Optimizing the error function in the form "cost-function analysis" to enable the PID control system with automatic Tuning
- Decide gain parameters by testing against the steady state error for automatic tuning of PID Control system.
- Take the output of the kinematic and dynamic modeling and make it as input for evaluating the error function with respect to time to nullify the steady state error.
- Develop and build SIMULINK* model of kinematic and dynamic model and control inputs to the PID control system on MATLAB ra13 simulation platform.

3.2 Generalized coordinate system

In analytical mechanics, specifically the study of the rigid body dynamics of multi body systems, the term generalized coordinates refers to the parameters that describe the configuration of the system relative to some reference configuration. These parameters must uniquely define the configuration of the system relative to the reference configuration. The generalized velocities are the time derivatives of the generalized coordinates of the system. An example of a generalized coordinate is the angle that locates a point moving on a circle. The adjective "generalized" distinguishes these parameters from the traditional use of the term coordinate to refer to Cartesian coordinates: for example, describing the location of the point on the circle using x and y coordinates.

Although there may be many choices for generalized coordinates for a physical system, parameters which are convenient are usually selected for the specification of the configuration of the system and which make the solution of its equations of motion easier. If these parameters are independent of one another, the number of independent generalized coordinates is defined by the number of degrees of freedom of the system. Generalized coordinates are usually selected to provide the minimum number of independent coordinates that define the configuration of a system, which simplifies the formulation of Lagrange's equations of motion. However, it can also occur that a useful set of generalized coordinates may be dependent, which means that they are related by one or more constraint equations.[5][7]

3.3 Holonomic constraints

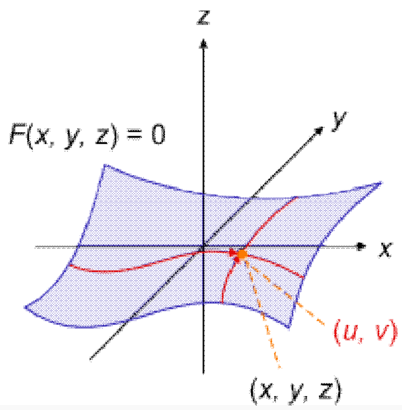


Fig 3.1(A) Open curved surface $F(x, y, z) = 0$

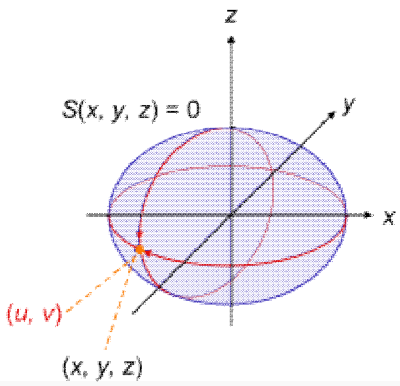


Fig 3.1(B) Closed curved surface *S*(*x*, *y*, *z*) = 0

For a system of *N* particles in 3d real coordinate space, the position vector of each particle can be written as a 3-tuple in Cartesian coordinates;

$$r_1 = (x_1, y_1, z_1), \quad r_2 = (x_2, y_2, z_2), \ldots, r_N = (x_N, y_N, z_N)$$

Any of the position vectors can be denoted r_k where $k = 1, 2, \ldots, N$ labels the particles. A holonomic constraint is a constraint equation of the form for particle *k*

$$f(r_k, t) = 0$$

Which connects all the 3 spatial coordinates of that particle together, so they are not independent.

It is ideal to use the minimum number of coordinates needed to define the configuration of the entire system, while taking advantage of the constraints on the system. These quantities are known as generalized coordinates in this context, denoted $q_j(t)$. It is convenient to collect them into an *n*-tuple

$$q(t) = (q_1(t), q_2(t), \ldots, q_n(t))$$

Which is a point in the configuration space of the system. They are all independent of one other, and each is a function of time. Geometrically they can be lengths along straight lines, or arc lengths along curves, or angles; not necessarily Cartesian coordinates or other standard orthogonal coordinates. There is one for each degree of freedom, so the number of generalized coordinates equals the number of degrees of freedom, *n*. A degree of freedom corresponds to one quantity that changes the configuration of the system, for example the angle of a pendulum, or the arc length traversed by a bead along a wire.[8]

If it is possible to find from the constraints as many independent variables as there are degrees of freedom, these can be used as generalized coordinates. The position vector r_k of particle *k* is a function of all the *n* generalized coordinates and time

$$r_k = r_k(q(t), t)$$

and the generalized coordinates can be thought of as parameters associated with the constraint.

The corresponding time derivatives of q are the generalized velocities,

$$\dot{q} = \frac{dq}{dt} = (\dot{q}_1(t), \dot{q}_2(t), \ldots, \dot{q}_n(t))$$

(each dot over a quantity indicates one time derivative). The velocity vector v_k is the total derivative of r_k with respect to time

$$V_k = \dot{r}_k = \frac{dr_k}{dt} = \sum_{j=1}^{n} \frac{\partial r_k}{\partial q_j} . \dot{q}_j + \frac{\partial r_k}{\partial t}$$

and so generally depends on the generalized velocities and coordinates. Since we are free to specify the initial values of the generalized coordinates and velocities separately, the generalized coordinates and velocities can be treated as independent variables. The generalized coordinates q_j and velocities dq_j/dt are treated as independent variables.[8]

3.4 Non-holonomic constraints

A mechanical system can involve constraints on both the generalized coordinates and their derivatives. Constraints of this type are known as non-holonomic. First-order non-holonomic constraints have the form

$$g(q,\dot{q},t)=0,$$

An example of such a constraint is a rolling wheel or knife-edge that constrains the direction of the velocity vector. Non-holonomic constraints can also involve next-order derivatives such as generalized accelerations.

3.5 Physical Quantities in Generalized coordinates

Kinetic energy

The total kinetic energy of the system is the energy of the system's motion, defined as

$$T=\frac{1}{2}\sum_{k=1}^{N} m_k \dot{r}_k . \dot{r}_k,$$

in which · is the dot product. The kinetic energy is a function only of the velocities v_k, not the coordinates r_k themselves. By contrast an important observation is

$$\dot{r}_k.\dot{r}_k=\sum_{i,j=1}^{n}\left(\frac{\partial r_k}{\partial q_i}.\frac{\partial r_k}{\partial q_j}\right)\dot{q}_i\dot{q}_j + \sum_{i=1}^{n}\left(\frac{2\,\partial r_k}{\partial q_i}.\frac{\partial r_k}{\partial t}\right)q_i + \left(\frac{\partial r_k}{\partial t}.\frac{\partial r_k}{\partial t}\right),$$

which illustrates the kinetic energy is in general a function of the generalized velocities, coordinates, and time if the constraint also varies with time, so T = T(q, dq/dt, t).

In the case the constraint on the particle is time-independent, then all partial derivatives with respect to time are zero, and the kinetic energy has no time-dependence and is a homogeneous function of degree 2 in the generalized velocities;

$$\dot{r}_k.\dot{r}_k=\sum_{i,j=1}^{n}\left(\frac{\partial r_k}{\partial q_i}.\frac{\partial r_k}{\partial q_j}\right)\dot{q}_i\dot{q}_j$$

$$ds^2=dr_k.\ dr_k=\sum_{i,j=1}^{n}\left(\frac{\partial r_k}{\partial q_i}.\frac{\partial r_k}{\partial q_j}\right)\dot{q}_i\dot{q}_j$$

and dividing by the square differential in time, dt^2, to obtain the velocity squared of particle *k*.

It is instructive to see the various cases of polar coordinates in 2d and 3d, owing to their frequent appearance. In 2d polar coordinates (r, θ),

$$\left(\frac{ds}{dt}\right)^2=\dot{r}^2+r^2\dot{\theta}^2$$

in 3d cylindrical coordinates (r, θ, z),

$$\left(\frac{ds}{dt}\right)^2=\dot{r}^2+r^2\dot{\theta}^2+\dot{z}^2$$

in 3d spherical coordinates (r, θ, φ),

$$\left(\frac{ds}{dt}\right)^2=\dot{r}^2+r^2\dot{\theta}^2+r^2(\sin\theta\dot{\emptyset}^2)^2$$

Chapter 4

Kinematic Model

4.1 Earth frame analysis v/s Body frame analysis

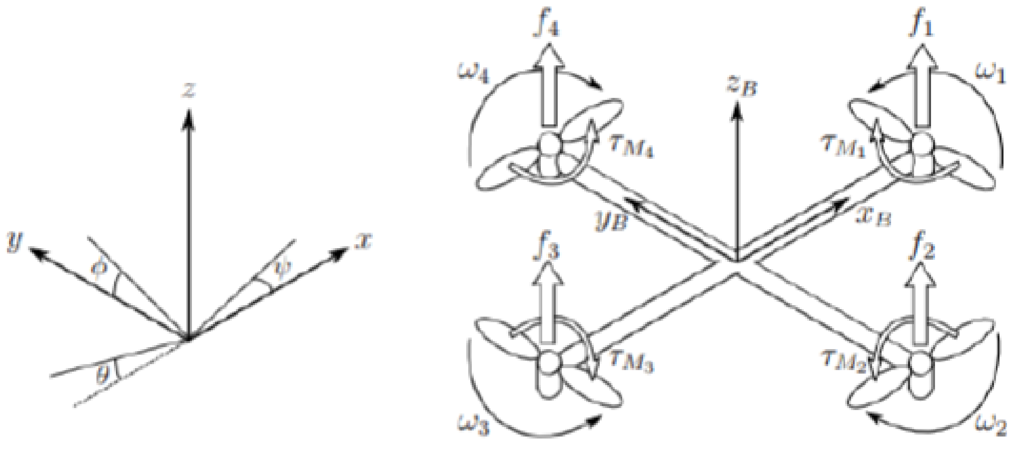


Figure 1: The inertial and body frames of a quadcopter

Fig 4.1 The inertia and body frames of a quad copter

Quad copter:

●Position

●Pitch, roll, yaw

●Pose

Body Frame:

●Linear Velocity

●Angular Velocity

Body-to-Inertial Frame:

●Rotation matrix orthogonal

- ➢ $R^{-1} = R^T$
- ➢ Inertia-to-body

$$\xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad q = \begin{bmatrix} \xi \\ \eta \end{bmatrix}, \quad V_B = \begin{bmatrix} v_{x}, B \\ v_{y}, B \\ v_{z}, B \end{bmatrix}, \quad v = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

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

$S_x = \sin x$ and $C_x = \cos x$

4.2 Transformation matrices (angular vel.)

- inertial-to-body
- body-to-inertial
  - Symmetric structure
- Inertia matrix is diagonal
  - Lift force -lift constant and angular vel.
  - Torque -drag constant and angular vel.
- inertia moment term small be omitted
  - Roll = -2nd rotor, +4th rotor
  - Pitch = -1st rotor, +3rd rotor
  - Yaw = +/-(+1st, +3rd, -2nd, -4th)

$\dot{\eta} = W_n^{-1} v$,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & S_\phi T_\theta & C_\phi T_\theta \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi / C_\theta & C_\phi / C_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$T_x = \tan x$

$v = W_n \dot{\eta}$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

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

$$I_{xx} = I_{yy}$$

$$f_i = kw_i^2, \quad T_{M_i} = bw_i^2 + I_M w_i,$$

$$T = \sum_{i=1}^{4} f_i = k \sum_{i=1}^{4} w_i^2, \qquad T^B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}$$

$$T_B = \begin{bmatrix} T_\phi \\ T_\theta \\ T_\psi \end{bmatrix} = \begin{bmatrix} lk(-w_2^2 + w_4^2) \\ lk(-w_1^2 + w_3^2) \\ \sum_{i=1}^{4} T_{M_i} \end{bmatrix}$$ [1]

Chapter 5

Dynamic Model

5.1 Euler-Lagrange equations

The Euler–Lagrange equation was developed in the 1750s by Euler and Lagrange in connection with their studies of the tautochrone problem. This is the problem of determining a curve on which a weighted particle will fall to a fixed point in a fixed amount of time, independent of the starting point.

Lagrange solved this problem in 1755 and sent the solution to Euler. Both further developed Lagrange's method and applied it to mechanics, which led to the formulation of Lagrangian mechanics. Their correspondence ultimately led to the calculus of variations, a term coined by Euler himself in 1766

The Lagrangian L is the sum of the translational Etrans and rotational Erot energies minus potential energy Epot

$L(q,\dot{q}) = E_{trans} + E_{rot} - E_{pot}$

$= (m/2)\dot{\xi}^T \dot{\xi} + (1/2)\, \nu^T I \nu - mgz$

As shown in the Euler-Lagrange equations with external force

s and torques are

$$\begin{bmatrix} f \\ \tau \end{bmatrix} = \frac{d}{dt}\left(\frac{\partial l}{\partial \dot{q}}\right) - \frac{\partial l}{\partial \dot{q}}$$

Where L is the Lagrangian of the quad rotor model, q=[Ω*z] is the state vector and “τ“ represents the roll, pitch and yaw moments and “f “ is the translational force applied to the quad rotor. In order to simplify the system dynamics, the total dynamics is divided into translational and rotational dynamics by considering the respective state vectors. The linear and angular components do not depend on each other thus they can be studied separately. The linear external force is the total thrust of the rotors. The linear Euler-Lagrange equations are

$$f = RT_B = m\ddot{\xi} + mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The Jacobian matrix J(η) from ν to η is

$J(\eta) = J = W^T_\eta I W_\eta$

$$\begin{bmatrix} I_{ZZ} & 0 & -I_{ZZ}S_\theta \\ 0 & I_{YY}C_\phi^2 + I_{ZZ}S_\phi^2 & (I_{YY} - I_{ZZ})C_\phi S_\phi C_\theta \\ -I_{XX}S_\theta & (I_{YY} - I_{ZZ})C_\phi S_\phi C_\theta & I_{ZZ}S_\theta^2 + I_{YY}S_\varphi^2 C_\theta^2 + I_{ZZ}C_\phi^2 C_\theta^2 \end{bmatrix}$$

Thus, the rotational energy Erot can be expressed in the inertial frame as

$E_{rot\,=}$ (1/2) $_V{}^T$ I $_V$= (1/2) $\ddot{\eta}$ J $\ddot{\eta}$

The external angular force is the torques of the rotors. The angular Euler-Lagrange equations are

$T=T_B=J\ddot{\eta}+\frac{d}{dt}(J)\dot{\eta}-\frac{1}{2}\frac{\partial}{\partial\eta}(\dot{\eta}^T J\dot{\eta})$ $=J\ddot{\eta}+ C(\eta,\dot{\eta})\dot{\eta}$

In which the matrix C(η,η) is the Coriolis term, containing the gyroscopic and centripetal terms

The matrix $C(\eta,\dot{\eta})$ has the form

$$C(\eta,\dot{\eta}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}$$

$C_{11}=0$

$C_{12}= (I_{YY} - I_{ZZ})(\dot{\theta}C_\phi S_\phi + \dot{\psi}S_\phi^2 C_\theta) + (I_{ZZ} - I_{YY})\dot{\psi}C_\phi^2 C_\theta - I_{XX}\dot{\psi}C_\theta$

$C_{13}=(I_{ZZ} - I_{YY})\dot{\psi}C_\phi S_\phi C_\theta^2$-

$C_{21}=(I_{ZZ} - I_{YY})(\dot{\theta}C_\phi S_\phi + \dot{\psi}S_\phi C_\theta) + (I_{YY} - I_{ZZ})\dot{\psi}C_\phi^2 C_\theta + I_{XX}\dot{\psi}C_\theta$

$C_{22}=(I_{ZZ} - I_{YY})\dot{\phi}C_\phi S_\phi$

$C_{23}=-I_{XX}\dot{\psi}S_\theta C_\theta + I_{YY}\dot{\psi}S_\phi^2 S_\theta C_\theta + I_{ZZ}\dot{\psi}C_\phi^2 S_\theta C_\theta$

$C_{31}=(I_{YY} - I_{ZZ})\dot{\psi}C_\theta^2 S_\phi C_\phi - I_{XX}\dot{\theta}C_\theta$

$C_{32}=(I_{ZZ} - I_{YY})(\dot{\theta}C_\phi S_\phi S_\theta + \dot{\phi}S_\phi^2 C_\theta) + (I_{YY} - I_{ZZ})\dot{\phi}C_\phi^2 C_\theta + I_{XX}\dot{\psi}S_\theta C_\theta - I_{YY}\dot{\psi}S_\varphi^2 S_\theta C_\theta -$
$I_{ZZ}\dot{\psi}C_\phi^2 S_\theta C_\theta$

$C_{33}= (I_{YY} - I_{ZZ})\dot{\phi}C_\phi S_\phi C_\theta^2 - I_{YY}\dot{\theta}S_\phi^2 C_\theta S_\theta - I_{ZZ}\dot{\theta}C_\phi^2 C_\theta S_\theta + I_{XX}\dot{\theta}C_\theta S_\theta$

$\ddot{\eta} = J^{-1}(\tau_B - C(\eta,\dot{\eta})\dot{\eta})$[2]

Chapter 6

Control

6.1 PID Control:

A proportional–integral–derivative controller (PID controller) is a control loop feedback mechanism (controller) commonly used in industrial control systems. A PID controller continuously calculates an error value as the difference between a desired set point and a measured process variable. The controller attempts to minimize the error over time by adjustment of a control variable, such as the position of a control valve, a damper, or the power supplied to a heating element, to a new value determined by a weighted sum:

$$u(t) = K_p e(t)\ K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}$$

Where K_p, K_i, and K_d all non-negative, denote the coefficients for the proportional, integral, and derivative terms, respectively (sometimes denoted P, I, and D). In this model, P accounts for present values of the error. For example, if the error is large and positive, the control output will also be large and positive. I accounts for past values of the error. For example, if the current output is not sufficiently strong, error will accumulate over time, and the controller will respond by applying a stronger action. D accounts for possible future values of the error, based on its current rate of change.

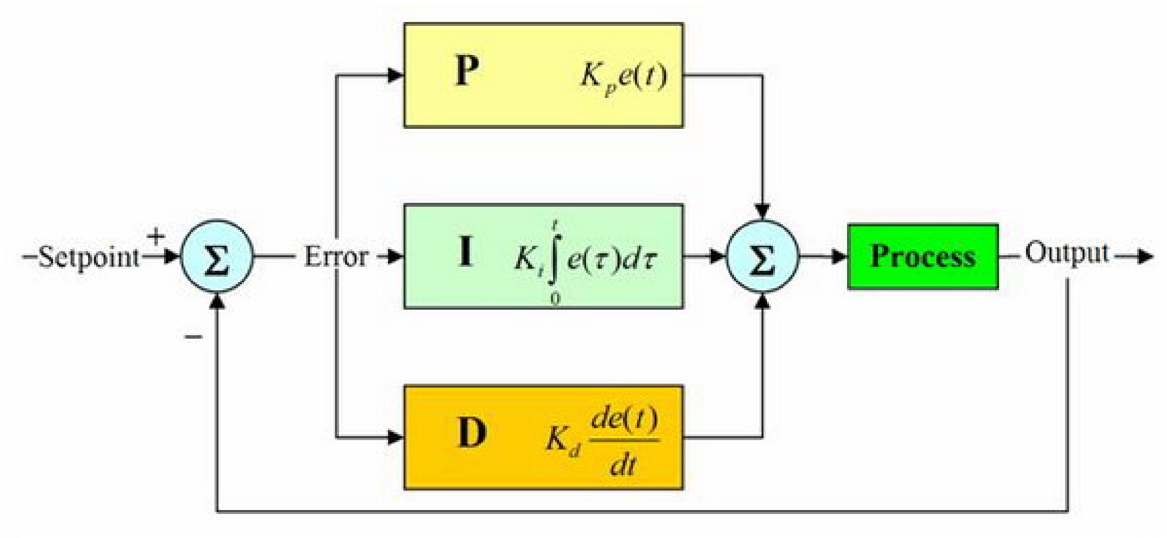


Fig 6.1(A) Flow diagram of PID control

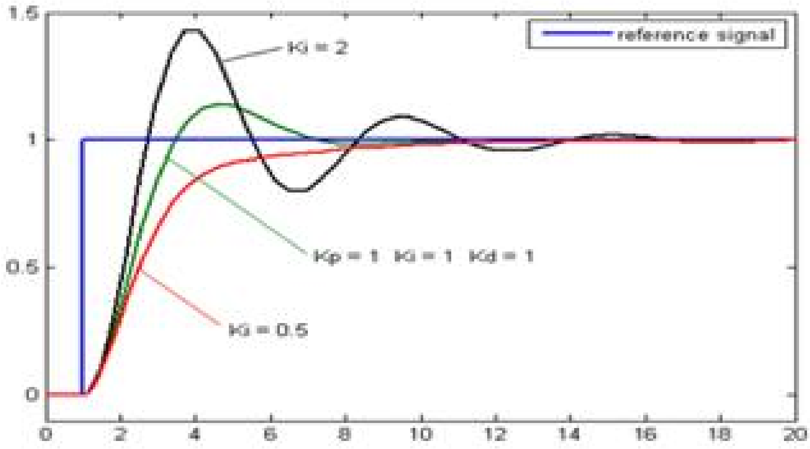


Fig 6.1(B) Graph of PID control

The PID controller is the most common form of feedback. It was an essential element of early governors and it became the standard tool when process control emerged in the 1940s. In process control today, more than 95% of the control loops are of PID type, most loops are actually PI control. PID controllers are today found in all areas where control is used. The controllers come in many different forms. There are standalone systems in boxes for one or a few loops, which are manufactured by the hundred thousand yearly. PID control is an important ingredient of a distributed control system. The controllers are also embedded in many special purpose control systems. PID control is often combined with logic, sequential functions, selectors, and simple function blocks to build the complicated automation systems used for energy production, transportation, and manufacturing.

Many sophisticated control strategies, such as model predictive control, are also organized hierarchically. PID control issued at the lowest level; the multivariable controller gives the set points to the controllers at the lower level. The PID controller can thus be said to be the “bread and butter's' of control engineering. It is an important component in every control engineer’s tool box. PID controllers have survived many changes in technology, from mechanics and pneumatics to microprocessors via electronic tubes, transistors, integrated circuits. The microprocessor has had a dramatic influence on the PID controller. Practically all PID controllers made today are based on microprocessors. This has given opportunities to provide additional features like automatic tuning, gain scheduling, and continuous adaptation.[8]

6.2 Tuning:

All general methods for control design can be applied to PID control. A number of special methods that are tailor made for PID control have also been developed, these methods are often called tuning methods. Irrespective of the method used it is essential to always consider the key elements of control, load disturbances, sensor noise, process uncertainty and reference signals. The most well known tuning methods are those developed by Ziegler and Nichols. They have had a major influence on the practice of PID control for more than half a century. The methods are based on characterization of process dynamics by a few parameters and simple equations for the controller parameters. It is surprising that the methods are so widely referenced because they give moderately good tuning only in restricted situations. Plausible explanations may be the simplicity of the methods and the fact that they can be used for simple student exercises in basic control courses.[8]

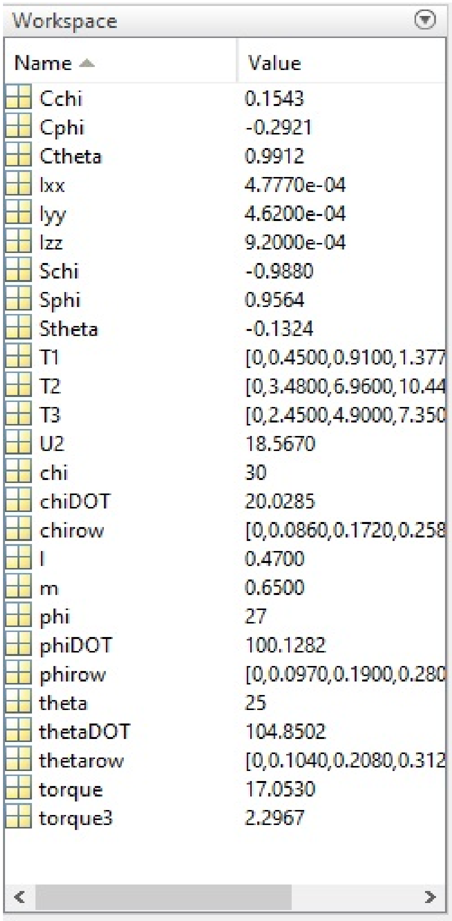
6.3 State Space Representation:

In control engineering, a state-space representation is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations. "State space" refers to the space whose axes are the state variables. The state of the system can be represented as a vector within that space.

To abstract from the number of inputs, outputs and states, these variables are expressed as vectors. Additionally, if the dynamical system is linear, time-invariant, and finite-dimensional, then the differential and algebraic equations may be written in matrix form. The state-space method is characterized by significant algebraization of general system theory, which makes possible to use Kronecker vector-matrix structures. The capacity of these structures can be efficiently applied to research systems with modulation or without it. The state-space representation (also known as the "time-domain approach") provides a convenient and compact way to model and analyze systems with multiple inputs and outputs. With p inputs and q outputs, we would otherwise have to write down p x q Laplace transforms to encode all the information about a system. Unlike the frequency domain approach, the use of the state-space representation is not limited to systems with linear components and zero initial conditions.[8]

Chapter 7
Simulation

Dynamic Analysis:



Workspace

Name	Value
Cchi	0.1543
Cphi	-0.2921
Ctheta	0.9912
Ixx	4.7770e-04
Iyy	4.6200e-04
Izz	9.2000e-04
Schi	-0.9880
Sphi	0.9564
Stheta	-0.1324
T1	[0,0.4500,0.9100,1.377
T2	[0,3.4800,6.9600,10.44
T3	[0,2.4500,4.9000,7.350
U2	18.5670
chi	30
chiDOT	20.0285
chirow	[0,0.0860,0.1720,0.258
l	0.4700
m	0.6500
phi	27
phiDOT	100.1282
phirow	[0,0.0970,0.1900,0.280
theta	25
thetaDOT	104.8502
thetarow	[0,0.1040,0.2080,0.312
torque	17.0530
torque3	2.2967

Fig 7.1(A) Dynamic Analysis of Input Torque using Euler-Lagrange formulations

Ixx=477.7e-6

Ixx =4.7770e-04

>> m=0.65

m =0.6500

>> l=0.47

l =0.4700

>>theta=25

theta =25

>>phi=25

```
>>theta=25

theta =25

>>phi=27

phi =27

>>chi=30

chi =30

>>Ctheta=cos(theta)

Ctheta =0.9912

>> Stheta=sin(25)

Stheta =-0.1324

>> Cphi=cos(27)

Cphi =-0.2921

>> Sphi=sin(27)

Sphi =0.9564

>> Schi=sin(30)

Schi =-0.9880

>>Cchi=cos(30)

Cchi =0.1543

>>thetaDOT = 315.9

thetaDOT =315.9000

>>thetaDOT=104.8502

thetaDOT =104.8502

>> U2=18.567

U2 =18.5670

>>phiDOT=   100.1282
```

phiDOT =100.1282

>>chiDOT= 20.0285

chiDOT =20.0285

>>Iyy=462.002e-6

Iyy =4.6200e-04

>>torque=(((m*phiDOT*l^2)+(Iyy*phiDOT)-(m*9.81*Sphi*Cphi*l))-((m*9.81*Cphi)+(m*0.4881*l^2)+(Ixx*0.4881)))

torque =17.0530

>>Izz=920.002e-6

Izz= 9.2000e-04

>>torque3=(((m*chiDOT*l^2)+(Izz*chiDOT)-(m*9.91*Schi*Cchi*l))-((m*9.81*Cchi)+(m*0.5233*l^2)+(Ixx*0.5233)))

torque3 =2.2967

>> T1=[0 0.5 1 1.5 2 2.297 2.5]

T1 = 0 0.5000 1.0000 1.5000 2.0000 2.2970 2.5000

>>chirow=[0 0.1 0.2 0.3 0.4 0.43 0.5]

chirow =0 0.1000 0.2000 0.3000 0.4000 0.4300 0.5000

>>plot(chirow,T1)

>>chirow=[0 0.086 0.172 0.258 0.344 0.43]

chirow =0 0.0860 0.1720 0.2580 0.3440 0.4300

>> T1=[0 0.45 0.91 1.3774 1.83 2.25]

T1 = 0 0.4500 0.9100 1.3774 1.8300 2.2500

>>plot(chirow,T1)

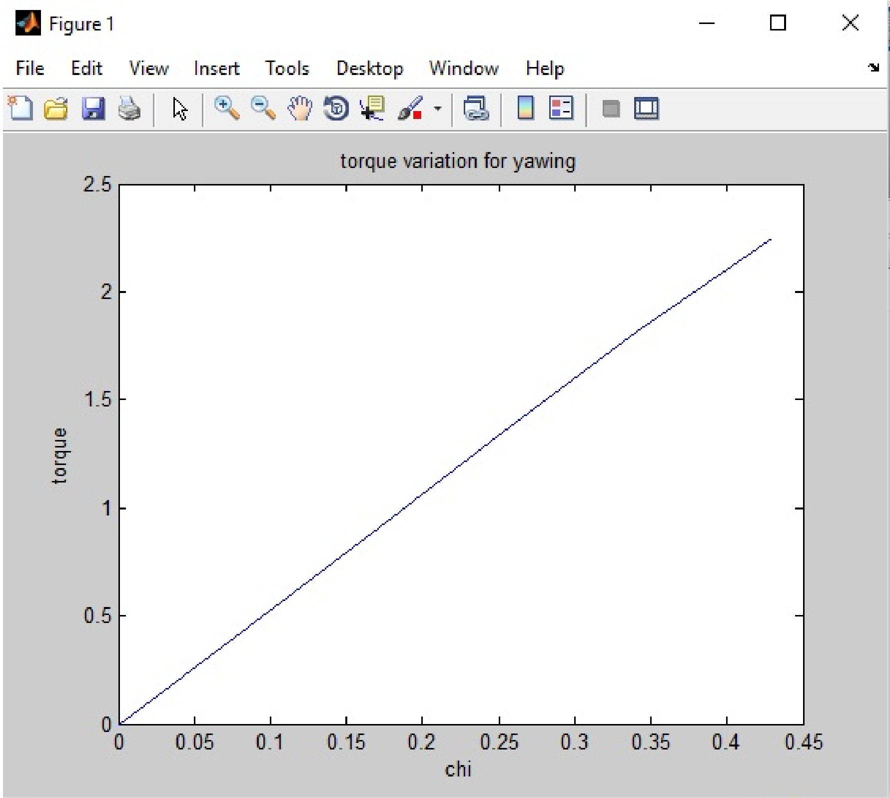


Fig 7.1(B) Torque variation for yawing vs chi angle

```
>> T2=[0 3.48 6.96 10.44 13.92 17.4]

T2 = 0    3.4800    6.9600   10.4400   13.9200   17.400

>>phirow=[0 0.097 0.19 0.28 0.384 0.48810]

phirow =0    0.0970    0.1900    0.2800    0.3840    0.4881

>>plot(phirow,T2)
```

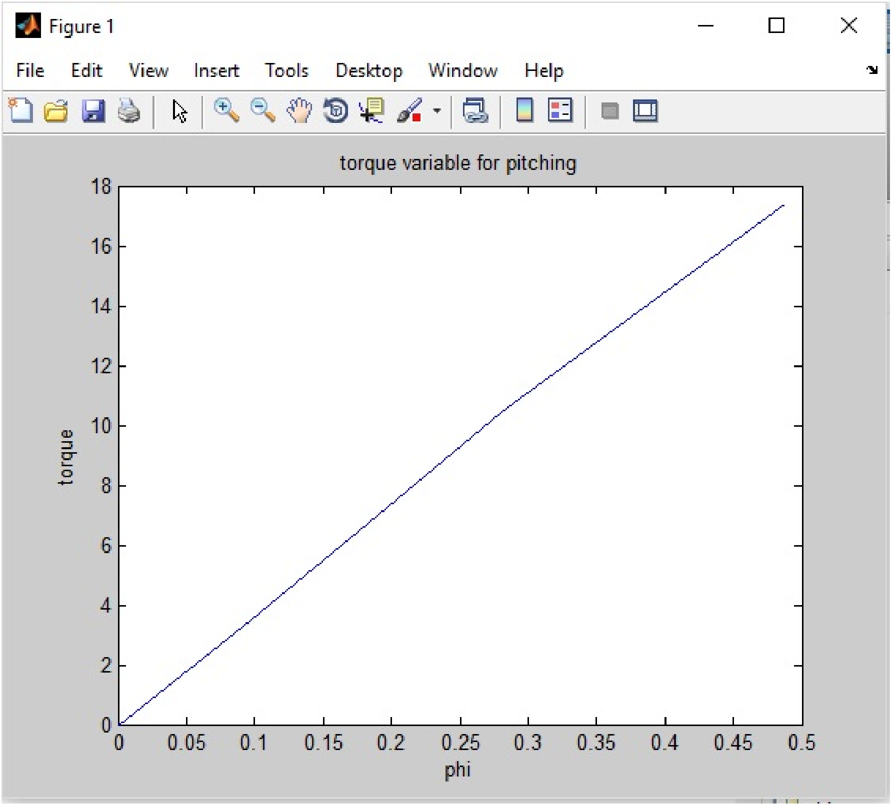


Fig 7.1 (C) Torque variable for pitching vs phi angle

```
>> T1=[0 0.45 0.91 1.3774 1.83 2.25]

T1 = 0    0.4500    0.9100    1.3774    1.8300    2.2500

>>chirow=[0 0.1 0.2 0.3 0.4 0.43 0.5]

chirow =0    0.1000    0.2000    0.3000    0.4000    0.4300    0.5000

>>chirow=[0 0.086 0.172 0.258  0.344 0.43]

chirow =0    0.0860    0.1720    0.2580    0.3440    0.4300

>>plot(chirow,T1)

>> T3=[0 2.45 4.9 7.35 9.8 12.46]

T3 = 0    2.4500    4.9000    7.3500    9.8000   12.4600

>>thetarow=[0 0.104 0.208 0.312 0.416 0.52]

thetarow =0    0.1040    0.2080    0.3120    0.4160    0.5200

>>plot(thetarow,T3)
```

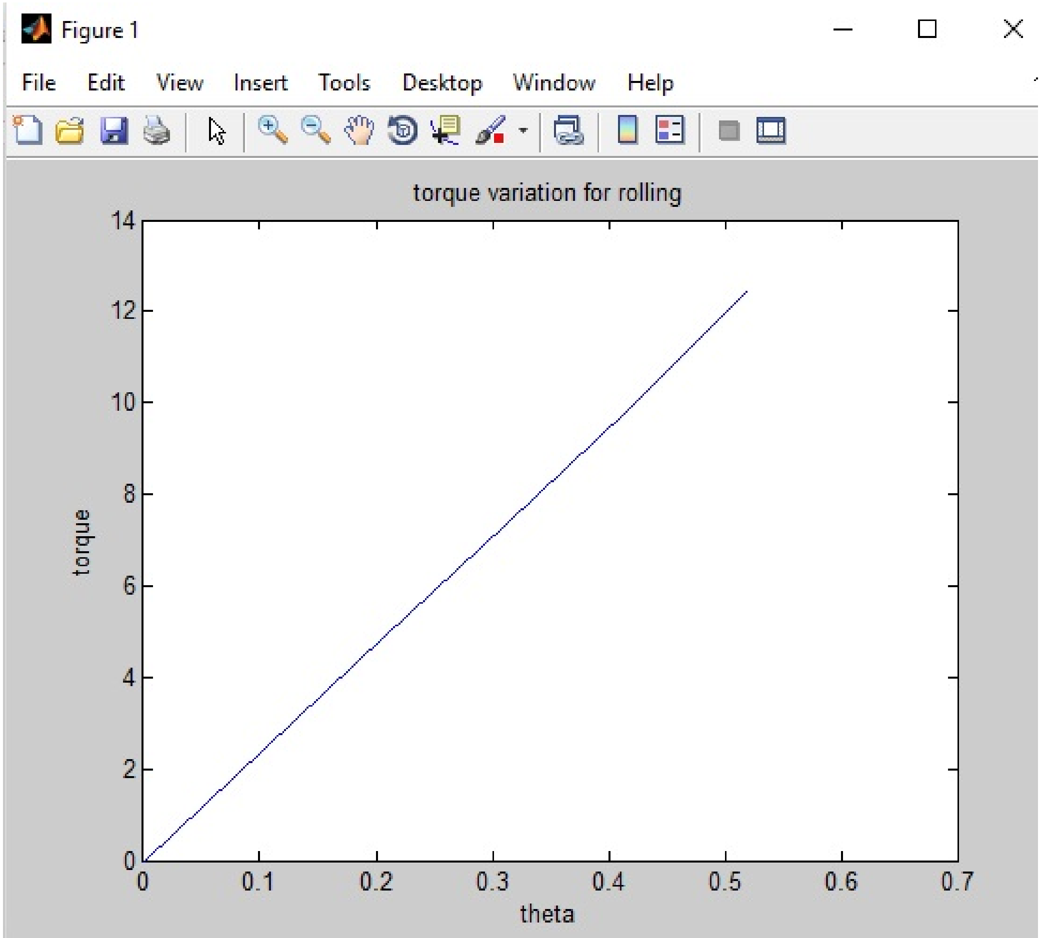
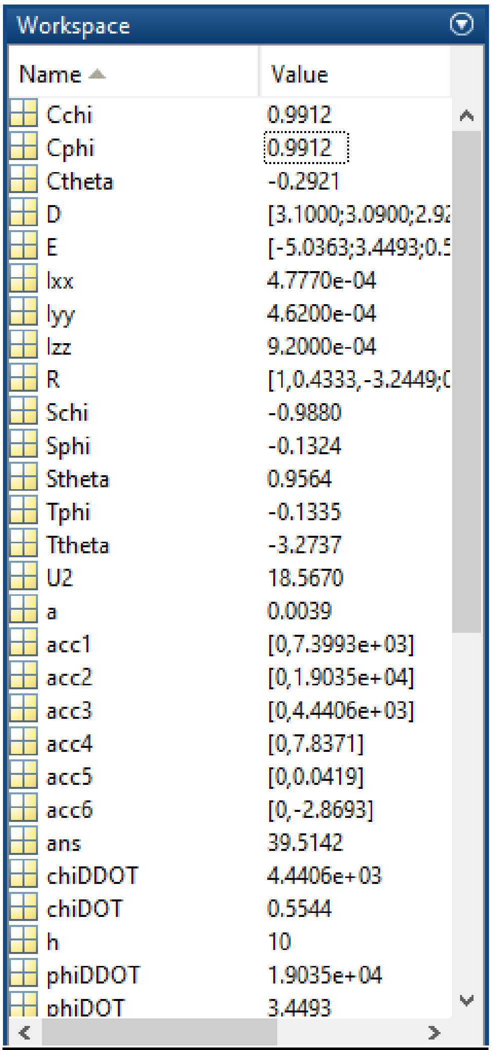


Fig 7.1(D) Torque variation for rolling vs theta angle

7.2 Kinematic analysis

Workspace:-


Workspace

Name	Value
Cchi	0.9912
Cphi	0.9912
Ctheta	-0.2921
D	[3.1000;3.0900;2.92
E	[-5.0363;3.4493;0.5
Ixx	4.7770e-04
Iyy	4.6200e-04
Izz	9.2000e-04
R	[1,0.4333,-3.2449;(
Schi	-0.9880
Sphi	-0.1324
Stheta	0.9564
Tphi	-0.1335
Ttheta	-3.2737
U2	18.5670
a	0.0039
acc1	[0,7.3993e+03]
acc2	[0,1.9035e+04]
acc3	[0,4.4406e+03]
acc4	[0,7.8371]
acc5	[0,0.0419]
acc6	[0,-2.8693]
ans	39.5142
chiDDOT	4.4406e+03
chiDOT	0.5544
h	10
phiDDOT	1.9035e+04
phiDOT	3.4493

Fig 7.2(A) Kinematic Analysis using forward kinematics:-

```
>> h=10;

>> r=1000;

>> w=0.71416;

>> t=(r*a*h)+ w

>> t =
```

```
>> 39.5142

ans =39.5142

>> r=1000;

>> a=0.00388;

>> h=10;

>> w=0.71416;

>> t=(((r*a*h)+ w)\ cos(27))

t =-0.0074

>> t=(((r*a*h)+ w)\ cos(25))

t =0.0251

>>tx=sqrt((t*(cos(27))^2)*(1-(1\(cos(27)^2))))

tx =0.0443

>>ty

Undefined function or variable 'ty'

>>tz

Undefined function or variable 'tz'.

>> ty= t*(cos(27))*(sin(25))

ty =9.6990e-04

>>tz= t*(cos(27))*(cos(25))

tz =-0.0073

>>Ctheta = cos(27)

>>Ctheta = cos(27)

Ctheta =-0.2921

>>Cphi=cos(25)

Cphi =0.9912
```

>> Cchi=cos(25)

Cchi =0.9912

>> Stheta=sin(27)

Stheta =0.9542

>> Sphi=sin(25)

Sphi = -0.1324

>> Schi=sin(30)

Schi =-0.9880

>>Ixx=477.7e-6

Ixx =4.7770e-04

>>Iyy=462.002e-6

Iyy =4.6200e-04

>>Izz=920.002e-6

Izz =9.2000e-04

>> Ttheta=tan(27)

Ttheta =-3.2737

>> Tphi=tan(25)

Tphi =-0.1335

>> R=[1 (Sphi*Ttheta) (Cphi*Ttheta); 0 Cphi -Sphi; 0 (Sphi/Ctheta) Ctheta*Cphi]

```
R =1.0000    0.4333   -3.2449
        0    0.9912    0.1324
        0    0.4530   -0.2896
```

>> D=[3.1,3.09,2.92]

D =3.1000   3.0900   2.9200

>> D=[3.1;3.09;2.92]

```
D =3.1000

    3.0900

    2.9200

>> E=R*D

E =-5.0363

    3.4493

    0.5544

>>thetaDOT=-5.0363

thetaDOT =-5.0363

>>phiDOT= 3.4493

phiDOT =3.4493

>>chiDOT=0.5544

chiDOT =0.5544


>>
chiDDOT=((phiDOT*chiDOT*Tphi)+((phiDOT*thetaDOT)/Cphi)+((Schi*Tphi*4.48)/Ixx)+((C
chi*Tphi*4.024)/Iyy)+(4.024/Izz)+(((Iyy-Izz)/Izz)*(chiDOT-
(thetaDOT*Sphi)))*(phiDOT*Tphi))

chiDDOT =4.4406e+03
```

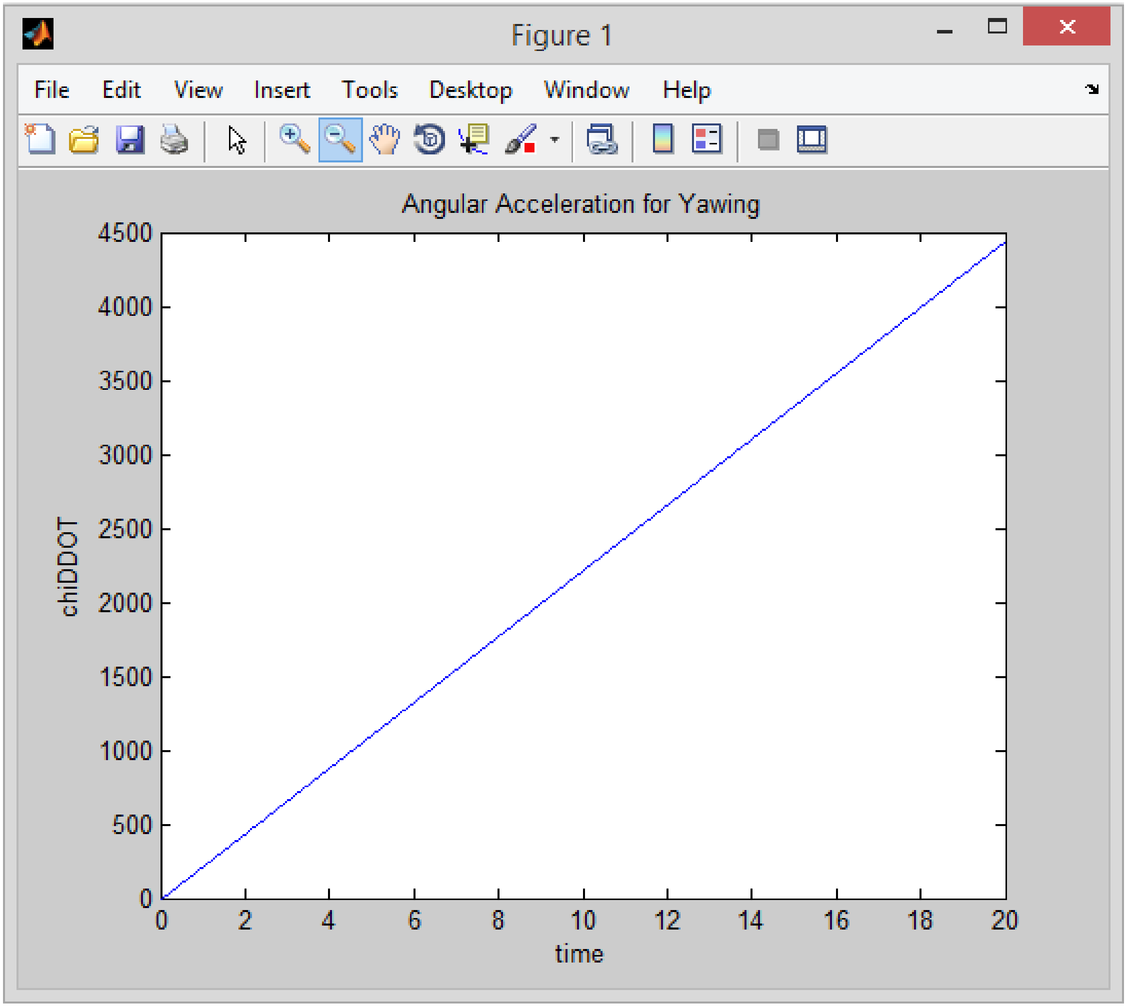


Fig 7.2(B) Angular Acceleration for yawing vs Time

```
>>thetaDDOT=(((phiDOT*chiDOT)/Cphi)+(phiDOT*thetaDOT*phiDOT)+((Sphi*4.48)/(Cphi*Ixx))+((Cchi*4.024)/(Cphi*Iyy))-(((Iyy-Izz)/Ixx)*(chiDOT-(thetaDOT*Sphi))*(phiDOT/Cphi)))

thetaDDOT =

7.3993e+03
```

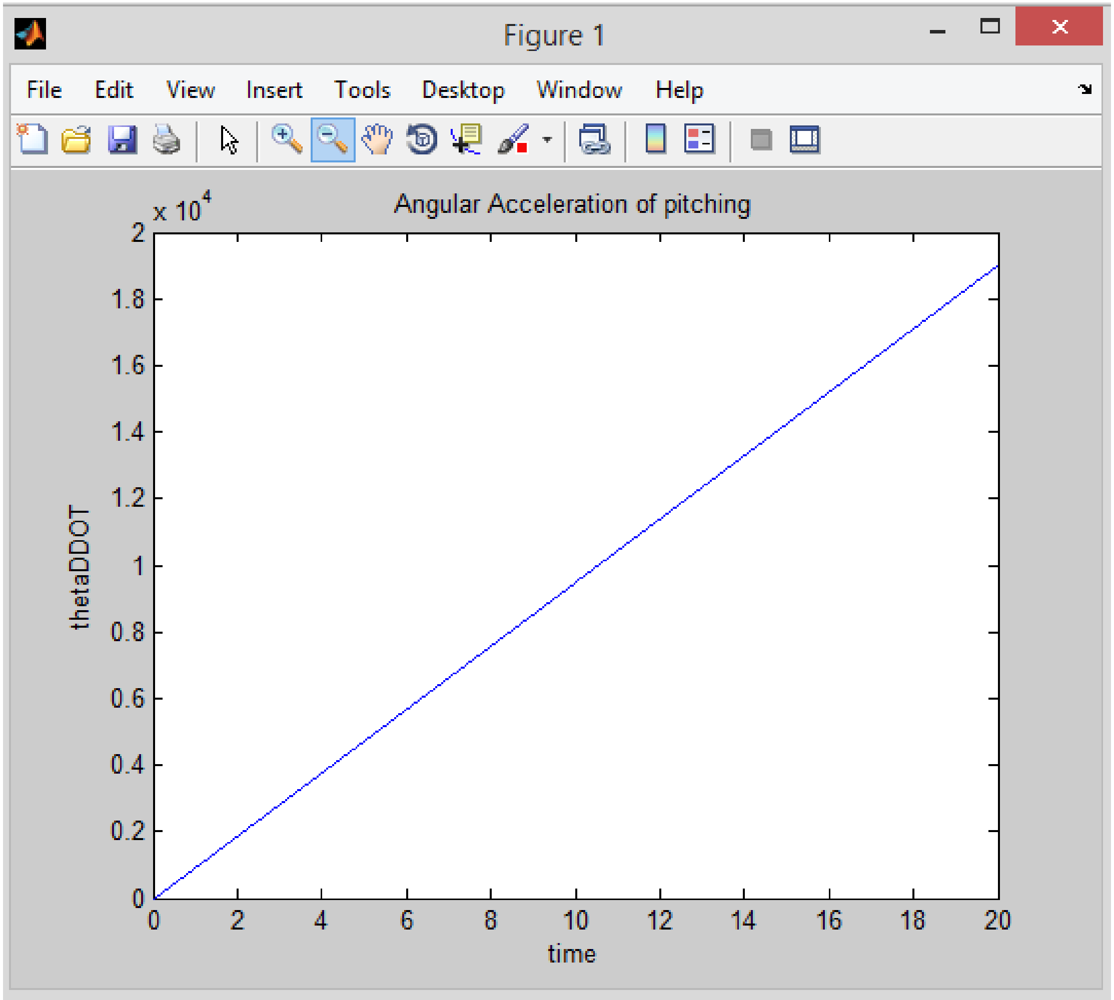


Fig 5.2(C) Angular acceleration for pitching vs Time

>> phiDDOT=((-chiDOT*thetaDOT*Ctheta)+((Cchi/Ixx)*4.56)-((Schi*4.48)/Iyy)+(((Ixx-Izz)/Ixx)*(chiDOT-((thetaDOT*Stheta)*(thetaDOT*Ctheta)))))

phiDDOT = 1.9035e+04

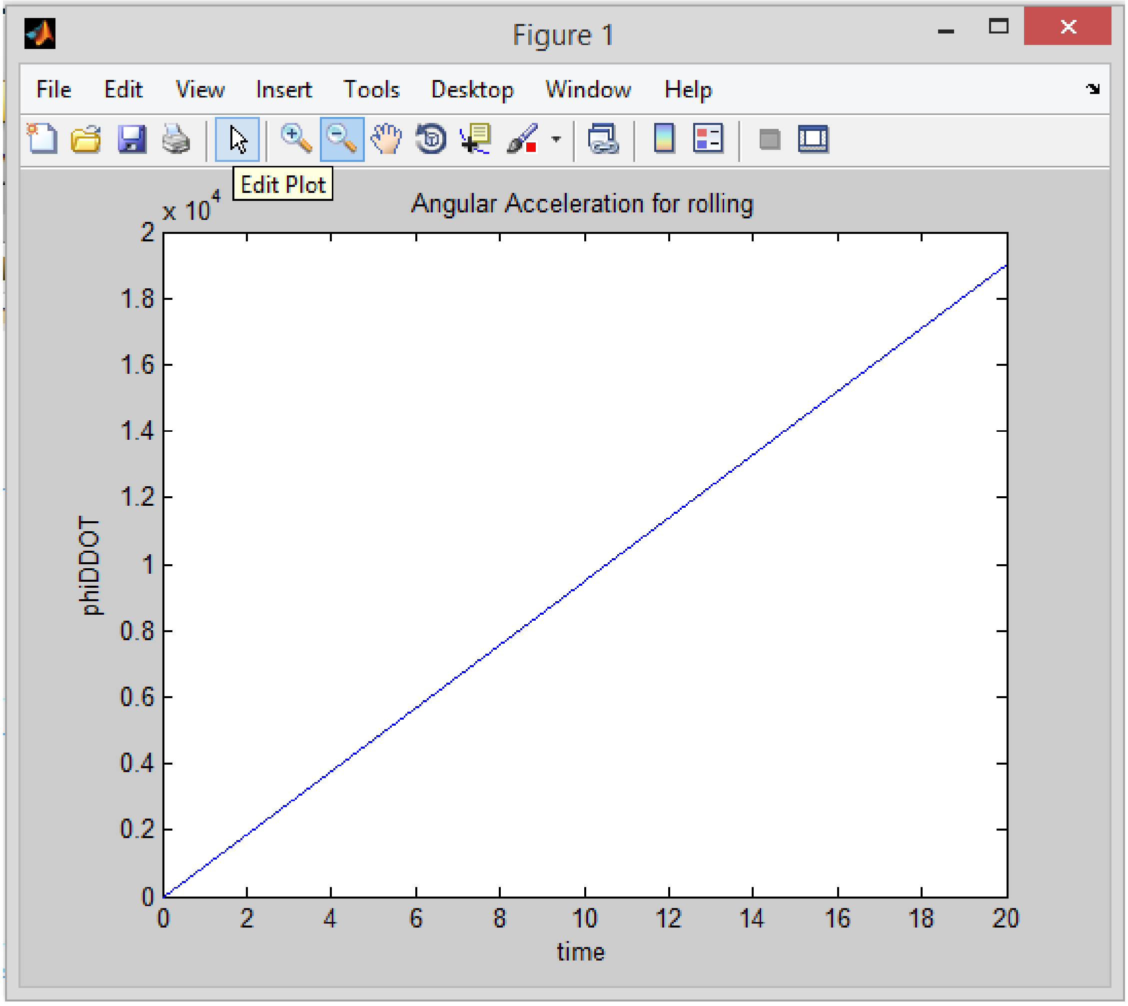


Fig 7.2(D) Angular acceleration for rolling vs time

>>xDDOT=(((Schi*Sphi)+(Cphi*Stheta*Cphi))*(5.03/0.687))

xDDOT =7.8371

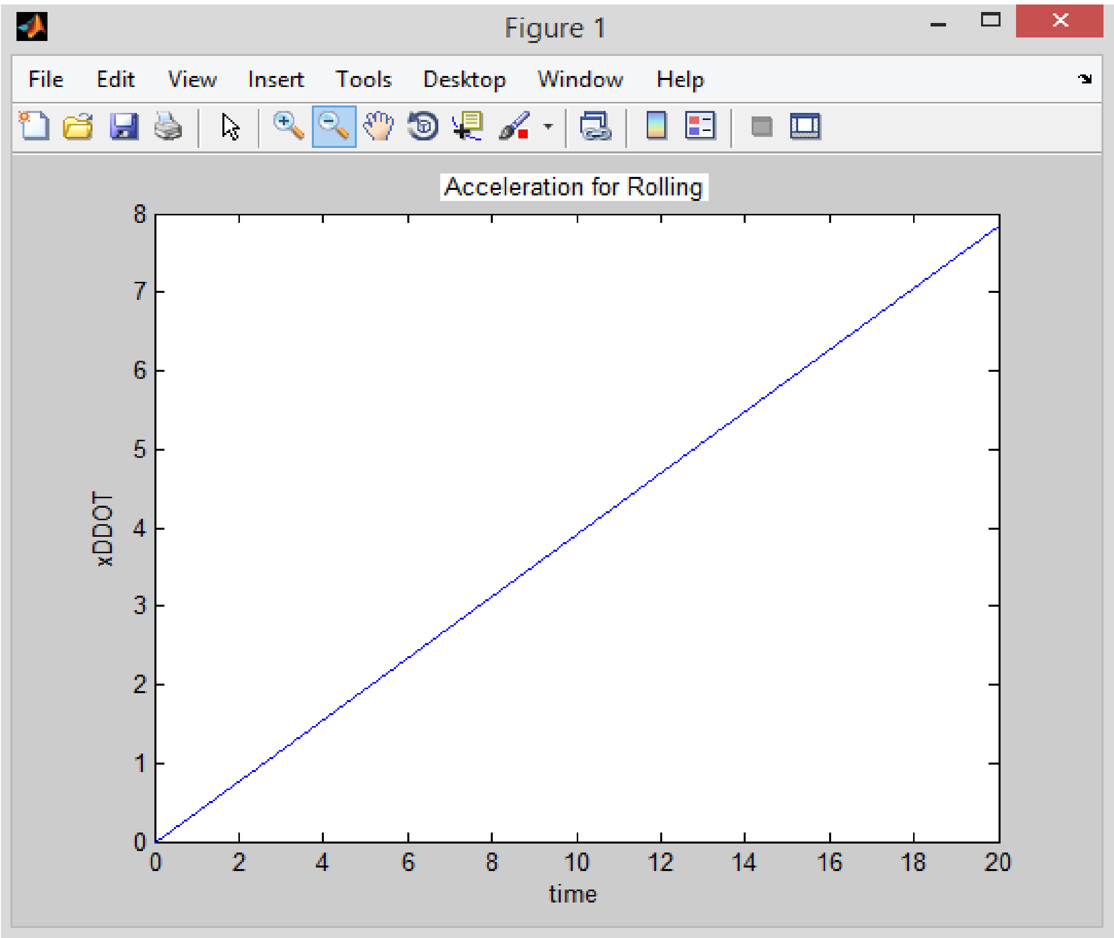


Fig 7.2(E) Acceleration for rolling vs time

>>yDDOT=(((-Cchi*Sphi)+(Sphi*Stheta*Cphi))*(5.03/0.687))

yDDOT =0.0419

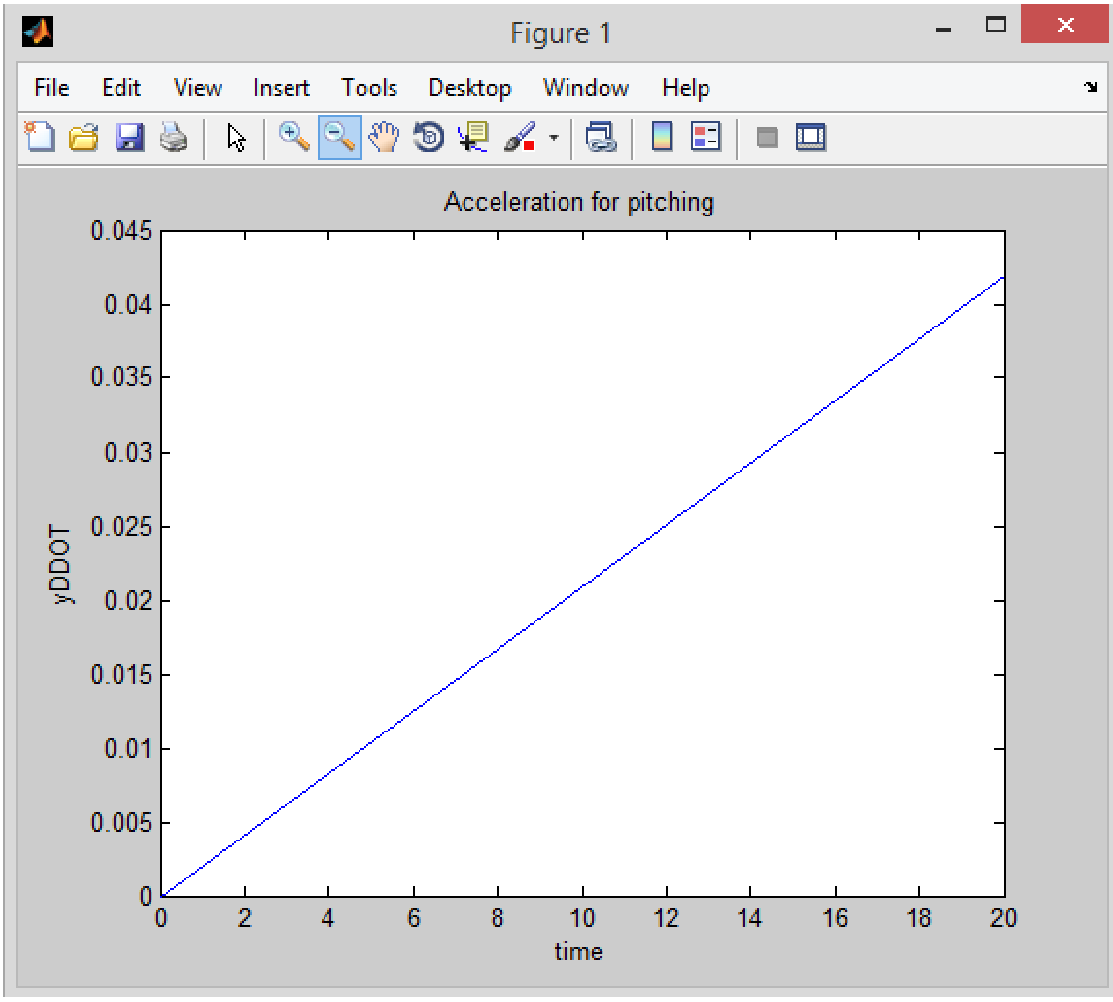


Fig 7.2(F) Acceleration for pitching vs time

>>zDDOT=(-9.81+((Stheta*Cphi)*(5.03/0.687)))

zDDOT =-2.8693

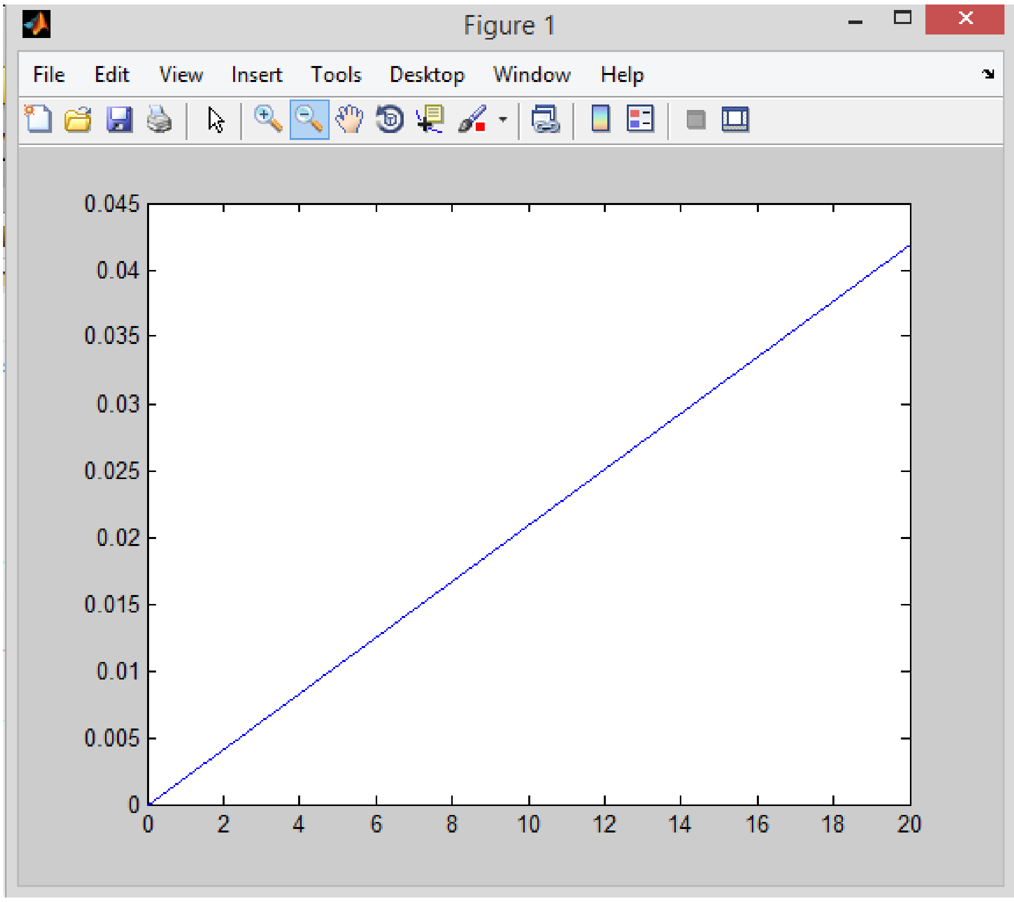


Fig 7.2(G) Acceleration for yawing vs time

>> acc1=[0 thiDDOT]

Undefined function or variable 'thiDDOT'.

>> acc1=[0 thetaDDOT]

acc1 =1.0e+03 *0 7.3993

>> t=[0 200]

t =0 200

>>plot(t,acc1)

>> acc2 = [0 phiDDOT]

acc2 = 1.0e+04 * 0 1.9035

>> acc3=[0 chiDDOT]

acc3 =1.0e+03 *0   4.4406

>>plot(t,acc3)

>> acc4=[0 xDD0T]

Undefined function or variable 'xDD0T'.

Did you mean:

>> acc4=[0 xDDOT]

acc4 =0   7.8371

>> acc5=[0 yDDOT]

acc5 =0   0.0419

>> acc6=[0 zDDOT]

acc6 =0   -2.8693

>> vel1= [0 thetaDOT]

vel1 =0   -5.0363

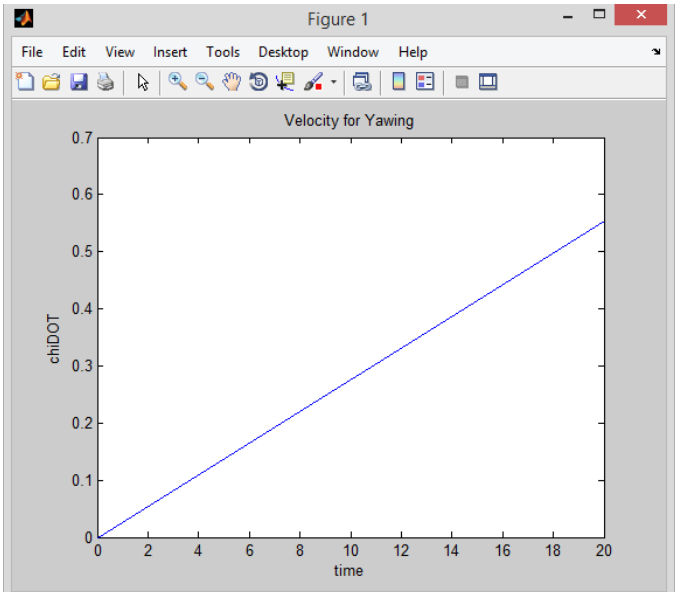


Fig 7.2(H) Velocity for yawing vs time

>> vel2= [0 phiDOT]

vel2 =0   3.4493

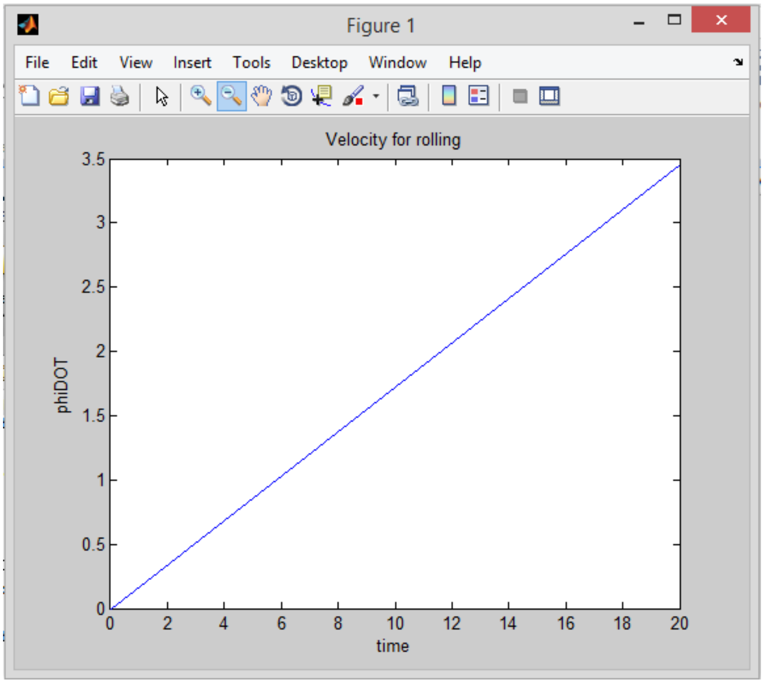


Fig 7.2(I) Velocity for rolling vs time

>> vel3= [0chiDOT]vel3 = 0   0.5544

7.3 SIMULINK Model analysis using Forward kinematics Method:-

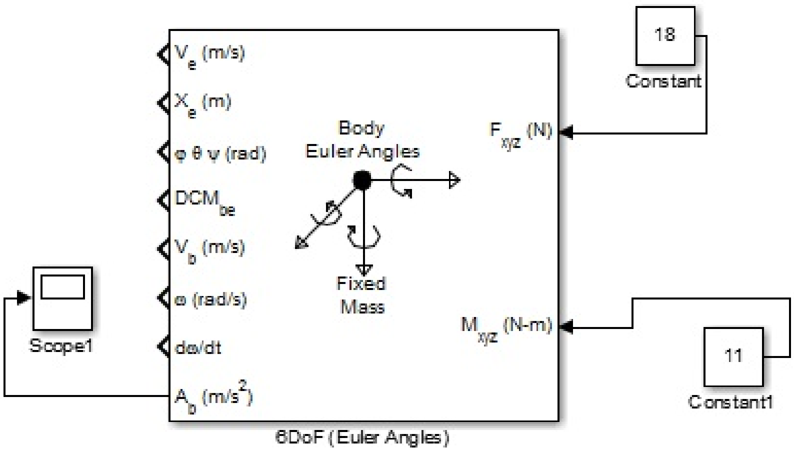


Fig 7.3(A) Body Acceleration analysis

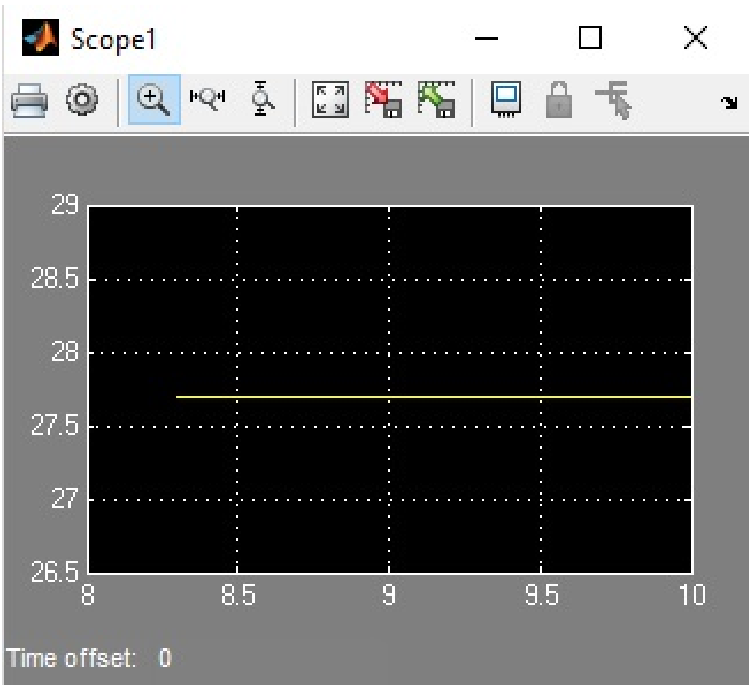
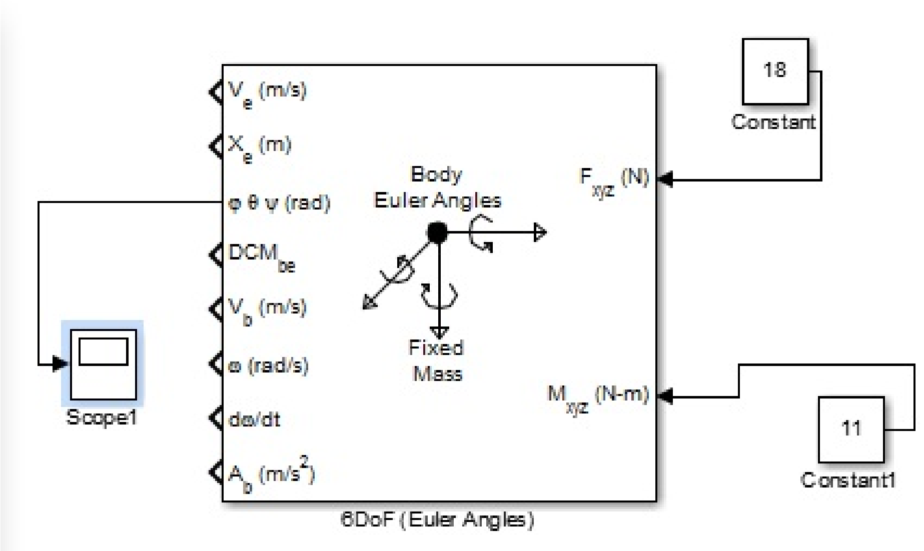
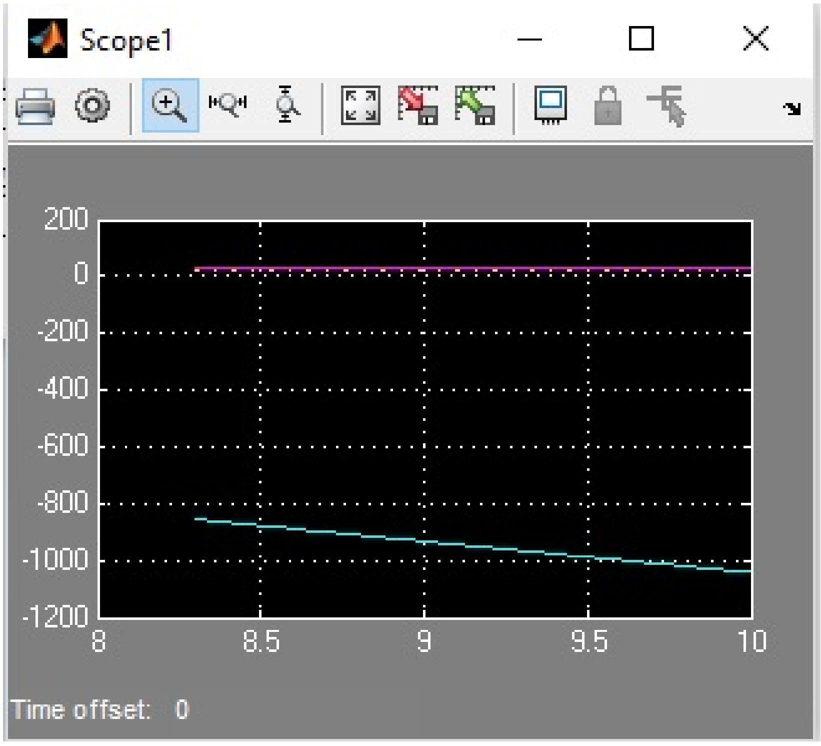


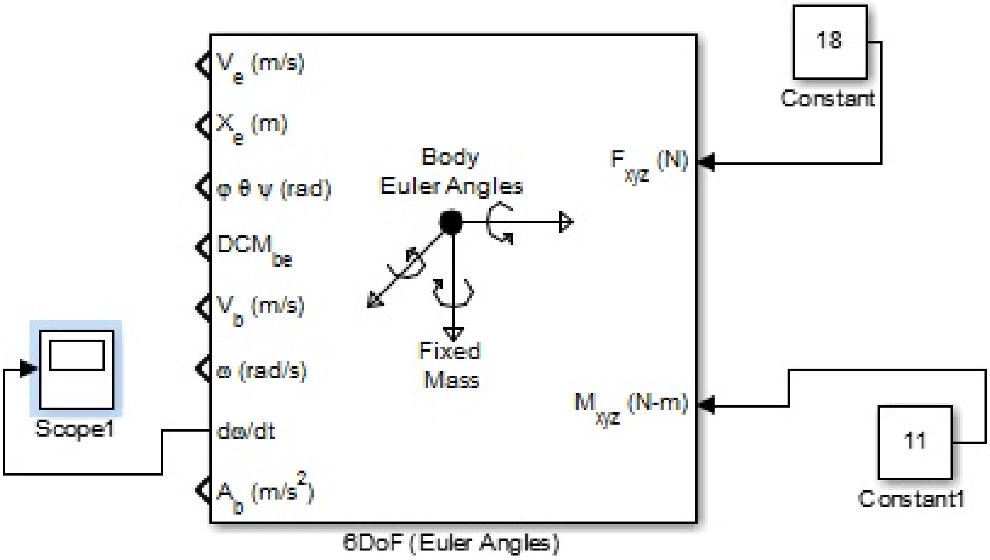
Fig 7.3(B) Output of Body Acceleration analysis



7.3(C) Angular positions by forward kinematics



7.3(D) Output of Angular positions by forward kinematics



7.3(E) Angular momentum analysis:-

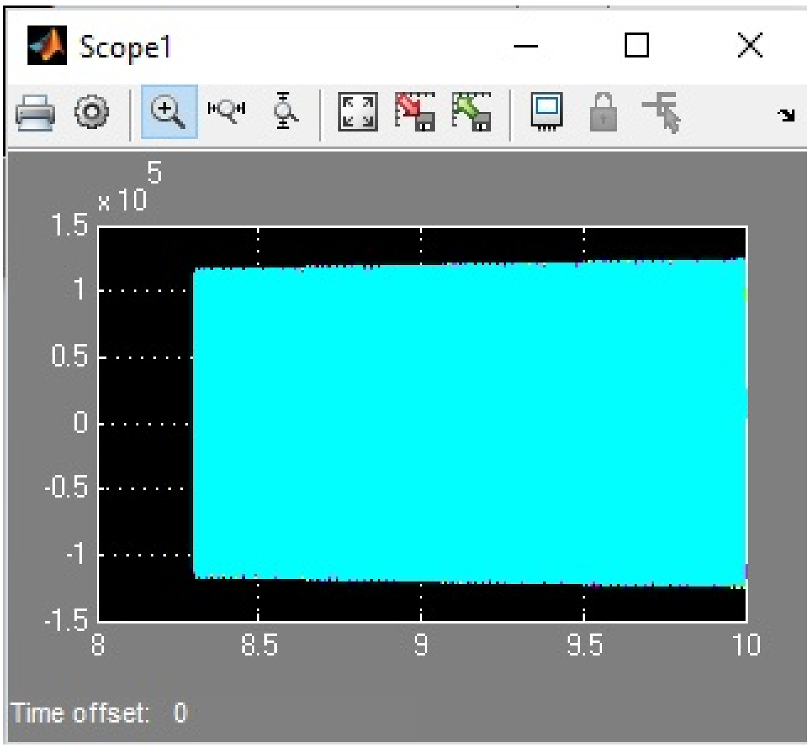


Fig 7.3(F) Output of Angular momentum analysis:-

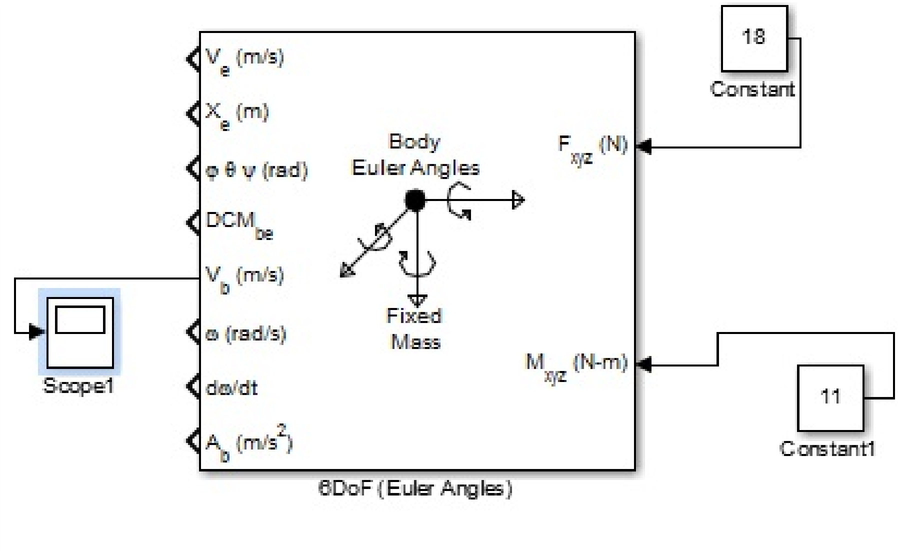


Fig 7.3(G) Body-frame velocity analysis:-

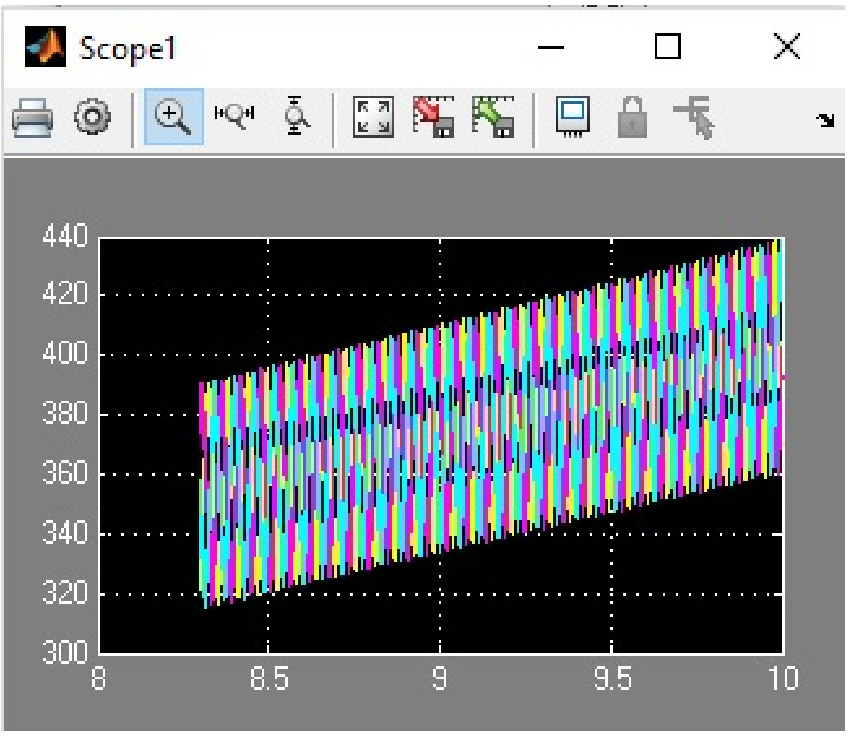


Fig 7.3(H) Output of Body-frame velocity analysis:-

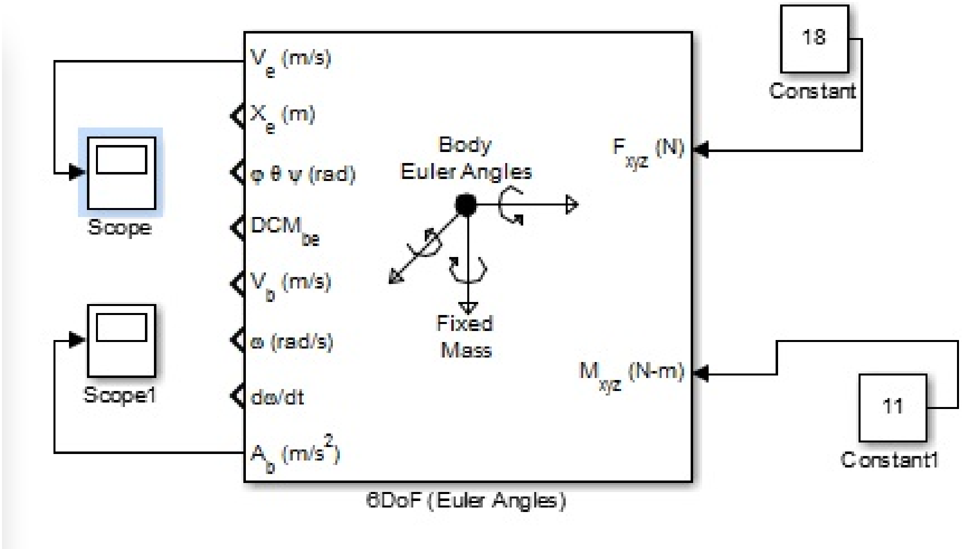


Fig 7.3(I) Earth-frame velocity analysis:-

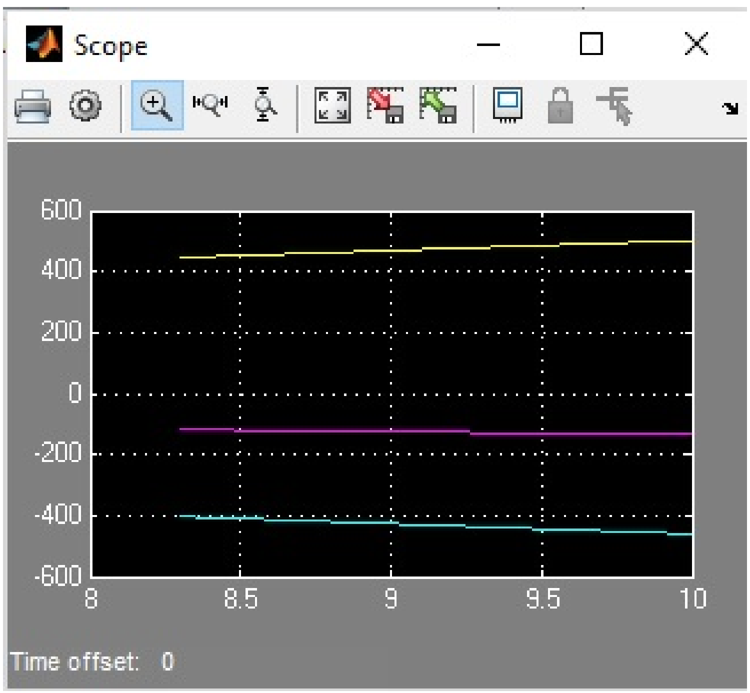


Fig 7.3 (J) Output of Earth-frame velocity analysis:-

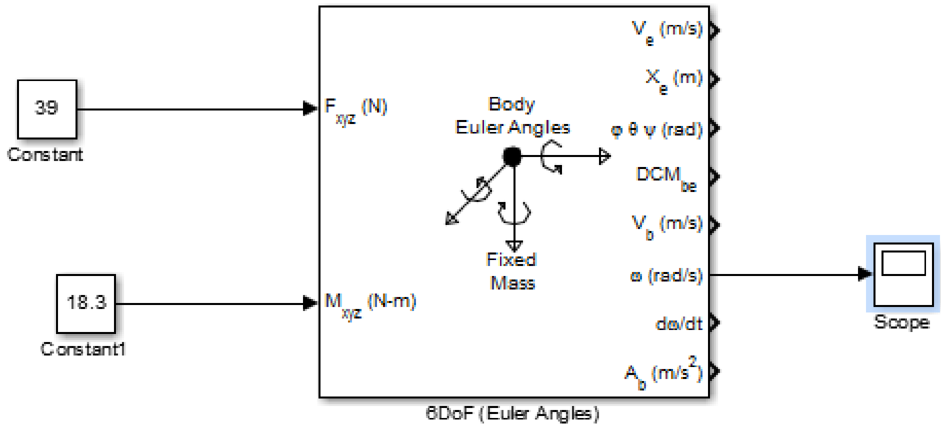


Fig 7.3(K) Angular positions analysis for pitching rolling and yawing respectively

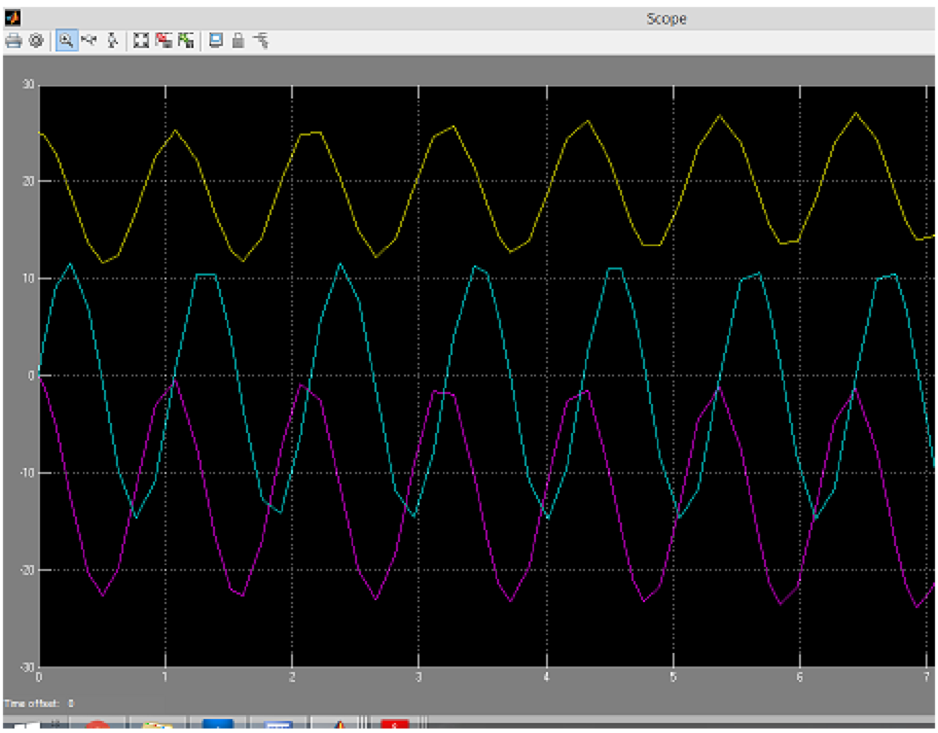


Fig 7.3(L) Output of Angular positions analysis for pitching rolling and yawing respectively:-

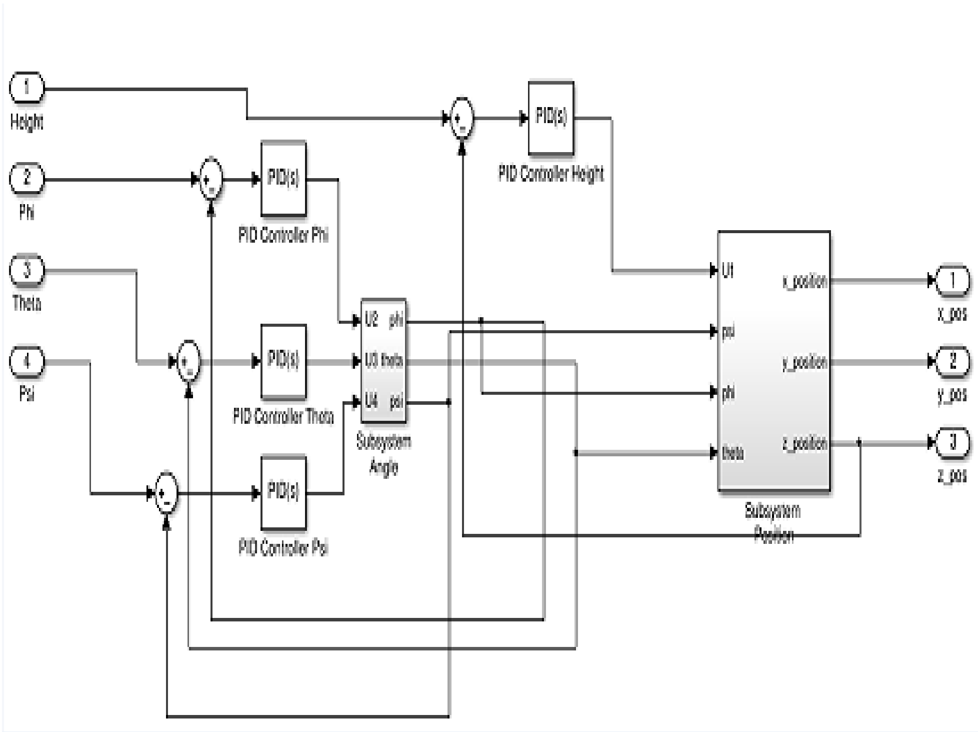


***Fig 7.3(M) SIMULINK* Model created by Forward Kinematics Method**

7.4 PID Stability Analysis by using different plant model analysis:-

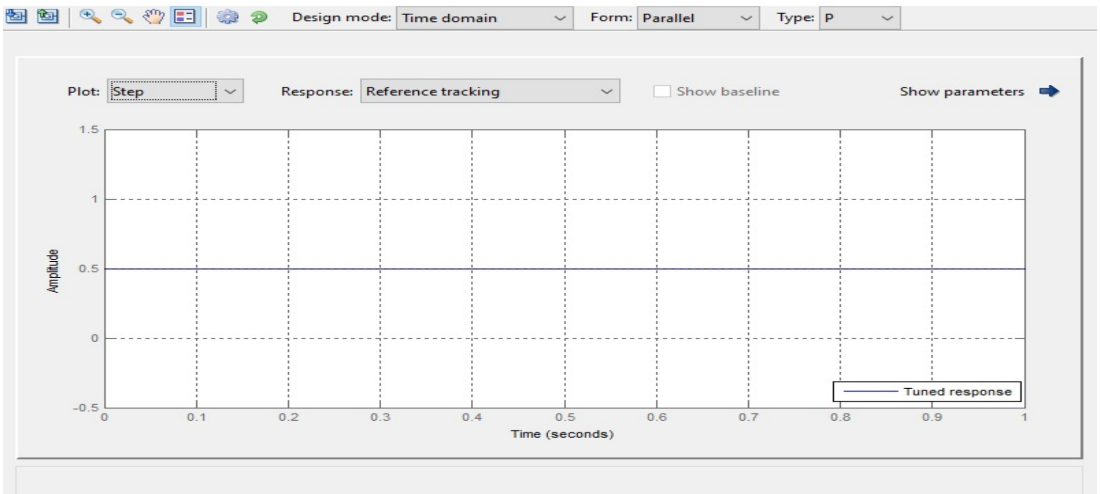


Fig 7.4(A) PID plant tuner in MATLAB13:-

Using different PID parameters for tuning:-

sample time=10 seconds

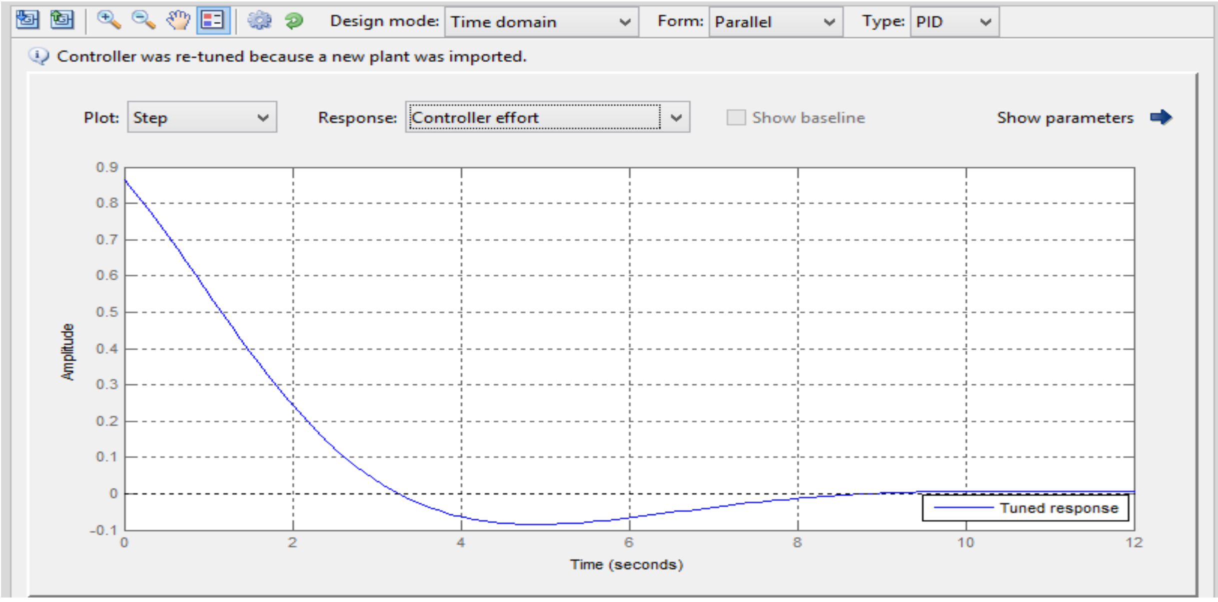


Fig 7.4(B) Controller effect response:-

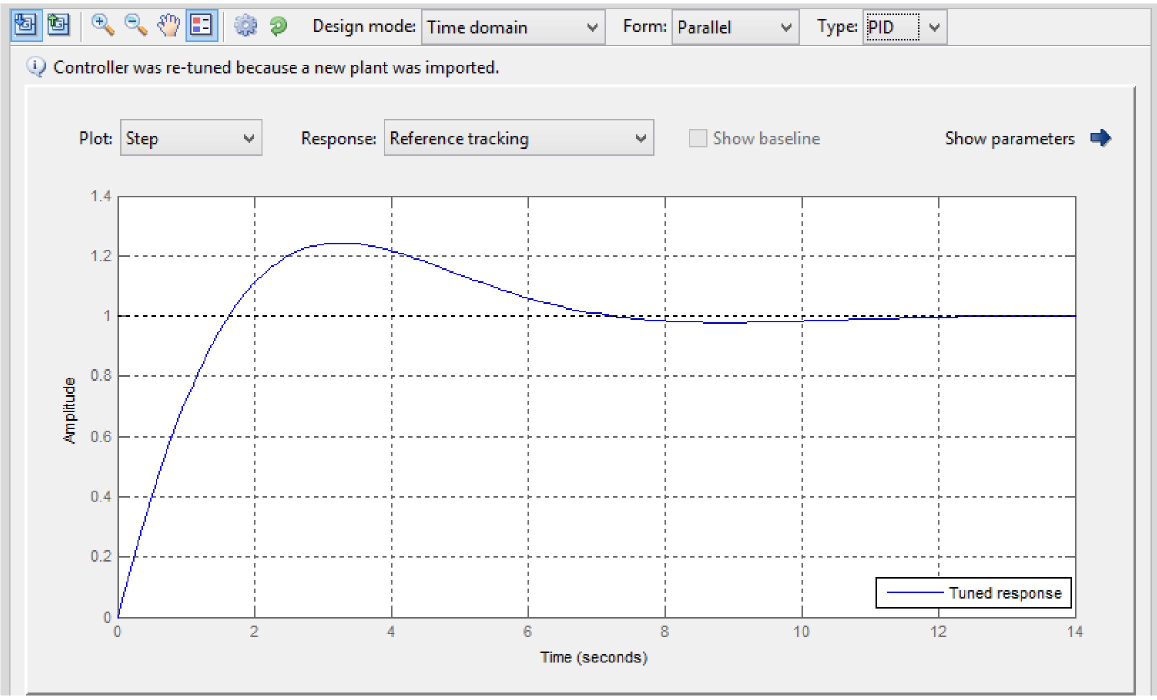


Fig 7.4(C) Reference Tracking response:-

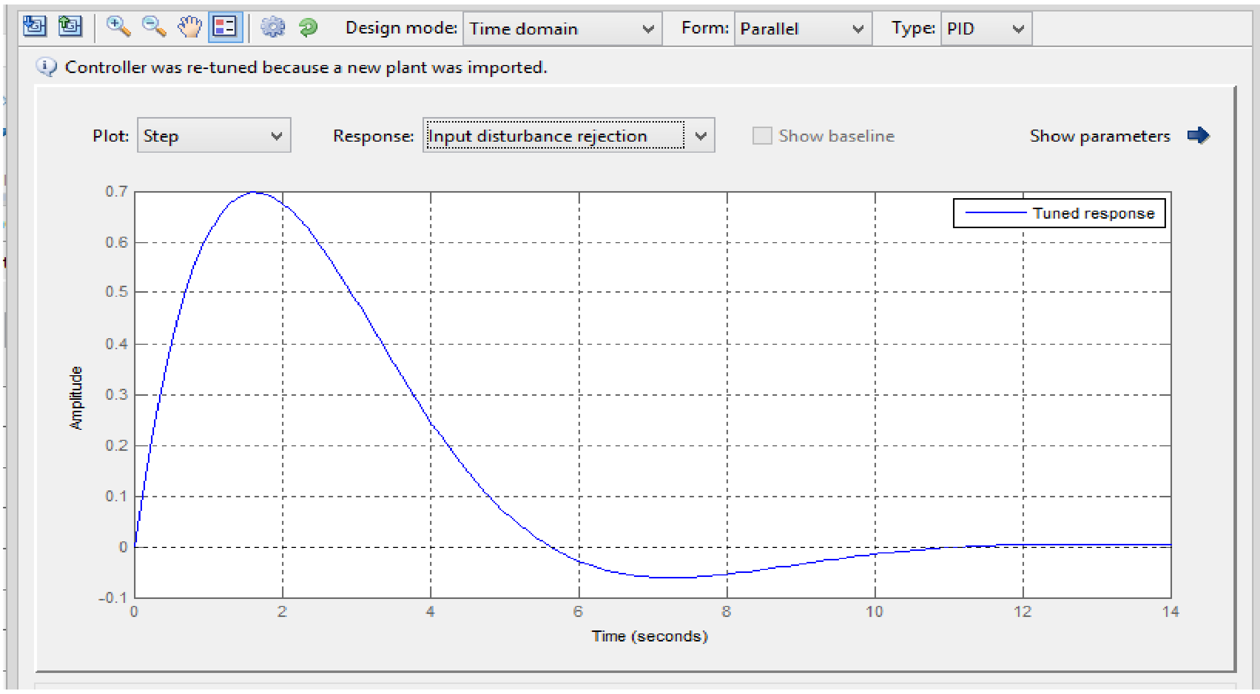


Fig 7.4(D) Input disturbance rejection response:-

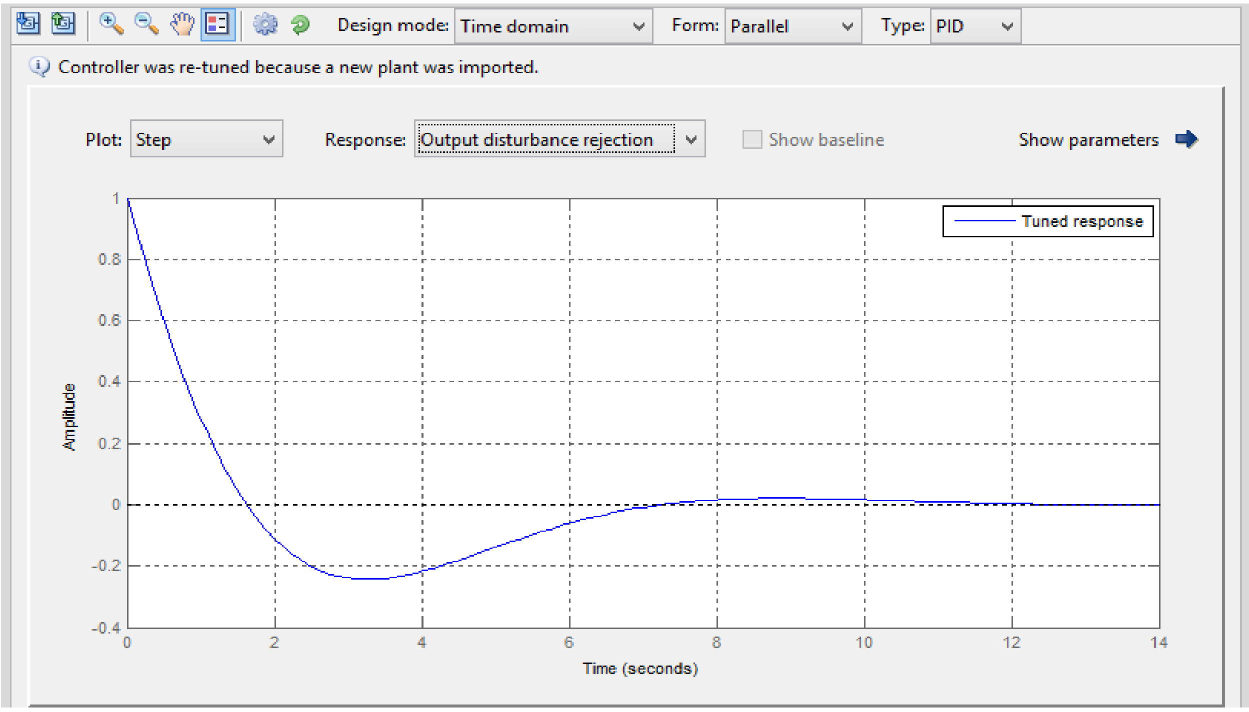


Fig 7.4(E) Output disturbance rejection response:-

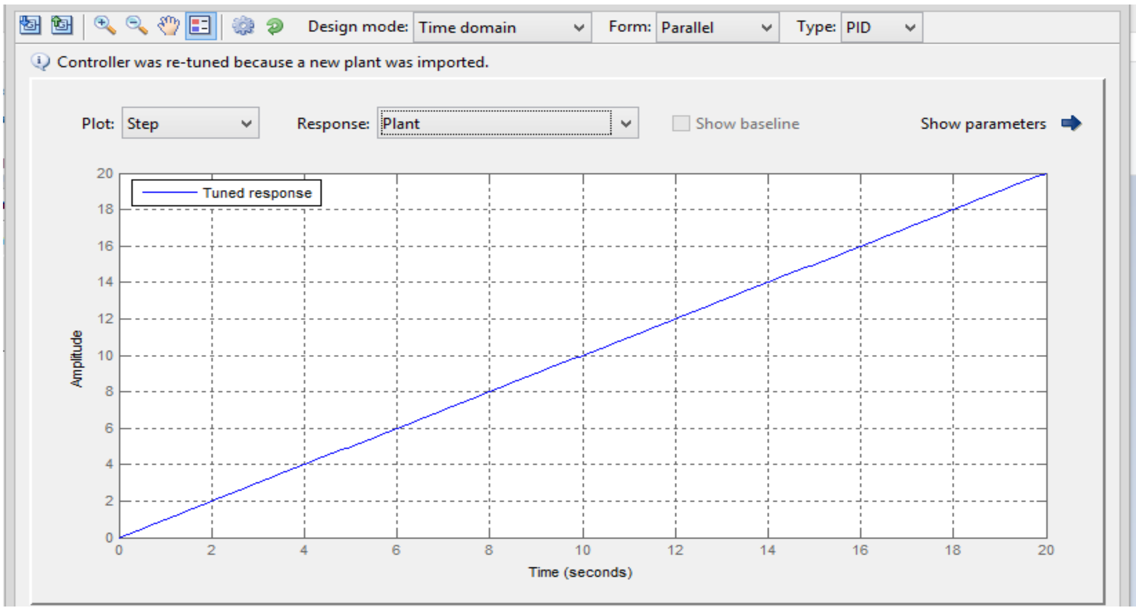
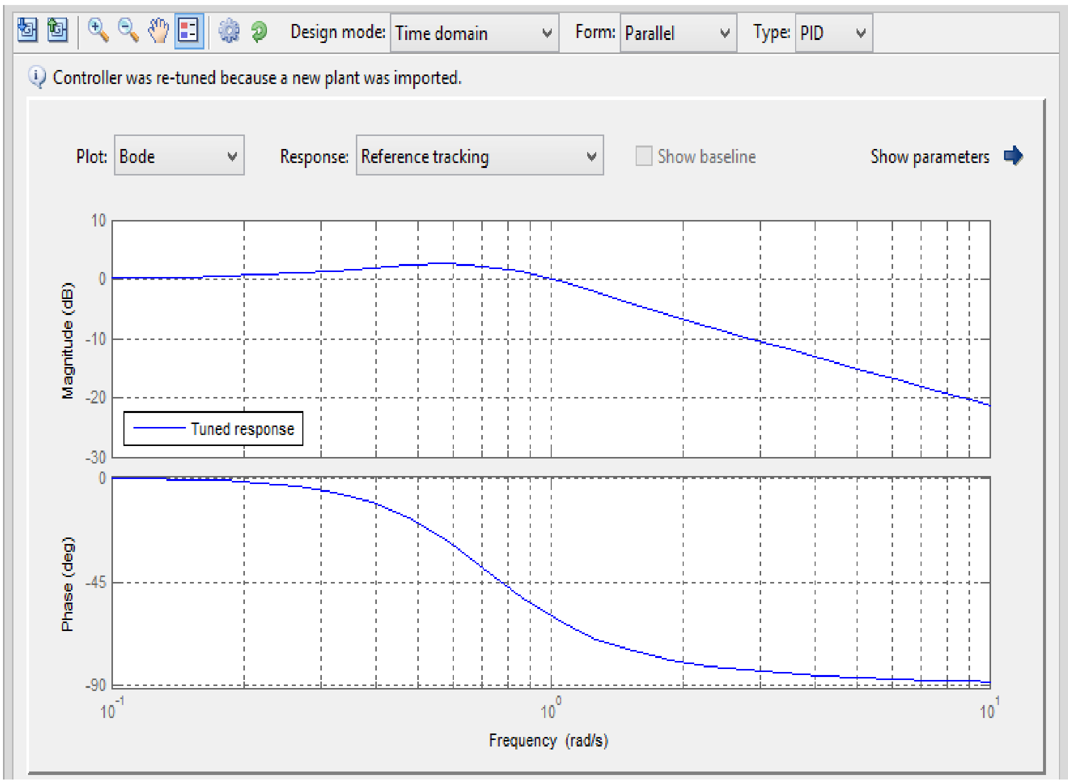


Fig 7.4(F) Plant response method



3

Fig 7.4(G) Stability using bode plot stability analysis

Conclusion

And

Future Scope

CONCLUSION

Validation accounts for feasibility of the work criteria set for the project and expected outcomes of the project our measure of conclusion of:-

- The experimentation and computational results almost can be equalized.
- Since because the computational platform is in real time analysis and multi scale modeling done on MATLAB ra13 simulation platform the results can be easily aggregated to the actual ones.
- The tuning of PID control system through automatic tuning system being done by analyzing and optimizing gain parameters may show a better version of results as far as stabilizing and precise operation and motion of a quad copter is concerned on a given trajectory.
- Usage of B-frame analysis and "exact differential equations" for implementation of kinematic and dynamic modeling ensures the interpolation of solutions for the equations rather than approximation.
- Developing a SIMULINK* model ensures interpretation of results and control inputs and improves feasibility for changing output variables with varying inputs instead referring the entire algorithm.

FUTURE SCOPE

The algorithm can be employed as is, or built upon We are currently seeking to extend the planning algorithm with these improvements to further improve the flight performance. The implementation and experimental validation of the method as presented herein is an important step towards this.

To improve this quad rotor project, a more accurate model of the helicopter can be studied, in particular aerodynamic considerations can help in non hovering operation. Together with this research, the identification of the real platform physics must be much more accurate. Several control algorithms can be investigated to find the best trade-off between performance and software complexity. A lot of articles which focus on quad rotor stabilization algorithm have been already written. However it would be great to compare them and find better solutions. Even though the simulator showed already good accuracy and testability, it would be great to be able to simulate the environment too and to use tools which interact with the real platform.

- Algorithm can be further be optimized and used in tandem with LQR method.
- Design and develop various *SIMULNK* patterns for interfacing various platform of CAD models.
- The results of the simulation model can be used as a reference input for:-
  - analysis model without centre at cog
  - For effect of rotor dynamics of rotor on the control of torque and speed of each motor
  - Reference model , architecture & structure for designing a PID and optimizing gain parameter

References

[1] K. Munson. (1968). *Helicopters and Other Rotorcraft Since 1907*

[2] Gabriel Hoffmann. (2007, January 15) *Schematic of reaction torques on each motor of a quad rotor aircraft, due to spinning rotors*

[3] FPVUK (2012) *How do I go about it?*

[4] Simon Size. Chapter 13.6 in *Physics of semiconductor devices* (3 ed.) John Wiley and Sons, 2007.

[5] Lorextechnology.com (2012) *Understanding Digital and Analog Wireless Technology*

[6] Rockwell Semiconductor Systems (1997) *Single-Chip Video and Broadcast Audio Capture for the PCI Bus*

[7] Howard Shane. (2005, January 29) *The BTTV HOWTO*

[8] Wikipedia

Publication

[1] Syed Adnan, Aamir Padwekar, Altamash Shaikh, Rahul Thavai, "Kinematic and Dynamic Modeling of a Quad Copter following a Trajectory using MATLAB", "International Conference on Modern Trends in Engineering, Science and Technology", Vishwatmak Om Gurudev College of Engineering, 9^{th} – 10^{th} April 2016.