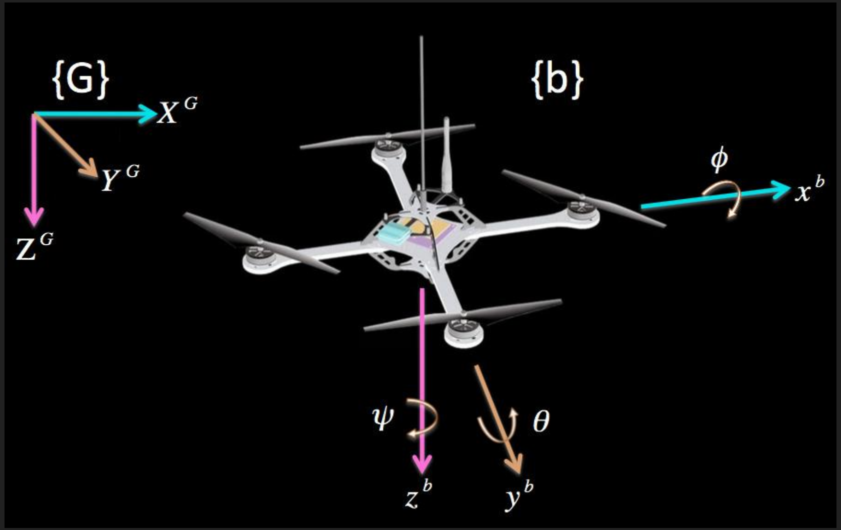
EE650A: Modern Control Systems

Quadcopter Attitude Control

Design and Implementation of the MIMO System

Aravind Seshadri 210180	Kapu Nirmal Joshua 210489	Sumay Avi 211071	Raghav Shukla 210800

Quadcopter Dynamics



Let there be a coordinate frame x-y-z.

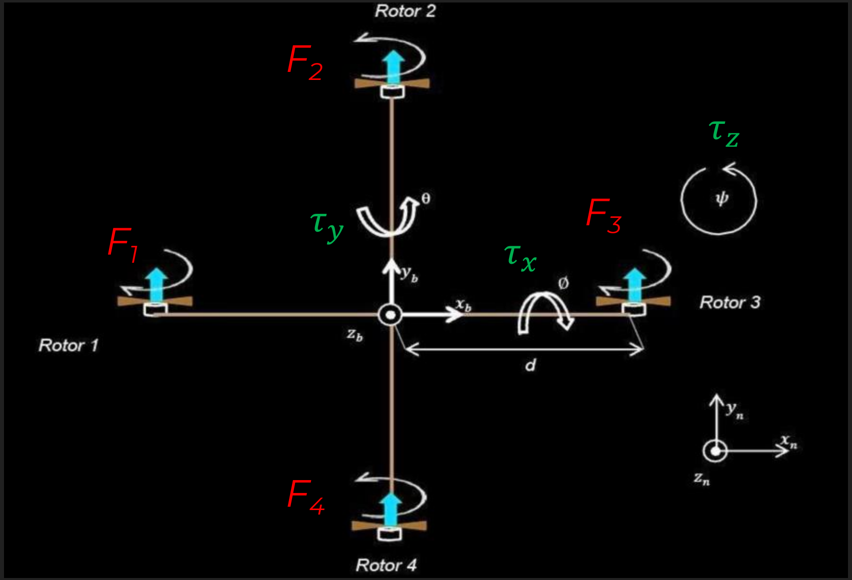
Our quadcopter model has 6 states –
Pitch, Roll, Yaw and their corresponding derivatives.
Thus we have 6 degrees of freedom

Our state variable X is given by:

$$\boldsymbol{X} = (\Phi, \theta, \Psi, d\Phi/dt, d\theta/dt, d\Psi/dt)$$

Φ = *Roll Angle*
θ = *Pitch Angle*
Ψ = *Yaw Angle*

Quadcopter Free Body Diagram



Forces:

Let F_i denote upward thrust at rotor i

The magnitude of the force can be controlled by varying the rpm of the rotor

Moments:

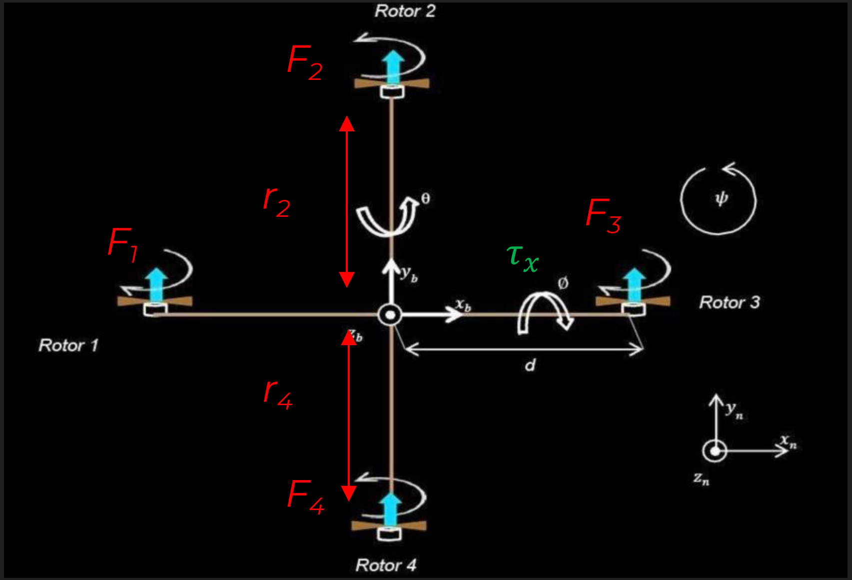
All forces contribute to moments about suitable axes.

In particular, we consider moments about x, y and z axes. We denote them by τ_x, τ_y, τ_z

Center Of Gravity

The center of gravity lies almost at the same plane which contains all the rotors

Torque Analysis of Quadcopter



Roll Motion:

This is the rotatory motion about the x axis.

We calculate the torque τ_x using the following moment balancing equation:

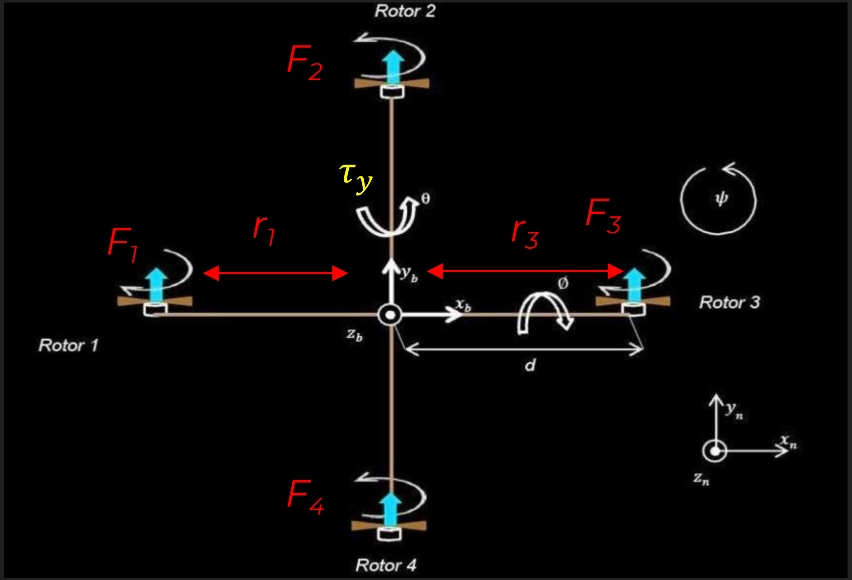
$$\tau_x = \sum_{i=1}^{4} r_i F_i$$

Observe that the thrusts F_1 and F_3 are located on the x axis. Their net contribution is 0.

$$\tau_x = r_2 F_2 - r_4 F_4$$

$$\tau_x = dF_2 - dF_4 \; is\ the\ final\ equation$$

Torque Analysis of Quadcopter



Pitch Motion:

This is the rotatory motion about the y axis.

We calculate the torque τ_y using the following moment balancing equation:

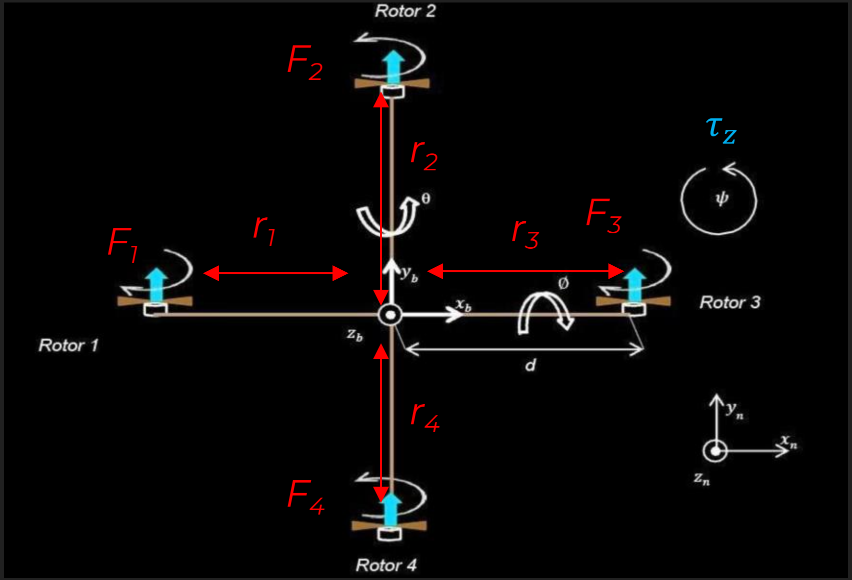
$$\tau_y = \sum_{i=1}^{4} r_i F_i$$

Observe that the thrusts F_2 and F_3 are located on the y axis.
Their net contribution is 0.

$$\tau_x = r_1 F_1 - r_3 F_3$$

$\tau_x = dF_1 - dF_3$ *is the final equation*

Torque Analysis of Quadcopter



Yaw Motion:

This is the rotatory motion about the z axis.

We calculate the torque τ_z using the following moment balancing equation:

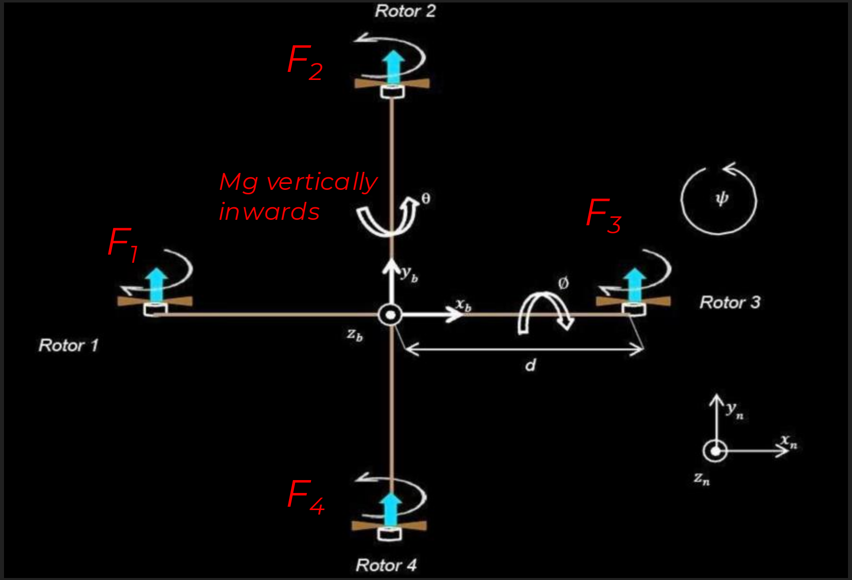
$$\tau_z = \sum_{i=1}^{4} r_i F_i$$

C here is the force to moment scaling factor

The yaw is accomplished by rotor rpm imbalance between clockwise and counter-clockwise rotating propellers.

$\tau_z = -cF_1 + cF_2 - cF_3 + cF_4$ *is the final equation*

Vertical Motion Analysis



This is the linear motion along the z axis.

The vertical motion is governed by the following:

***Thrust force** due to each rotor*

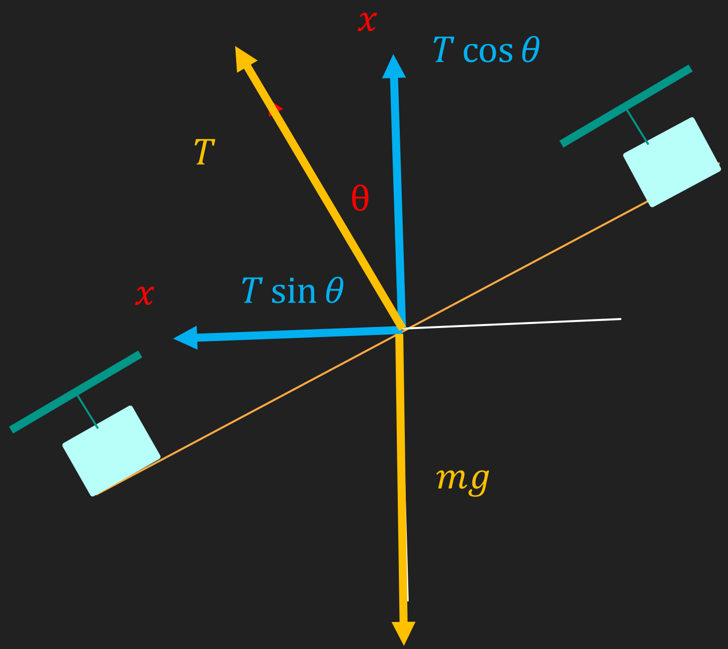
$$\boldsymbol{T} = \sum_{i=1}^{4} \boldsymbol{F}_i$$

Weight of the quad copter: ***mg***

The resulting differential equation:

$$\ddot{\mathbf{z}} = \frac{T - mg}{m}$$

Horizontal Motion Analysis



This is the linear motion along the x-y plane.

The horizontal motion is governed by the following:

The quad copter tilts in the direction of the slow spinning rotor.

There are 2 assumptions:
Theta is very small so $sin\theta = \theta$
T is equal to mg to avoid up and down movement

The resulting differential equation along x and y are as follows:

$$\ddot{x} = -\frac{Tsin\theta}{m} = -g\theta$$

$$\ddot{y} = -\frac{Tsin\phi}{m} = -g\phi$$

State Space Equations

States of quadcopter:

$$\boldsymbol{roll\,angle}: \ \boldsymbol{\phi}$$
$$\boldsymbol{pitch\,angle}: \ \boldsymbol{\theta}$$
$$\boldsymbol{yaw\ angle}: \boldsymbol{\psi}$$

$$\boldsymbol{roll\,rate}: \ \boldsymbol{\phi'}$$
$$\boldsymbol{pitch\,rate}: \ \boldsymbol{\theta'}$$
$$\boldsymbol{yaw\ rate}: \boldsymbol{\psi'}$$

Vectors:

$$\boldsymbol{x^T} = [\boldsymbol{\phi, \theta, \psi, \phi', \theta', \psi'}]$$
$$\boldsymbol{u^T} = [\boldsymbol{F_{1,} F_{2,} F_{3,} F_4}]$$
$$\boldsymbol{y^T} = [\boldsymbol{\phi, \theta, \psi}]$$

Equations:

$$\boldsymbol{\phi'} = \boldsymbol{\phi'}$$
$$\boldsymbol{\theta'} = \boldsymbol{\theta'}$$
$$\boldsymbol{\psi'} = \boldsymbol{\psi'}$$

$$\boldsymbol{\phi''} = \frac{dF_2 - dF_4}{I_x}$$

$$\boldsymbol{\theta''} = \frac{dF_1 - dF_3}{I_y}$$

$$\boldsymbol{\psi''} = \frac{-cF_1 + cF_2 - cF_3 + cF_4}{I_z}$$

State Space Equations

$$
\overset{x'}{\begin{bmatrix} \varphi' \\ \theta' \\ \psi' \\ \varphi'' \\ \theta'' \\ \psi'' \end{bmatrix}}
=
\overset{A}{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}
\overset{x}{\begin{bmatrix} \varphi \\ \theta \\ \psi \\ \varphi' \\ \theta' \\ \psi' \end{bmatrix}}
+
\overset{B}{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{d}{I_x} & 0 & -\frac{d}{I_x} \\ \frac{d}{I_y} & 0 & -\frac{d}{I_y} & 0 \\ -\frac{c}{I_z} & \frac{c}{I_z} & -\frac{c}{I_z} & \frac{c}{I_z} \end{bmatrix}}
\overset{u}{\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}}
$$

$$
\overset{y}{\begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}}
=
\overset{C}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}}
\overset{x}{\begin{bmatrix} \varphi \\ \theta \\ \psi \\ \varphi' \\ \theta' \\ \psi' \end{bmatrix}}
+
\overset{D}{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}
\overset{u}{\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}}
$$

$$
A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\qquad
B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{d}{I_x} & 0 & -\frac{d}{I_x} \\ \frac{d}{I_y} & 0 & -\frac{d}{I_y} & 0 \\ -\frac{c}{I_z} & \frac{c}{I_z} & -\frac{c}{I_z} & \frac{c}{I_z} \end{bmatrix}
$$

$$
C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}
\qquad
D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
$$

Objective of Our Controller

Open Loop System

Unstable Output

The Open Loop formulation of our system results in it being unstable.

However, we notice that *rank(C) = rank(O) = 6*

Therefore we have full rank matrices.

<u>Thus, we have a controllable and observable system!</u>

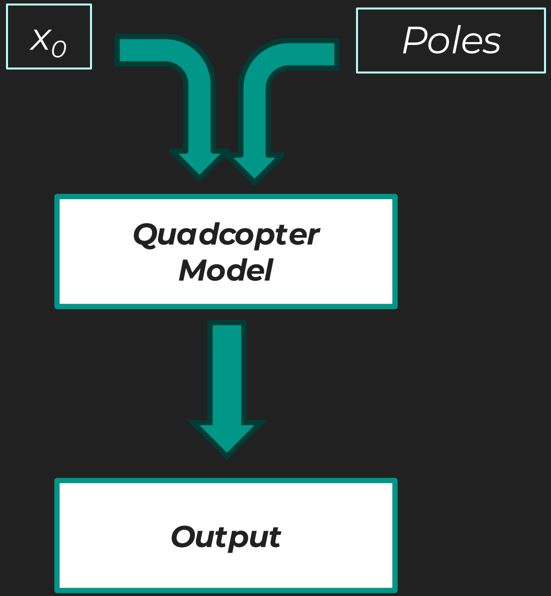
[C = controllability matrix, O = observability matrix]

Therefore we can design a closed loop state feedback controller to stabilize our system (with an appropriate observer model as well)

Closed Loop System

Stable Output

System Specifications



Let x_0 denote our initial state and P denote our set of poles

$$x_0 = [\quad 0.01 \quad 0.02 \quad 0 \quad 0 \quad 0 \quad 0 \quad]^T$$

$$P = [\quad -4-3i \quad -4+3i \quad -20 \quad -30 \quad -40 \quad -50 \quad]^T$$

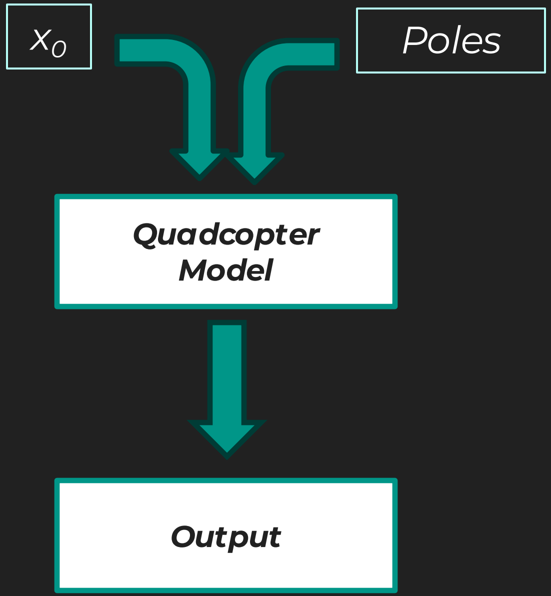
Recall that for our closed loop system:

$$A_{cl} = (A-BK)$$

We calculate K using MATLAB's "place" function which calculates it such that:

$$\boldsymbol{eigenvalues(A\text{-}BK) = P}$$

Justification for Controller Performance



We choose the following parameters for our system:

Settling Time (T_s) = 1 second

Damping Ratio (ζ) = 0.8

We require a tolerance band of *2%*

Therefore, $\boldsymbol{T_s = 4 / (\zeta * f_n)}$

[formula taught in EE250: Control Systems Analysis]

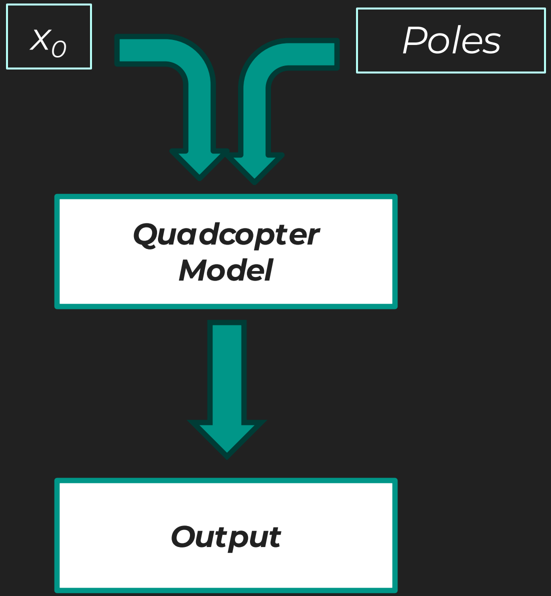
We use **f_n = 5Hz** [natural frequency]

Now we get our poles via the following quadratic:

$$s^2 + 2\zeta f_n s + f^2 = 0 \Rightarrow s^2 + 8s + 25 = 0$$

This gives us s = -4-3i and s = -4 + 3i

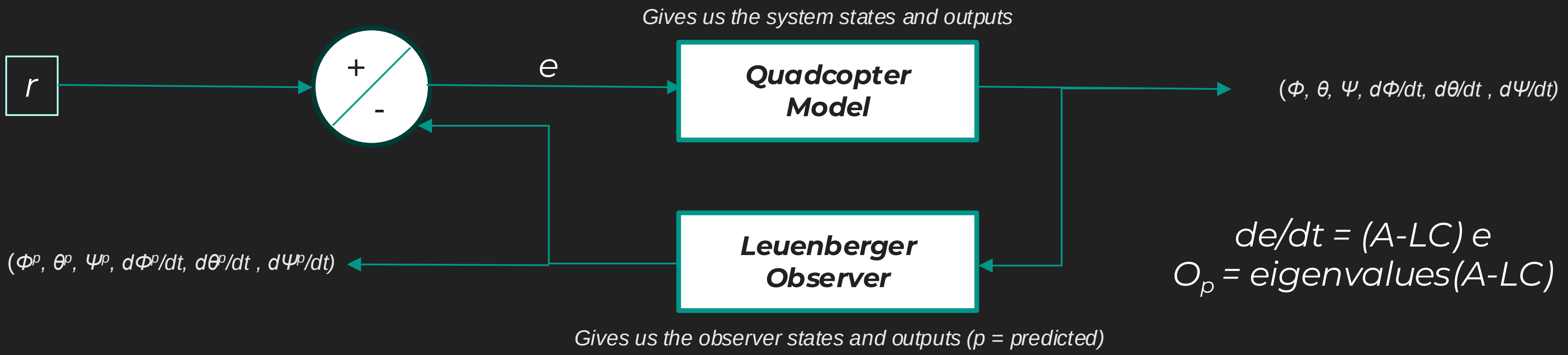
Justification for Controller Performance - Continued



To set the remaining 4 poles, we place them at a further distance, away from the dominant poles to minimize their effect.

Observation: In some of the outputs, we notice a high overshoot value. This can be reduced by increasing the magnitude of the dominant poles, however this would cause a tradeoff between the settling time and overshoot

Design and Justification – Leuenberger Observer



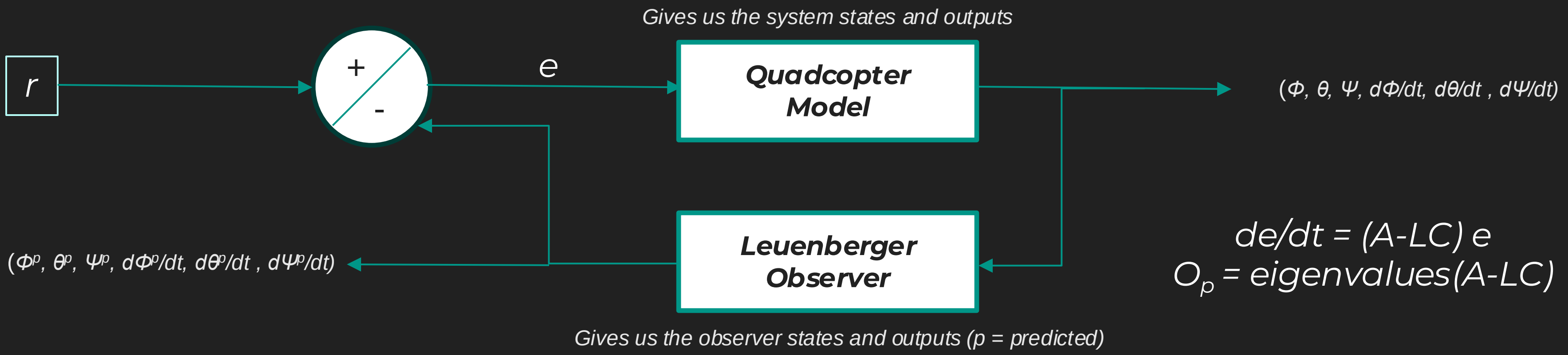
$$de/dt = (A-LC)\, e$$
$$O_p = eigenvalues(A-LC)$$

Our observer poles are chosen as $O_p = [\ -40 \quad -41 \quad -42 \quad -43 \quad -44 \quad -45\ ]$

The poles are chosen at a suitable distance from the axis, so that they have a larger magnitude compared to the dominant poles of the state feedback controller.

This will ensure that the estimated states track the actual states at a faster rate.

Design and Justification – Leuenberger Observer

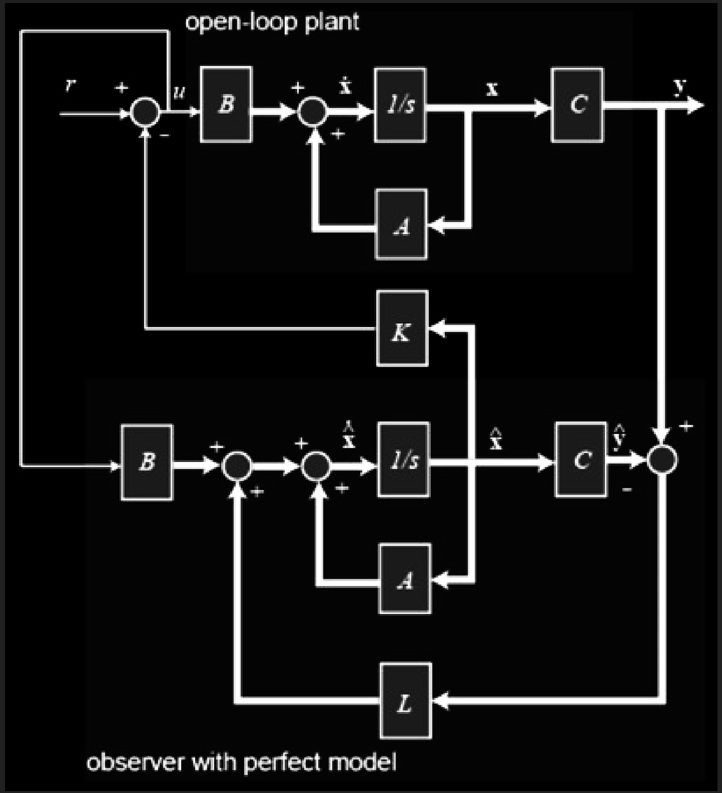


$$de/dt = (A-LC)\,e$$
$$O_p = eigenvalues(A-LC)$$

We notice that the observer successfully tracks the states of the system. However initially, there is a noticeable difference between both of them. This is due to the rapid changes that occur because of high frequency in the initial stages of the state feedback controller.

The observer is unable to track it properly initially, this can be improved by changing the location of our chosen poles.

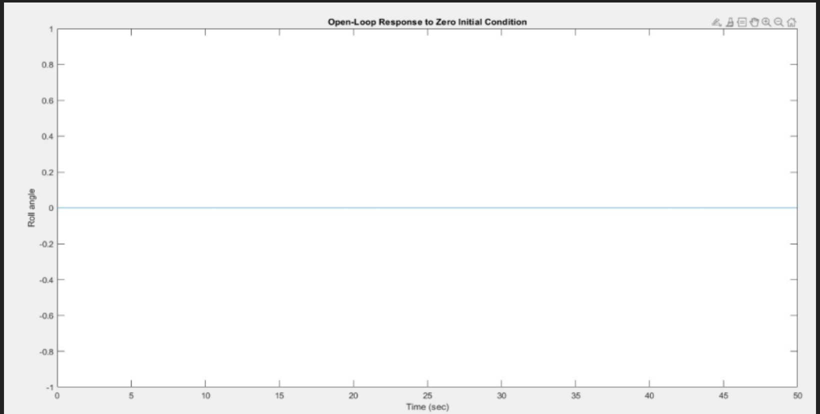
Final System

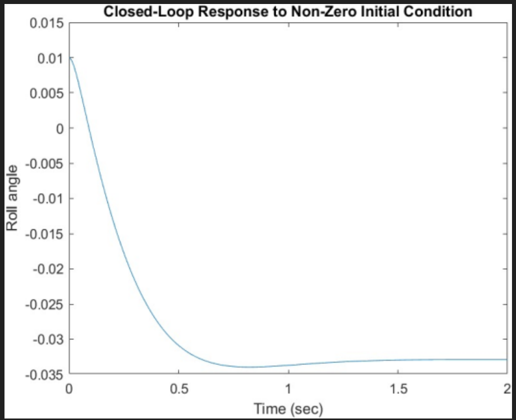


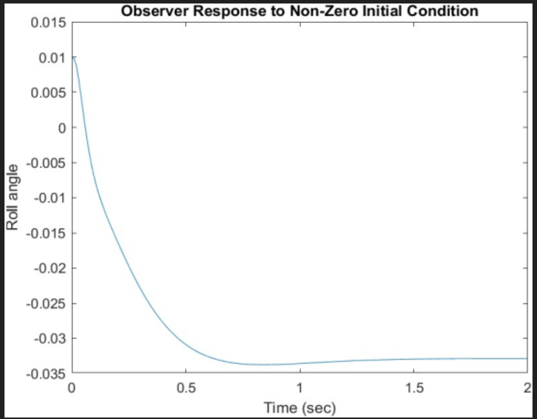
Quadcopter Model

Leuenberger Observer

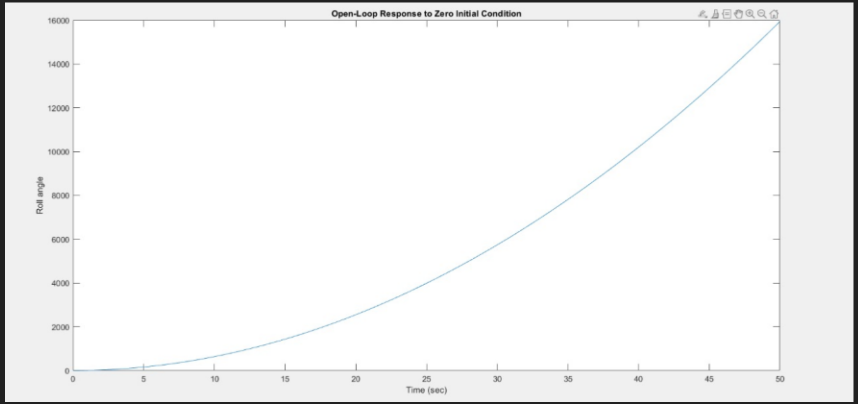
Results - (Plots of *Roll* (Φ) Output) - F_1 *alone is given step input*

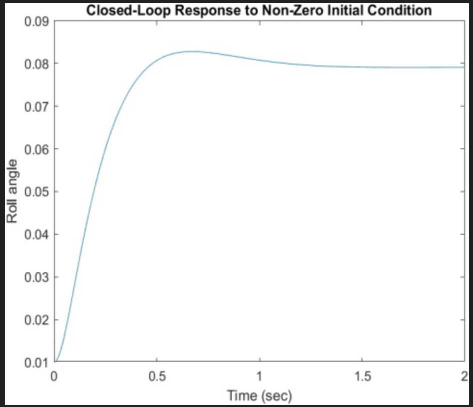


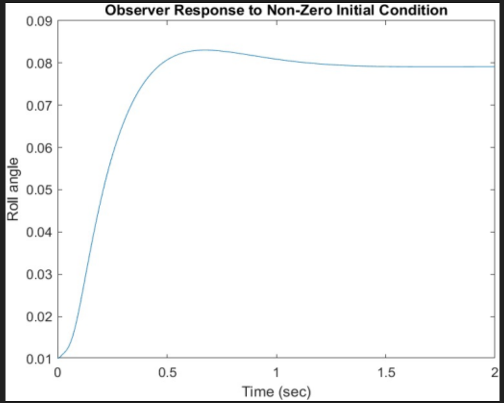




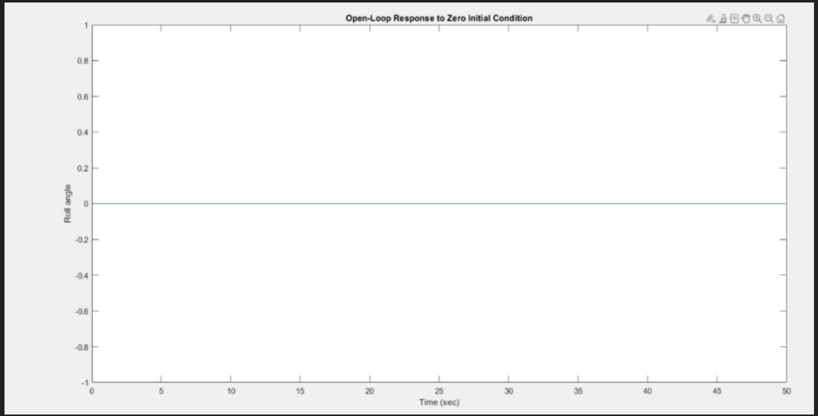
Results - (Plots of *Roll* (Φ) Output) - *F_2 alone is given step input*

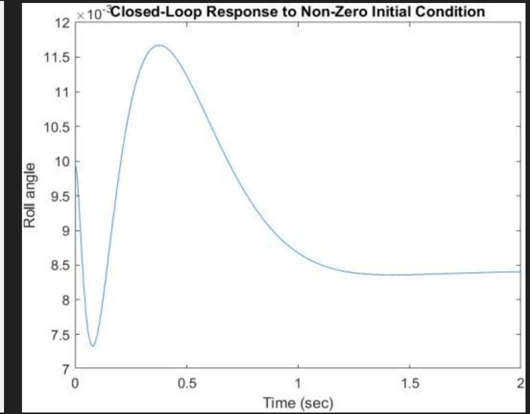


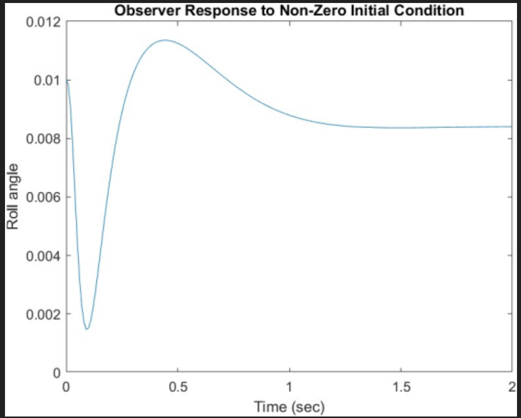




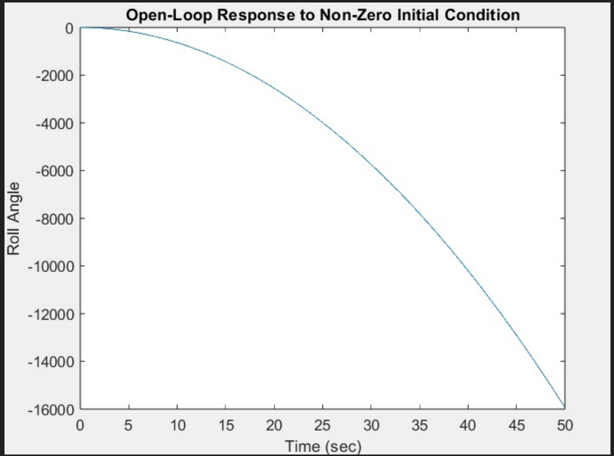
Results - (Plots of *Roll* (Φ) Output) - *F_3 alone is given step input*

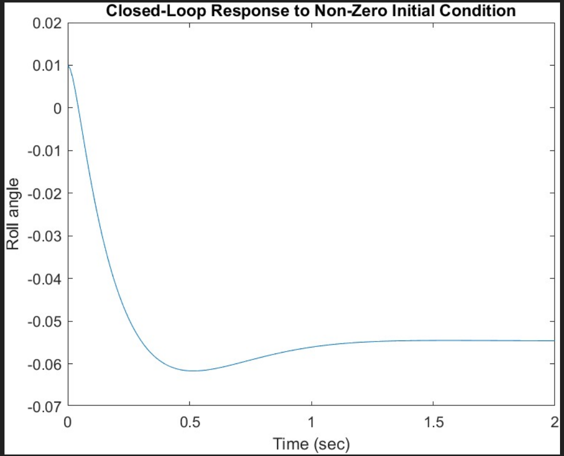


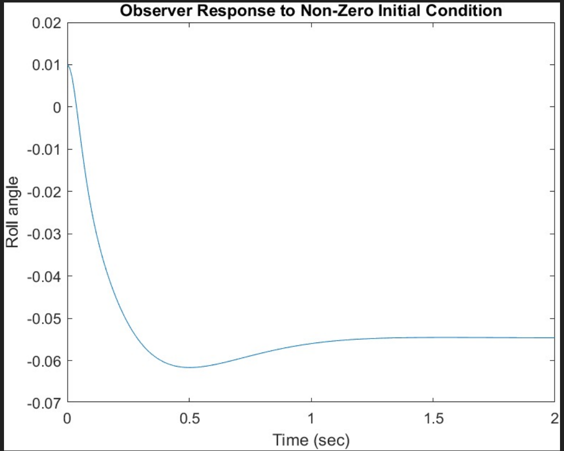




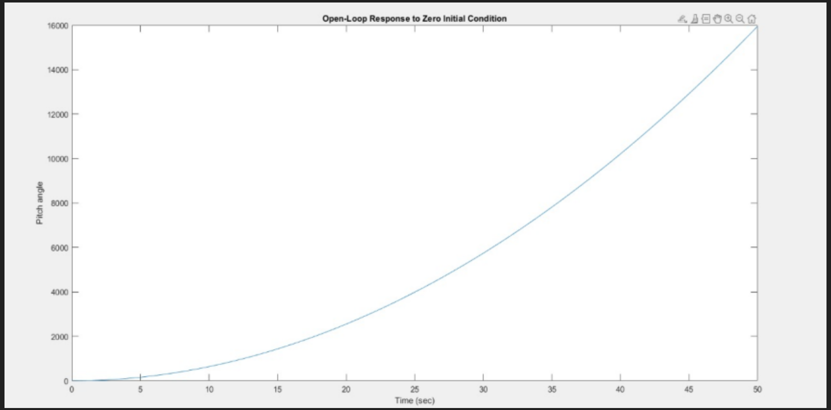
Results - (Plots of *Roll* (Φ) Output) - F_4 *alone is given step input*

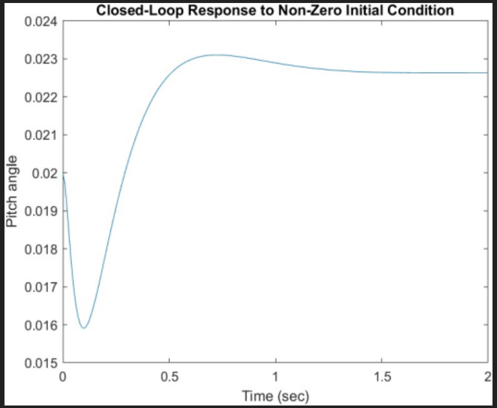


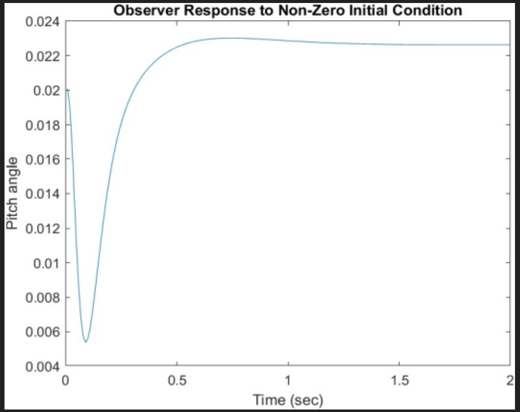




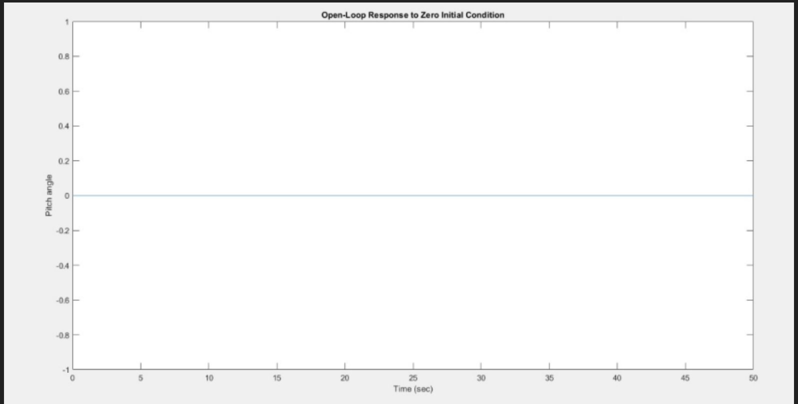
Results - (Plots of *Pitch* (θ) Output) - *F_1 alone is given step input*

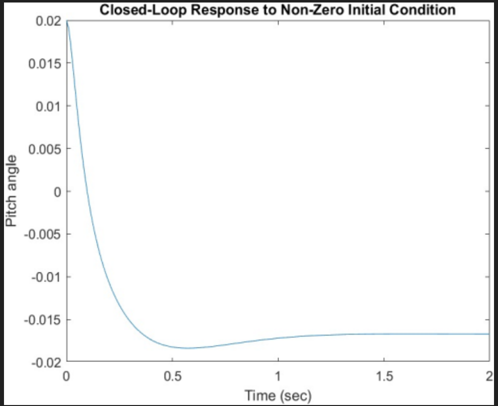


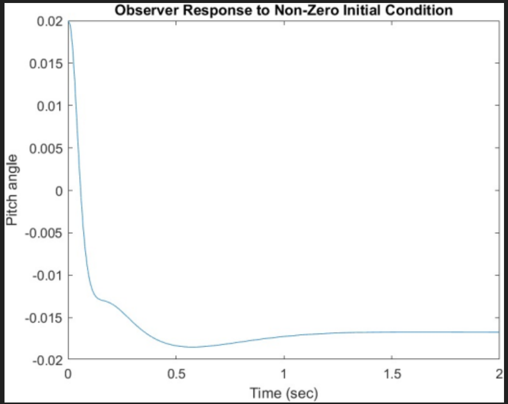




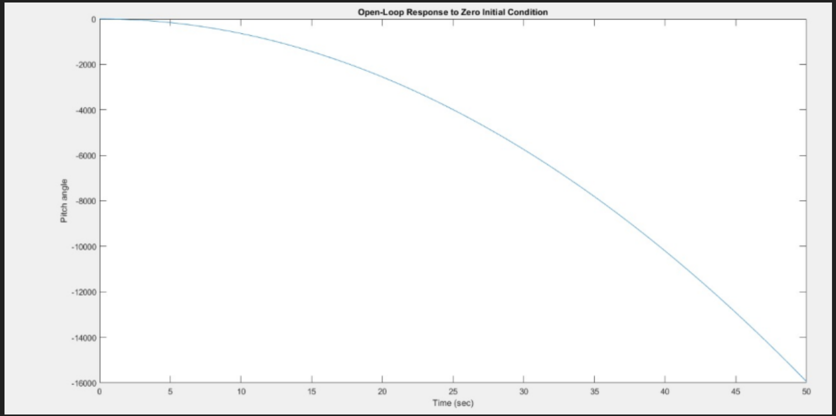
Results - (Plots of *Pitch* (θ) Output) - *F_2 alone is given step input*

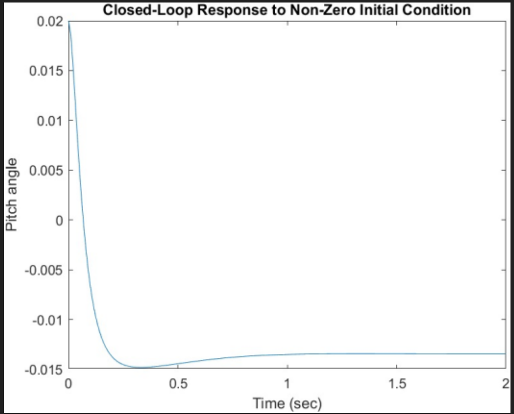


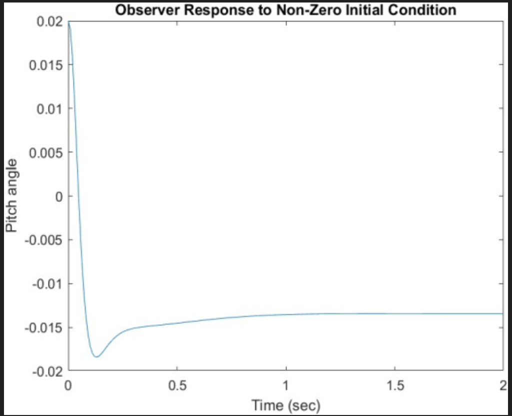




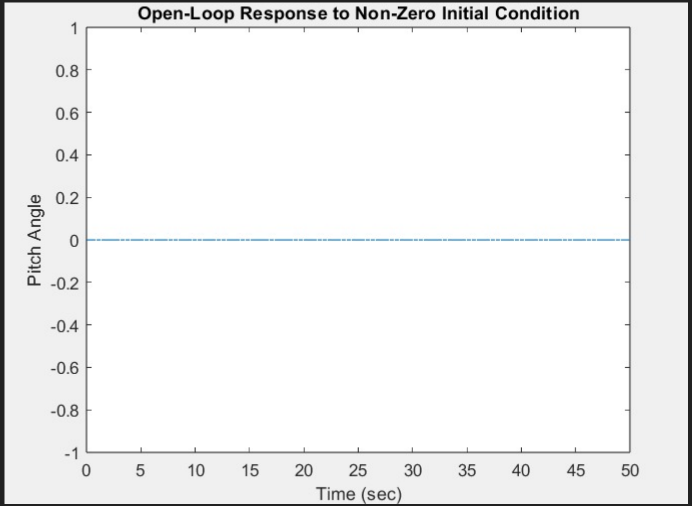
Results - (Plots of *Pitch* (θ) Output) - *F_3 alone is given step input*

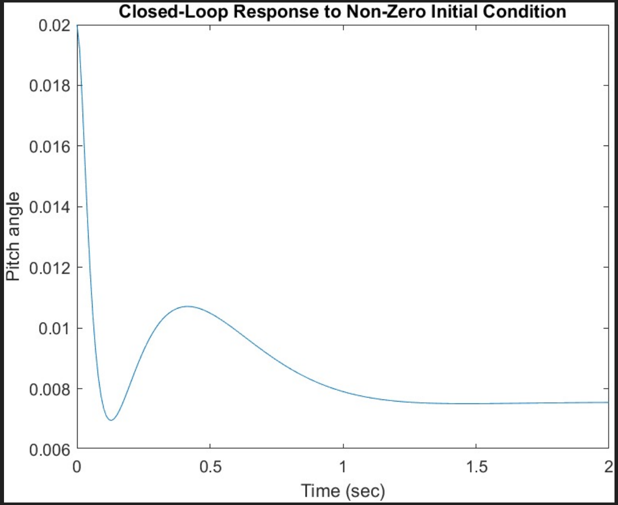


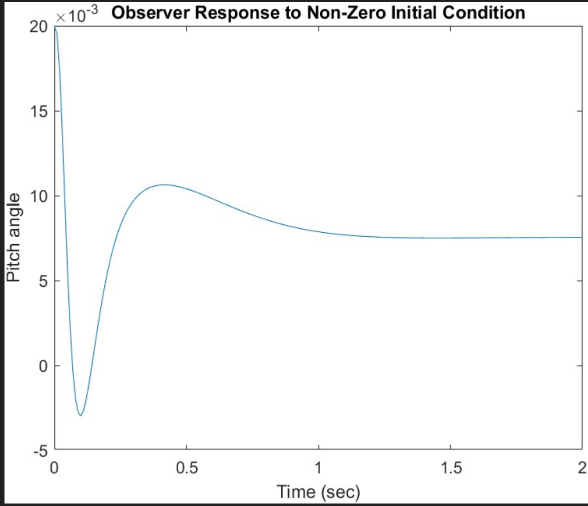




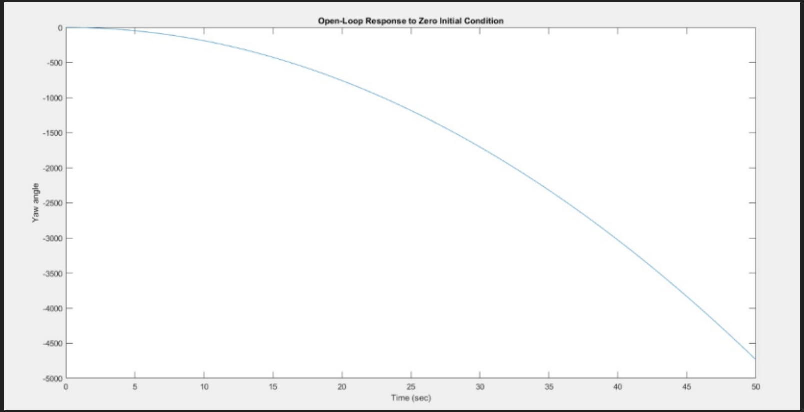
Results - (Plots of *Pitch* (θ) Output) - *F_4 alone is given step input*

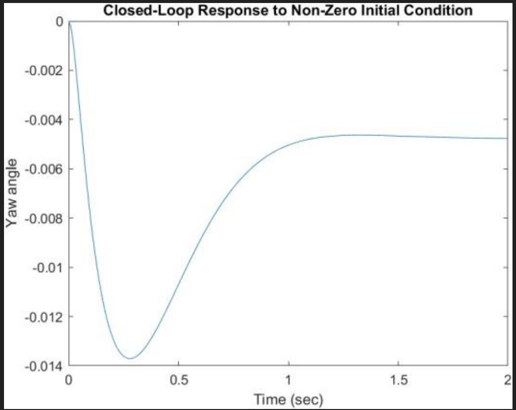


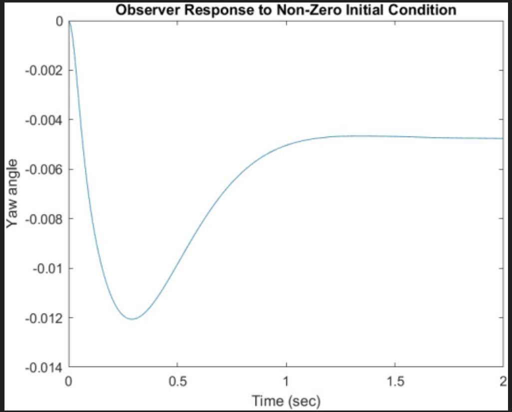




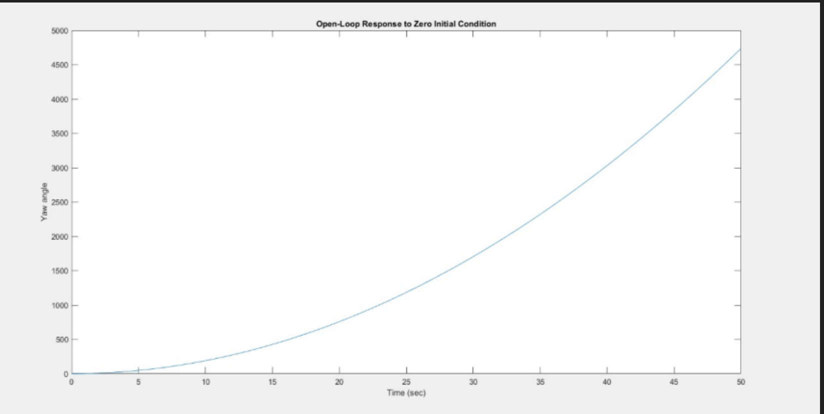
Results - (Plots of *Yaw (Ψ)* Output) - *F_1 alone is given step input*

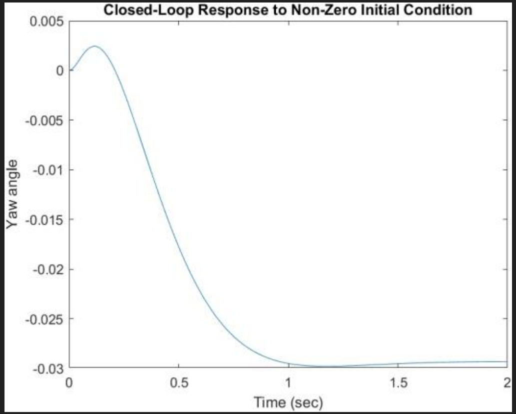


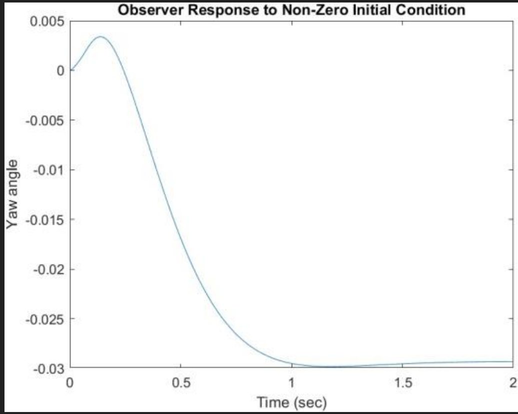




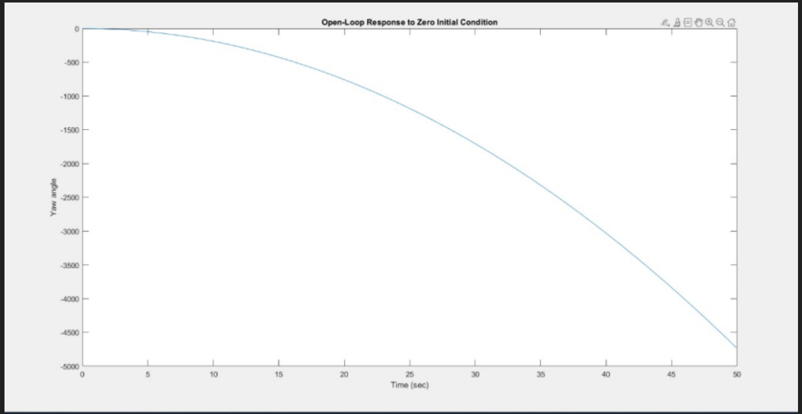
Results - (Plots of *Yaw* (Ψ) Output) - F_2 *alone is given step input*

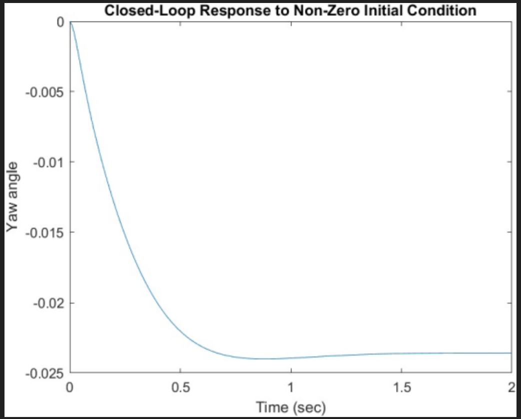


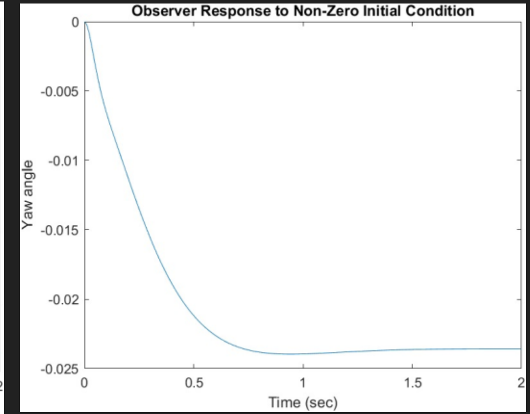




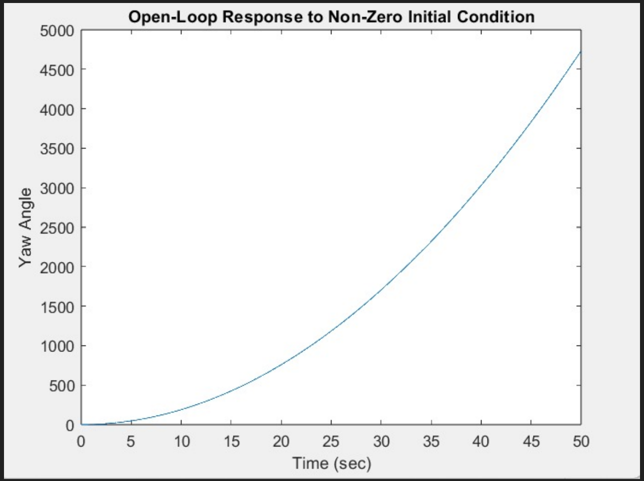
Results - (Plots of *Yaw (Ψ)* Output) - *F_3 alone is given step input*

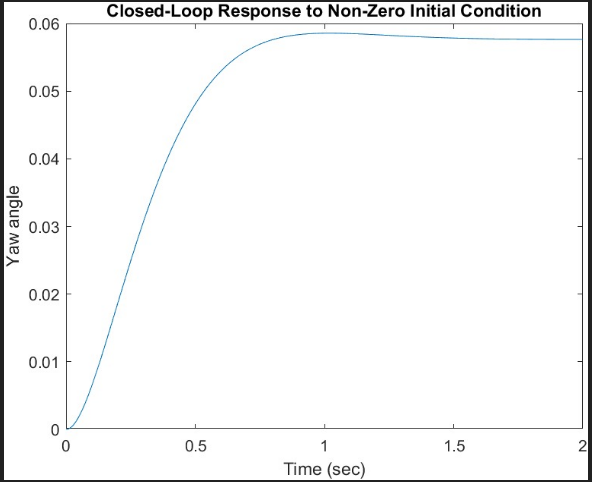


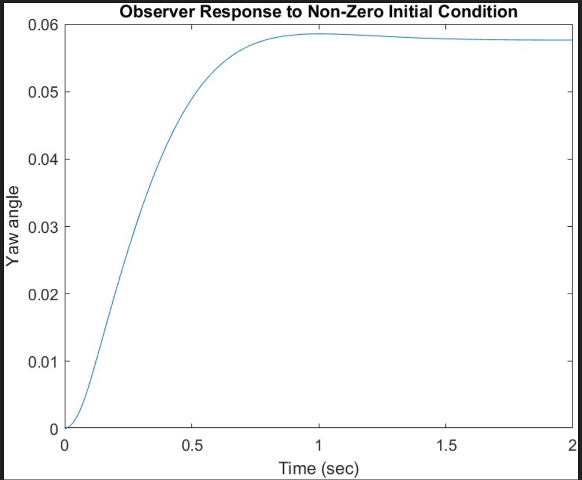




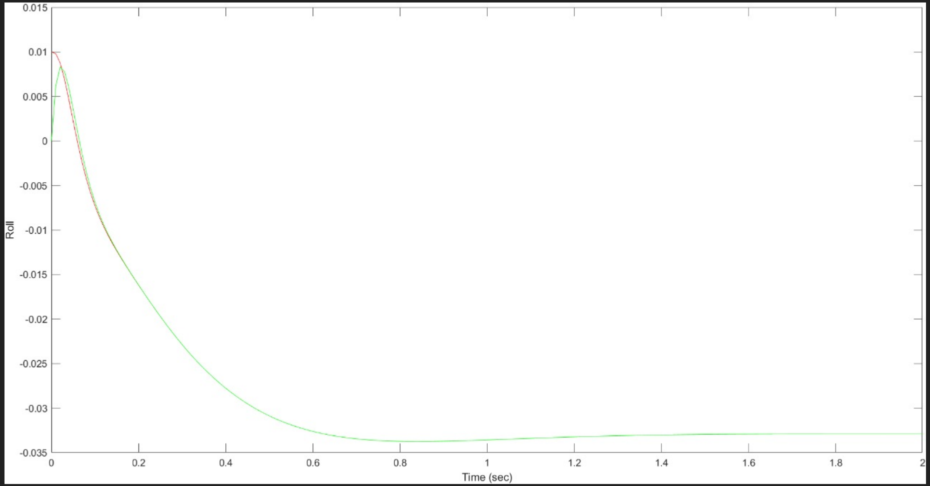
Results - (Plots of *Yaw* (Ψ) Output) - F_4 *alone is given step input*



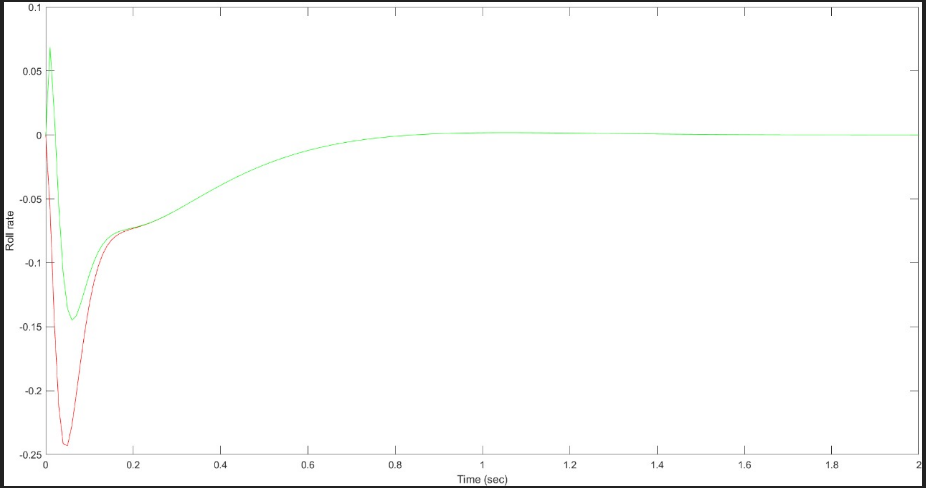




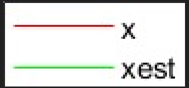
Results - (Plots of Φ and $d\Phi/dt$ States vs t) - *F_1 alone is given step input*



Variation of Φ and Φ^{est} with t

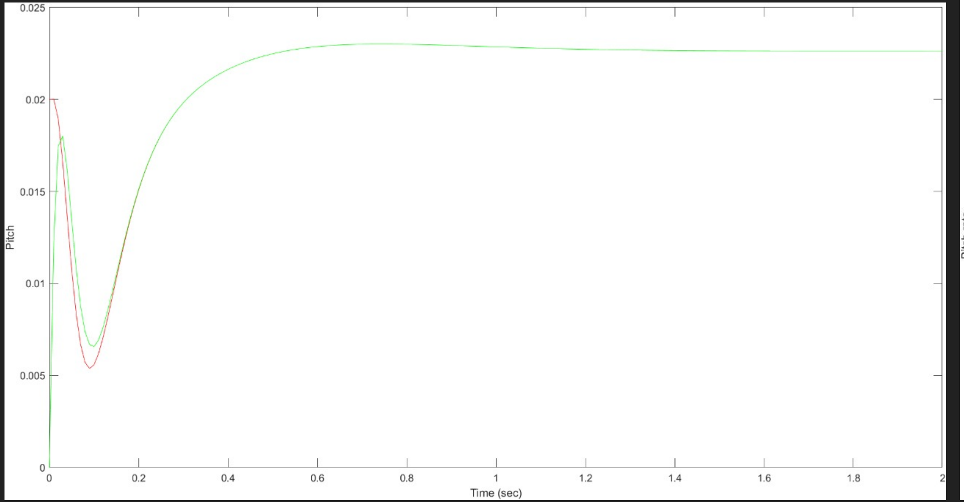


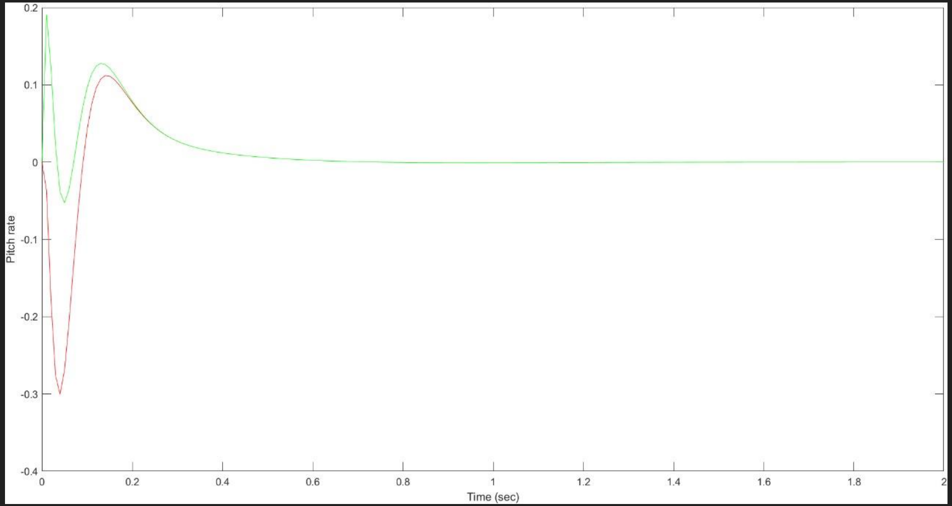
Variation of $d\Phi/dt$ and $d\Phi^{est}/dt$ with t



Legend

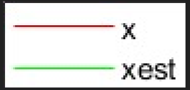
Results - (Plots of θ and $d\theta/dt$ States vs t) - *F_1 alone is given step input*





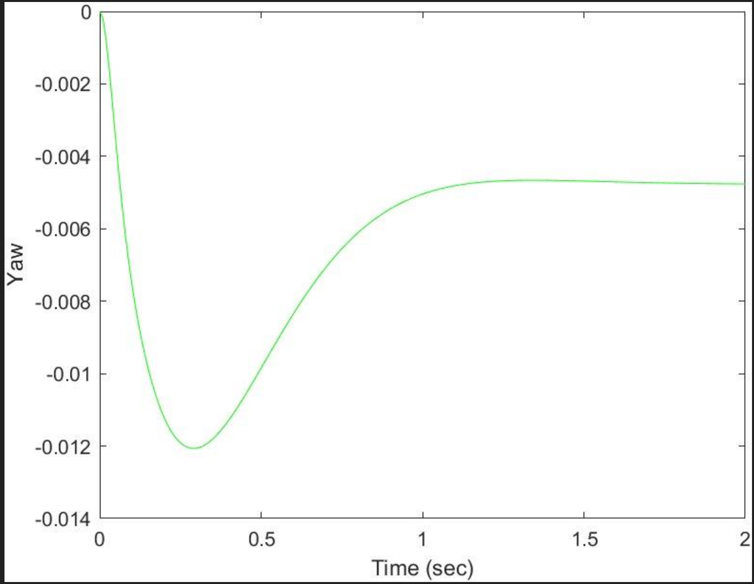
Variation of θ and θ^{est} with t

Variation of $d\theta/dt$ and $d\theta^{est}/dt$ with t

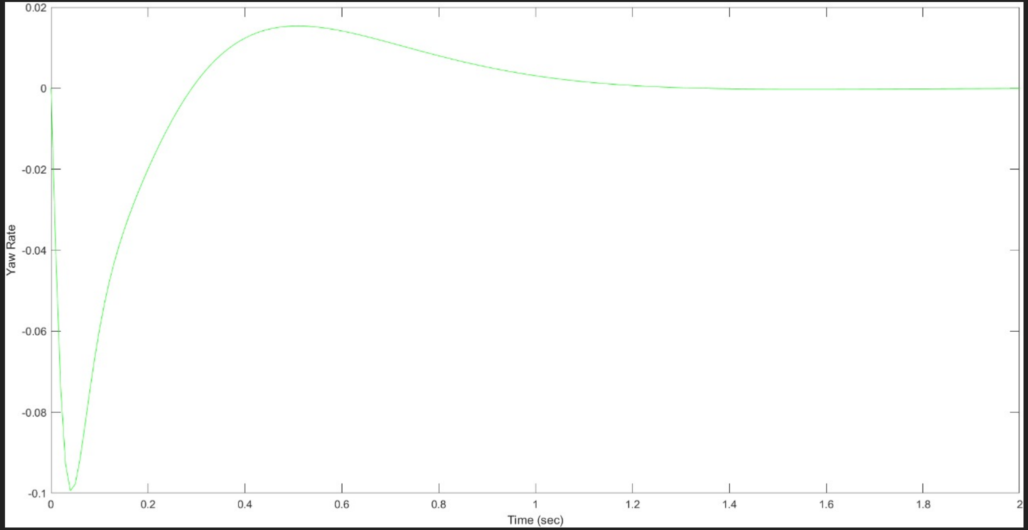


Legend

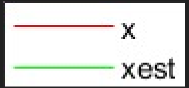
Results - (Plots of Ψ and $d\Psi/dt$ States vs t) - *F_1 alone is given step input*



Variation of Ψ and Ψ^{est} with t



Variation of $d\Psi/dt$ and $d\Psi^{est}/dt$ with t



Legend

References

State Space System Modelling of a Quad Copter UAV. Zaid Tahir Waleed Tahir Saad Ali Liaqat. ArXiv preprint. 2019

Link 1: https://arxiv.org/abs/1908.07401

Introduction: State-Space Methods for Controller Design. MATLAB Tutorial.

Link 2: https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlStateSpace

EE650A: Modern Control Systems

Thank You!

Aravind Seshadri *210180*	*Kapu Nirmal Joshua* *210489*	*Sumay Avi* *211071*	*Raghav Shukla* *210800*