Miami University

Department of Electrical and Computer Engineering

A perspective analytic of a Planar Motion of Quadrotor

Authors
Mirza Sanita Haque
Md Rakibul Ahasan

Advisor
Dr. Bryan Van Scoy

Contents

1 Introduction

As in the name itself, a quadrotor consists of multiple rotors which stabilize themselves using a controller and fly in upward, downward, back, and forth directions. In another way, an effective controlling system enhance quadrotor uses in different applications. One of the major application is the Unmanned Aerial Vehicle(UAV) which has two types fixed wings(FW) and rotating wings(RW). The fixed wing concept of a UAV requires constant wind flow over its span to make it move forward but it does not support stationary motion. Because of this rotating wing UAV contribute vastly to military support, wireless communication, distance learning, and many more. Using the basic principle of quadrotor, it can instantly hover, fly in any direction, and with convenient speed and angular momentum.

The most challenging part of designing a quadrotor is its controller. This controller should have a stable position in all cases of the thrust and torque produced by the rotors. The planar motion is the simple position of a quadrotor means going up and down. Due to its simplicity, the mathematical model of the controller is quite straight forward.

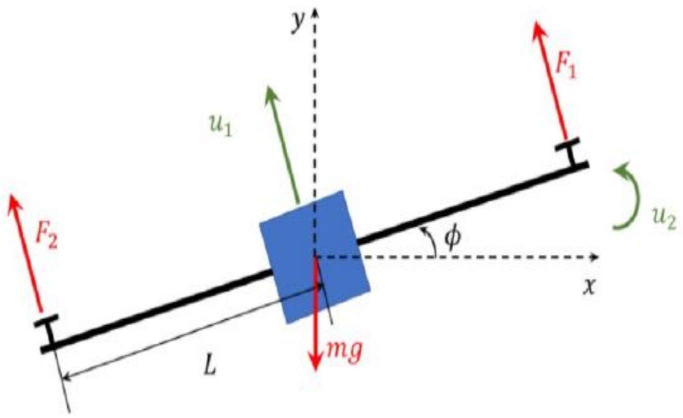

Figure 1.1: A Simple Planator Quadrotor.

Fig1.1 is defining the basic properties of a quadrotor. The coordinate (X, Y) defines the base position and we are expecting the quadrotor will hover without any issue. The net thrust force u_1 is equal to the sum of the force F_1 and F_2 exerted from the rotors. This is initiating the hovering of the quadrotor. This force will be implied on two positions at the same time the net torque will be denoted by u_2. We need to stable the position of quadrotor using input of u_1 and u_2. The angular position of the quadrotor will be zero as we will be working in the hovering aspect of a quadrotor meaning to say in the initial stage the quadrotor will be going up and down. Other important parameters will be the mass of the quadrotor, its acceleration, and its span.

In the subsequent section of this report, we will discuss the modeling and simulation of the planar motion of a quadrotor. Thereafter we will show the linearization and estimation aspect and finally the control aspect of the quadrotor with pole placement, state space feedback and observer based feedback control for non linear dynamics.

2 Quadrotor Basics

The name quadrotor came from having four rotors and it usually consists of frame, motors, ESC (electronic speed controller), propeller, battery, Flight Controller, rc receiver and sensor. The sensor plays important steps to identify the system states.

2.1 Sensors

There are many sensors and the important one are listed below:

1. Ultrasound: It is in the bottom part of the quadrotor which use high frequency sound to measure the distance of quadrotor from the ground. It is the main driver to measure the altitude of the quadrotor.
2. Camera: This is also in the bottom part of the quadrotor. This sensor take photos and using image processing techniques it measures horizontal motion and speed of the quadrotor.
3. Pressure Sensor: This is a supplement sensor which compliments the ultrasound sensor. If quadrotor is hovering low altitude then the pressure will be high and subsequently if the quadrotor hovering high then the pressure is low.
4. IMU sensor: This inertial measurement unit sensor is crucial as it measures the linear acceleration and angular rate. This feed the system state roll, pitch and yaw in the feedback loop to more stable hovering of quadrotor.

2.2 Actuators/Motor Principle

Another important parameter are rotor position of planator and quadrotor. In planator the rotor is place in the opposite side showed in Fig 2.1. Where force is in the upward direction and the generating torque will cancel out each other and if other state does not change then the planator will hover in upward direction.

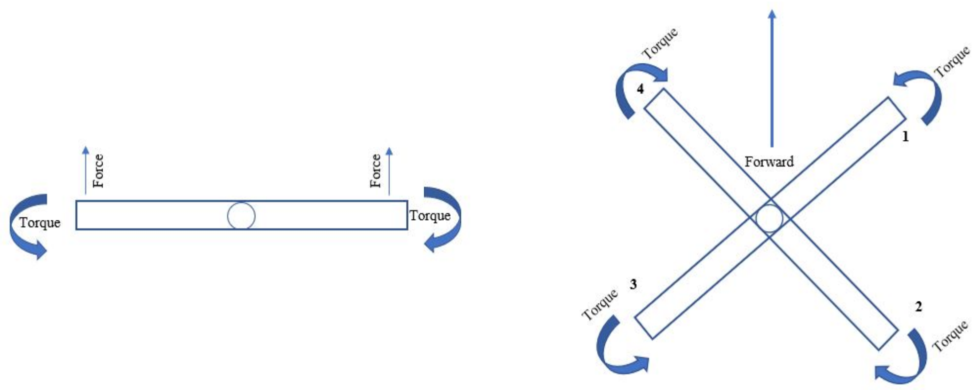

Figure 2.1: Force and Torque state of a Planar Quadrotor and a Quadrotor

For the quadrotor perspective the rotor are in 'X' position. To perfectly hovering the quadrotor the motor in 1,3 will spin in one direction and the motors in 2,4 will hover in opposite direction. An efficient maneuvers of the rotor's will make the quadrotors move in any direction.

2.3 Control Mechanism of a quadrotor

Let us look on below Fig 2.2 where quadrotor input is taken from 4 motors and the expected output we are looking for that the hovering will be perfect. During this the output state is measured through the different sensors and fed back to the system. This system state input are crosschecked with maneuver points or reference point and joint input is plugged to the control system. In the control system the motor mix algorithm is established with combination of thrust, yaw, roll and pitch. This algorithm is established such a way that each thrust, yaw, roll and pitch can control independently. The quadrotor is an under-actuated system which has 4 motors with 6 degree of freedom. The output is classfied to transitional direction and rotational direction. The transitional direction are up/down, left/rigt and forward/backward. The rotational directions are roll, pitch or yaw.

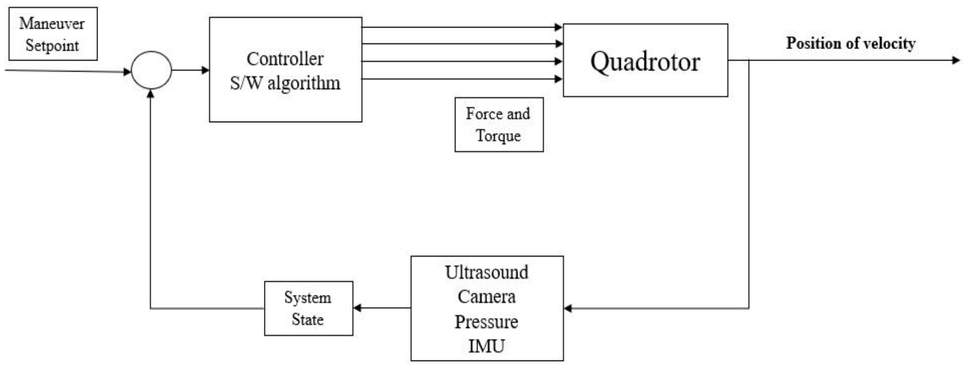

Figure 2.2: Quadrotor Control Principle

3 Mathematical Modeling

3.1 Qualitative Explanation

A quadrotor is a highly unstable system with complex dynamics. From Fig 1.1 lets consider F_1 and F_2 are two perpendicular force and the net thrust force u_1 can calculated from below equation,

$$u_1 = F_1 + F_2. \tag{3.1}$$

The force is imposed from different location hence any imbalance force will generate a torque. The u_2 is the next thrust torque which calculated from below equation where L is the distance from center of force towards the middle point.

$$u_2 = \frac{L}{2}(F1 - F2) \tag{3.2}$$

hence the overall momentum of the system is dependent of the input vector $u = [u_1\ u_2]$.

3.2 Equation of Motion

We have defined the input and output vector and now let's observe the mathematical model of the system. From Newton's second law of motion, the horizontal and vertical direction can be written as below:

$$\ddot{x} = \frac{-u_1 \sin(\phi)}{m}. \tag{3.3}$$

$$\ddot{y} = -g + \frac{u_1 \cos(\phi)}{m}. \tag{3.4}$$

The torque equation of center of mass can be defined from below

$$\ddot{\phi} = \frac{u_2}{J}. \tag{3.5}$$

In matrix from it will be

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{-\sin(\phi)}{m} & 0 & 0 \\ \frac{\cos(\phi)}{m} & 0 & -1 \\ 0 & \frac{1}{J} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ g \end{bmatrix} \tag{3.6}$$

3.3 Equilibrium Point Determination

We can define the state vector as $\begin{bmatrix} x & y & \phi & \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}^T$ where x y is the co-ordinate of the quadrotor initial position and ϕ is the angle, $\dot{x}$ and $\dot{y}$ are the linear and horizontal velocities respectively and $\dot{\phi}$ is the angular velocity. In the initial state of stationary position of system is $\begin{bmatrix} x & y & \phi & 0 & 0 & 0 \end{bmatrix}^T$ as there will be not angular or horizontal or vertical velocity. Putting this in the eq 3.6 will give as below.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{-\sin(\phi)}{m} & 0 & 0 \\ \frac{\cos(\phi)}{m} & 0 & -1 \\ 0 & \frac{1}{m} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ g \end{bmatrix}$$

Solving this equation w.r.t the above condition of equilibrium points we get the below:

$$\begin{gathered} \mathrm{u}_1 \sin(\phi) = 0 \\ \mathrm{u}_1 \cos(\phi) = mg \\ \mathrm{u}_2 = 0 \end{gathered}$$

Now $u_1 \sin(\phi) = 0$ when $\phi = 0$, in such case the state vector will transform as below

$$X = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{3.7}$$

Since the equilibrium angle of the quadrotor is $\phi = 0$, the equilibrium state also forces the constraint $u_1 = mg$.

It is necessary to mention the physical parameter of the quadrotors.

Table 3.1: Parameter of Planator Quadrotor

Parameter	**Value**
Quadrotor mass, m	0.18 kg
Acceleration of gravity, g	9.8 m/s
Span of quadrotor, L	0.086 m
The moment of inertia of quadrotor, J	2.5*$10^{-4} kgm^2$

3.4 Linearization

In Fig1.1 considering net thrust as u_1, net momentum u_2 and gravitational acceleration as input. The co-ordinates of quadrotors x and y also consider to be the output state. Now below states needs to linearize:

$$\ddot{x} = \frac{-u_1 \sin(\phi)}{m} = f_1(x, y, \phi, \dot{x}, \dot{y}, \dot{\phi}, u_1, u_2, g) \tag{3.8}$$

$$\ddot{y} = -g + \frac{u_1 \cos(\phi)}{m} = f_2(x, y, \phi, \dot{x}, \dot{y}, \dot{\phi}, u_1, u_2, g) \tag{3.9}$$

$$\ddot{\phi} = \frac{u_2}{J} = f_3(x, y, \phi, \dot{x}, \dot{y}, \dot{\phi}, u_1, u_2, g) \tag{3.10}$$

Linearize the first state will give us below outcome which will be simplified in eq 3.11.

$\delta\ddot{x} = \frac{\partial f_1}{\partial x}|\delta x + \frac{\partial f_1}{\partial y}|\delta y + \frac{\partial f_1}{\partial \phi}|\delta\phi + \frac{\partial f_1}{\partial \dot{x}}|\delta\dot{x} + \frac{\partial f_1}{\partial \dot{y}}|\delta\dot{y} + \frac{\partial f_1}{\partial \dot{\phi}}|\delta\dot{\phi}$

$\delta\ddot{x} = -\frac{u_1}{m}cos(\phi)$

$$\delta\ddot{x} = -\frac{mg}{m}\delta\phi = -9.8\delta\phi \tag{3.11}$$

In similar manner we will linearize other two state

$\delta\ddot{y} = \frac{\partial f_2}{\partial x}|\delta x + \frac{\partial f_2}{\partial y}|\delta y + \frac{\partial f_2}{\partial \phi}|\delta\phi + \frac{\partial f_2}{\partial \dot{x}}|\delta\dot{x} + \frac{\partial f_2}{\partial \dot{y}}|\delta\dot{y} + \frac{\partial f_2}{\partial \dot{\phi}}|\delta\dot{\phi} + \frac{\partial f_2}{\partial u_1}|\delta u_1 + \frac{\partial f_2}{\partial g}|\delta g$

$\delta\ddot{y} = -\frac{u_1}{m}sin(\phi) + \frac{cos(\phi)}{m}\delta u_1 - \delta g$

when ϕ=0 then it becomes

$$\delta\ddot{y} = -\delta g + \frac{\delta u_1}{m} \tag{3.12}$$

$\delta\ddot{\phi} = \frac{\partial f_3}{\partial u_2}|\delta u_2$

$$\delta\ddot{\phi} = \frac{\delta u_2}{g} \tag{3.13}$$

3.5 State Space Realization

From equation 3.11, 3.12 and 3.13 we have the linearized system. Let us draw those equation in state space realization form

The state-equation:

$$\begin{bmatrix} \delta\dot{x} \\ \delta\dot{y} \\ \delta\dot{\phi} \\ \delta\ddot{x} \\ \delta\ddot{y} \\ \delta\ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -9.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta\phi \\ \delta\dot{x} \\ \delta\dot{y} \\ \delta\dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & -1 \\ 0 & \frac{1}{J} & 0 \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \delta u_2 \\ \delta g \end{bmatrix} \tag{3.14}$$

and the output equation

$$\begin{bmatrix}\delta x\\ \delta y\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\delta x\\ \delta y\\ \delta\phi\\ \delta\dot{x}\\ \delta\dot{y}\\ \delta\dot{\phi}\end{bmatrix} + \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\delta u_1\\ \delta u_2\\ \partial g\end{bmatrix} \tag{3.15}$$

From eq 3.14 and 3.15 we can deduced system matrix A, control matrix B, output matrix C and feed-forward matrix D.

$$\text{A}=\begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & -9.8 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}, \text{B}=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\\ \frac{1}{m} & 0 & -1\\ 0 & \frac{1}{J} & 0\end{bmatrix},$$

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

As we have the matrix we can deduce the transfer function from below formula in matrix form

$$H(s) = C(sI - A)^{-1}B + D \tag{3.16}$$

Hence the open loop transfer function matrix is determined by MATLAB

$$\left(H(s)\right) = \begin{pmatrix}0 & -\frac{g}{Js^4} & 0\\ \frac{1}{ms^2} & 0 & -\frac{1}{s^2}\end{pmatrix} \tag{3.17}$$

The transfer function indicates that the poles are at the origin and there are no zeros. We can can inject the physical parameter in eq 3.17

$$\left(H(s)\right) = \begin{pmatrix}0 & -\frac{39200}{s^4} & 0\\ \frac{50}{9s^2} & 0 & -\frac{1}{s^2}\end{pmatrix} \tag{3.18}$$

Output vector $Y_{out}(s) = H(s)U(s)$ where

$$\text{H(s)}= \begin{bmatrix}X(s) & Y(s)\end{bmatrix}^T$$

$$\text{U(s)}= \begin{bmatrix}U_1(s) & U_1(s) & \frac{g}{s}\end{bmatrix}^T$$

3.6 Controllability and Observability

From the eq 3.14 and 3.15 we have the necessary matrix to compute the controllability and observability matrix. The system matrix A is a 6*6 matrix hence the equation of controllability and observability matrix are

$$\mathrm{P} = \begin{bmatrix} B & AB & A^2B & A^3B & A^4B & A^5B \end{bmatrix}$$

$$\mathrm{Q} = \begin{bmatrix} C \\ CA \\ C^2A \\ C^3A \\ C^4A \\ C^5A \end{bmatrix}$$

To deduce the matrix value we have taken help from matlab and the P and Q matrix is as below

Controllability Matrix of quadrotor:

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -g/j & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & -1/m & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 1/j & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g/j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\ -1/m & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\ 0 & 1/j & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \end{bmatrix} \tag{3.19}$$

Observability Matrix of quadrotor:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.20}$$

Upon putting the observability and controllability matrix in matlab we got the full row rank hence the system is both controllable and as well as full column rank hence the system is observable.

4 Simulation and Controller Design

Now we will observe multiple aspect of controller both for linear dynamics and non linear dynamics.

4.1 Controllability to the Origin Using Feedback Controller (Linearized Part)

Initially is checked bringing the initial state condition to the origin. The initial states are $x = 10, y = 10, and\phi = 0$. The dynamic response of controller is plotted considering 10 percent overshoot and settling time 6 seconds. The first two eigenvalues are determined, however the remaining eigenvalues are assumed 10 time further away from the dominant poles. The desired eigenvalues of the closed-loop system were found as following:

$$\text{Poles} = \left[-0.6667 + 0.9096i, -0.6667 - 0.9096i, -6.6667, -7.6667, -8.6667, -9.6667\right]$$

A state feedback of form u = -Kx with zero reference input is enough to control the states to the origin. The gain matrix w.r.t poles are

$$\text{K} = \begin{bmatrix} 0.5932, 11.1698, -4.1168, 1.2623, 2.8543, -0.3273 \\ -0.0023, -0.0016, 0.0235, -0.0031, -0.0002, 0.0045 \end{bmatrix}$$

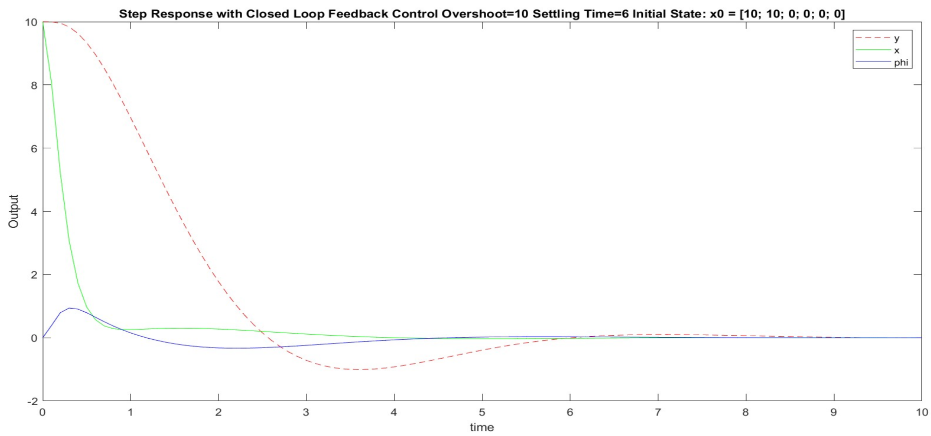

Figure 4.1: Step response analysis of Feedback Controller

The step response of the feedback controller is as above figure 4.2 where the output states close to equilibrium after 6 second.

4.2 Observer based Feedback Controller(Linearized Part)

In the closed loop feedback controller the assumption was all the state variable is know in every point of time. But in real world there are many it is not possible to know all the state due to many physical constraint. The efficient way is to design an observer where the states are estimated asymptotically.

The observer is design with correspondence of the observer gain matrix L which calculates from the state estimator. It is expected that observed based controller will response to dynamic error much faster then the closed loop feedback controller. The poles are placed 10 times further from the closed loop states, percent overshoot is 15 and settling time is 5 for this response.

$$\text{Poles} = \left[-0.8000 + 1.3248i, -0.8000 - 1.3248i, -8.0000, -9.0000, -10.0000, -11.0000\right]$$

The observer gain matrix L was computed by placing the poles at desired location.

$$\text{L} = 1000* \begin{bmatrix} 0.1203 & -0.0286 & 0 & 0 & 0 & 0 \\ 0.0233 & 0.0857 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1900 & 0 & 0 & 0 \\ 2.7366 & -2.6109 & -0.0098 & 0 & 0 & 0 \\ 2.3259 & -1.4314 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8.8000 & 0 & 0 & 0 \end{bmatrix}$$

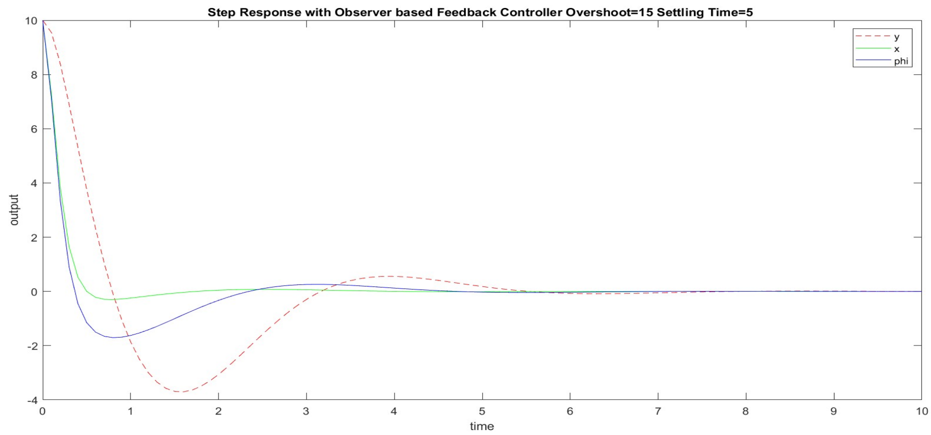

Figure 4.2: Step response analysis of Observer based Feedback Controller

The observer-based feedback response estimates the feedback response closely with a small error and close to equilibrium near about 5 second.

4.3 Position Controller Design Considering the non-linear Dynamics

In earlier section we have seen the footprint of a controller in linearized system. But it is quite uncertain such type of controller will work on a non linear dynamics system specially the motion of the quadrotor. Eq 3.3, 3.4 and 3.5 describe the no-linear dynamics of the planator quadrotor. We have tried to visualize the of those no-linear second order differential equation using matlab function ode45 and outcome proved the planatar motion of a quadrotor is dynamical.

Figure 4.3 give an high level response of x and y for fixed input $u_1 = 15$ and $u_2 = 0.215$. The entire state response with respect to x and y fallen into a way which dignifies the motion are non -linear dynamic as well.

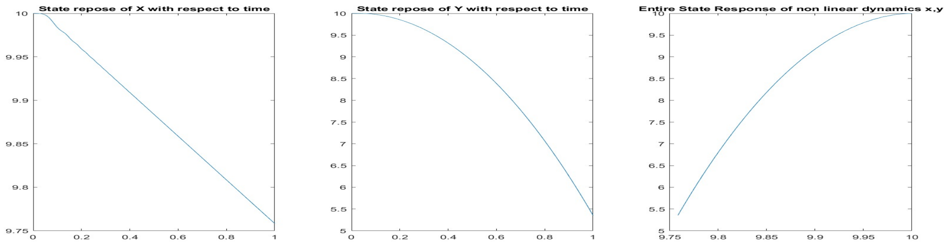

Figure 4.3: Non linear dynamics of Planator Quadrotor

We already know quadrotor is an unstable system. Let us create a positional controller which will take a quadtrotor from position (x_i, y_i) to (x_{des}, y_{des}) for a given input of u_1 and u_2.To move from one position to another position it require altitude control and position control. The key parameters are ϕ the arbitrary angle of quadrotor, thrust force u_1 is y direction and another is in x direction. The angle of the quadrotor is controlled by the momentum u_2. It is quite a bit clear if we want to change the position the thrust will be initially high and later it will rapidly decrease to reach the desired state.

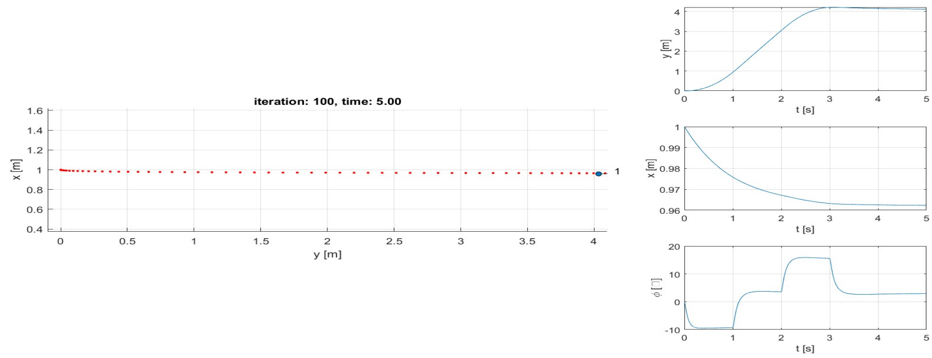

Figure 4.4: Simulation Result of a Quadrotor is Hovering

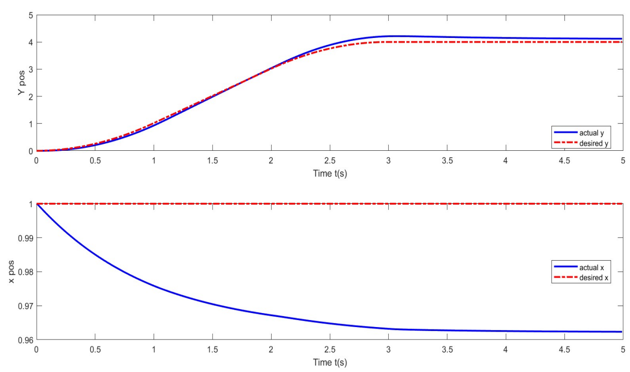

Figure 4.5: Non linear dynamics of Planator Quadrotor

In the complete scenario of planator motion the initial angle ϕ is need to point towards the destination and in destination it will be zero. Hence for each of the motion in we need to have 3 controller to minimize or control the error. One controller will control the thrust depending on the vertical error, one that controls the desired angle based on horizontal error and others control the moment depend on the angle error.

We can also think it of a different way the angle on quadrotor determined by the input moment and location of quadrotor is depend on the current angle and thrust force. Hence three controller is require. The controller equations are

$$\phi_{des} = -(K_{d2} * (\dot{x}_{des} - x) + K_{p2} * (x_{des} - x))/g; \tag{4.1}$$

$$u_1 = m * (K_{d3} * (\ddot{y} - \dot{y}) + K_{p3} * (y_{des} - y)); \tag{4.2}$$

$$u_2 = J * (K_{prot1} * (\phi_{des} - \phi) + K_{drot1} * (\dot{\phi}_{des} - \dot{\phi})); \tag{4.3}$$

There $K_p = [10, 10, 25]$ and $K_d = [10, 10, 25]$ are the certain parameter for u_1 and u_2 for oscillating the position of planator quadrotor with certain overshoot and rise time.

We have plugged the eq 4.1, 4.2 and 4.3 in the ode45 and observe the response of planator quadrotor dymaics in figure 4.4 and figure 4.5 respectively. In figure 4.5 it is evident for our K_p and K_d parameter the actual and desire state in y position is quite a bit similar to the actual.

4.4 State space and observer feedback controller for non linear dynamics

Though we design the position controller in the earlier section we found we need to manually tune the K_p and K_d gain for each state. So the best way is to design a controller which take the state space and observer feedback and states will be more stabilize over time. Figure4.6 shows a basic diagram require for state space observer feedback controller. Here we need plant which has the dynamics of the system. These are:

$$\begin{gathered} \dot{s} = As + Bu \\ y = Cs \end{gathered}$$

The observer dynamics need to define as well. An observer will have both system dynamics along with its own dynamics. These are:

$$\begin{gathered} \dot{\hat{s}} = A\hat{s} + Bu + L(y - \hat{y}) \\ \hat{y} = C\hat{s} \end{gathered}$$

Here L is a 6*6 matrix which we get from observer based feedback controller of our linearized system from section 4.2.

The input vector will be redefine with respect to state space feedback which is,

$$u = \tilde{u} - K(s - \tilde{s})$$

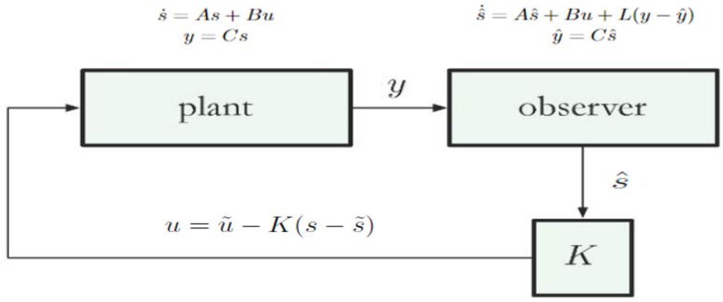

Figure 4.6: State space and observer feedback controller

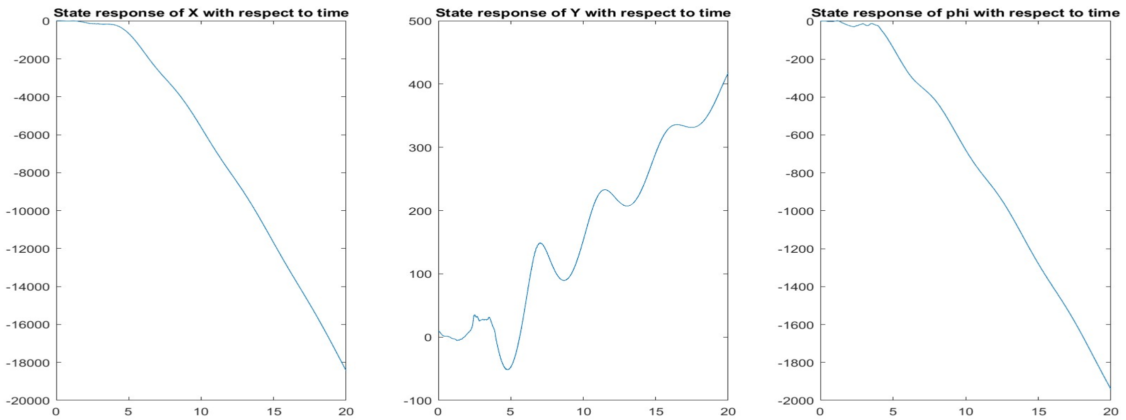

Figure 4.7: State response where initial $dynamics = [10; 10; 0; 0; 0; 0; 10; 10; 0; 0; 0; 0]$

here $\tilde{u} = [0; 0]$ and $\tilde{u} = [mg; 0]$ and observer state $\tilde{s} = [0; 0; 0; 0; 0; 0]$ and later we will see how it varies over time. In that case $\tilde{s} = [t; 0; 0; 0; 0; 0]$.

We will be taking an iterative approach to understand which parameter will make the state close to the equilibrium. Our initial parameters are input $\tilde{u} = [0; 0]$, $\tilde{s} = [0; 0; 0; 0; 0; 0]$ and initial $dynamics = [10; 10; 0; 0; 0; 0; 10; 10; 0; 0; 0; 0]$. Figure 4.7 is showing the outcome, but the initial state is far from the origin hence it has not reached the equilibrium during our time frame $t = [0, 20]$

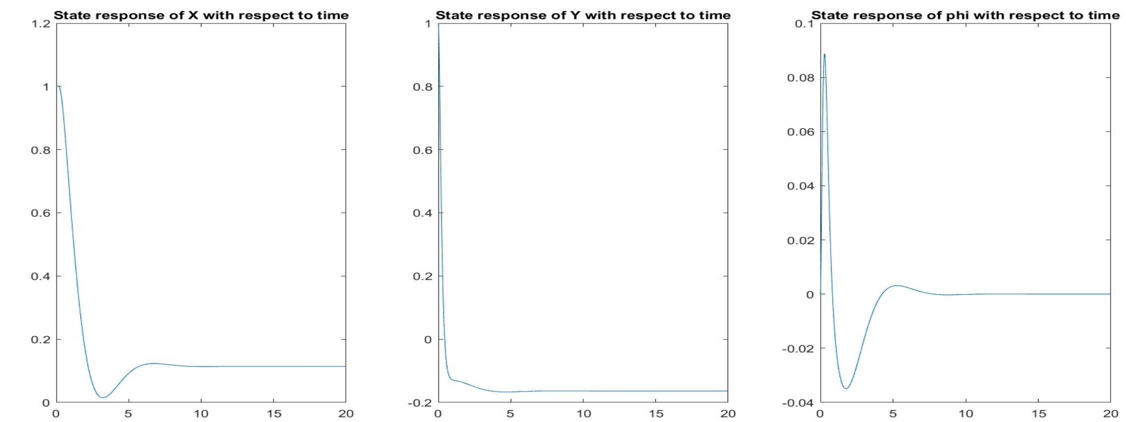

Figure 4.8: State response where initial $dynamics = [1; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 0]$

Figure 4.8 is showing the state response where we initiate the initial dynamics close to the origin.

Here initial $dynamics = [1; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 0]$. Under this circumstance the state response is improved and much more visible. Here the input $\tilde{u} = [0; 0]$ so the gravitational force g has make the quadrotor steady in its position.

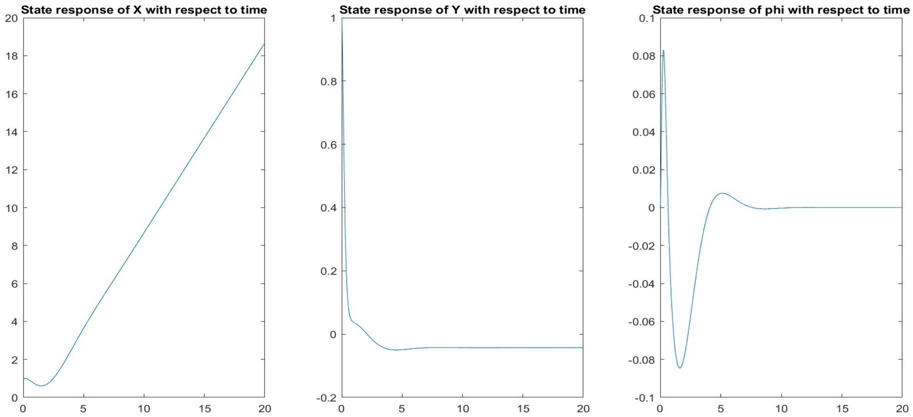

Figure 4.9: State response where initial $dynamics = [1; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 0]$ and $\tilde{u} = [mg; 0]$

Figure 4.9 is showing another combination of parameter we use for our desired state response. Now we use $\tilde{u} = [mg; 0]$ and $state = [1; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 0]$. Also to make the system response more understanding we took $\tilde{s} = [t; 0; 0; 0; 0; 0]$. Now we see the state response x is increasing linearly over time means the quadrotor is hovering. And y and ϕ is also closing to the equilibrium close to 5s. Here because of the input $\tilde{u} = [mg; 0]$ the quadrotor will hover against the gravitational force with input $u_1 = mg$ and $u_2 = 0$ and reach to the equilibrium.

5 What We Learn

This project report is a graduate project of subject ECE601-State Variable for Engineers. The main objective of this project is to understand and simulate a non-linear system states and control the non-linear system with PD controller. The more focused learning is in below:

1. We have learned the non-linear dynamics of a quadrotor and how the planator motion of quadrotor is derived from Newton's 2nd Law.
2. We have understand how the equilibrium point is set for this non-linear system and finally linearize the system.
3. From the linearized system we figured out system matrix A, control matrix B, output matrix C and feed-forward matrix D, in other way the complete state space realization of a quadrotor.
4. We have also identified the controllability matrix and observability matrix and found the system is both observable and controllable.
5. We deduced state space feedback and observer based feedback controller for a linear state. However, that is proven wrong as quadrotor itself is an non-linear system.
6. We also build a PD controller for the demonstration and observe how a non-linear system can be controlled with PD controller.
7. Finally we conclude by designing a controller which consider both the state space and observer feedback and we found our desired state by taking an iterative approach.