PROJECT REPORT

PROJECT TITLE: Modeling of attitude controller in Bicopter

GROUP MEMBERS:

Aniruddha Joshi (2020A8PS2152H)

Tanay Ranjan (2020A3PS0483H)

Aayush Hedaoo (2020A3PS0547H)

COURSE TITLE: Modern Control System (MCS)

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

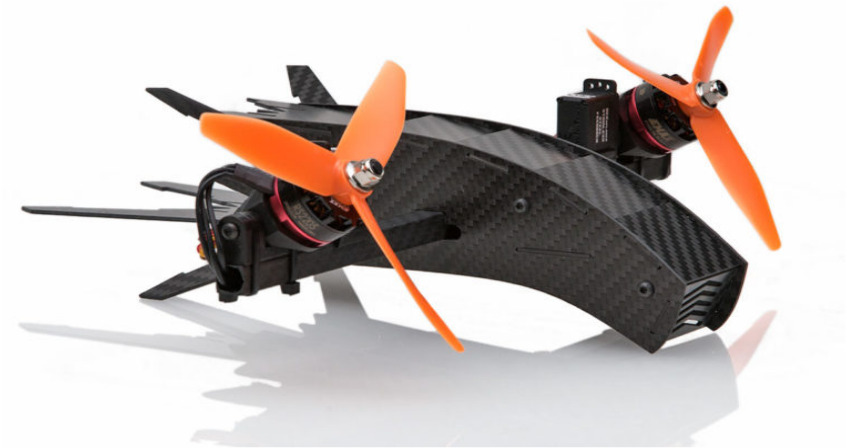
CONTENTS

ABSTRACT

The aim of this project is to design an attitude controller for a bicopter. A bicopter is a type of drone which has 4 motors - 2 motors for giving thrust (will be referred to as thrust motors) and other 2 motors for controlling the angle of thrust (will be referred to as angle motors).

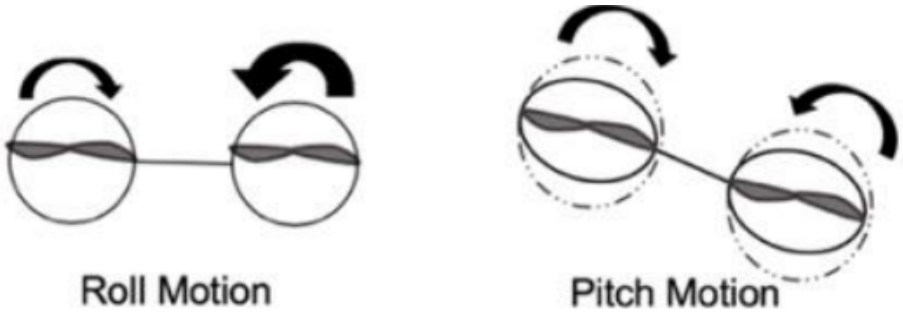
Some common multi rotor drones like quadcopters and hexacopters are easy to stabilize but consume too much power and hence won't be able to stay up in the air for long. Bicopters, even though hard to stabilize and control, consume less power as compared to the others.

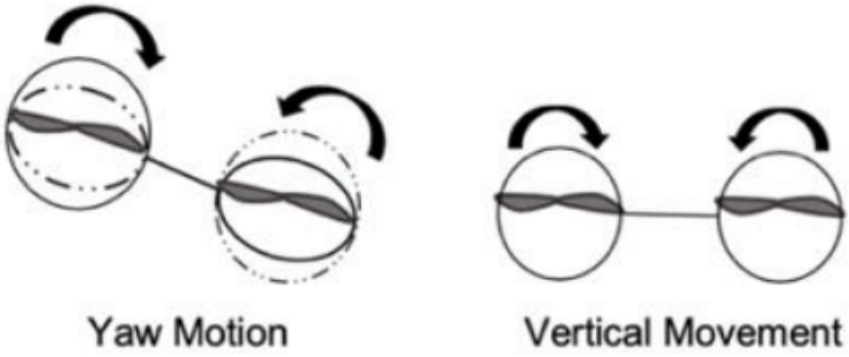
We have used MATLAB software to see how a bicopter reacts to a sudden impulse and how to stabilize it. We have used PID as well as fuzzy logic to simulate our result.



In order to generate roll motion, motors are driven with different speeds. The pitch motion is generated by tilting the rotors in the same direction when the propellers are co-rotating. The yaw motion, on the other hand, is produced by tilting the rotors in the opposite direction while rotating both of them with the

same speed. Finally, the vertical motion is generated by increasing or decreasing the motor speeds at the same time.





IMPLEMENTATION

2D System

Modeling

We can do roll and altitude control using a 2D model of a system only since we don't require the angle motors for controlling them. So we can derive the model expressions by considering a 2D model of the bicopter.

PID

PID stands for proportional-integral-derivative which is used to control different process variables like temperature, and speed in various applications. Here, a control loop feedback device is used to regulate all the process variables.

Altitude Control

We control the altitude of the bicopter after giving a short impulse in the vertical direction. This is done by using a PID controller whose parameters are derived from the bicopter's dynamic equations. Using the dynamic equations we can make the model in MATLAB which consists of a plant and reference. We have assumed the mass of the bicopter to be 2kg and acceleration due to gravity is taken as 9.8 m/s^2.

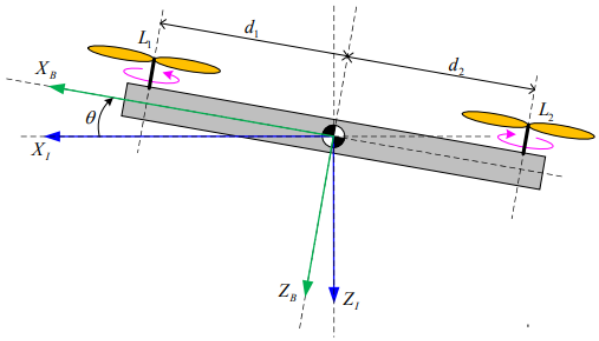


Fig. 1. Bicopter model.

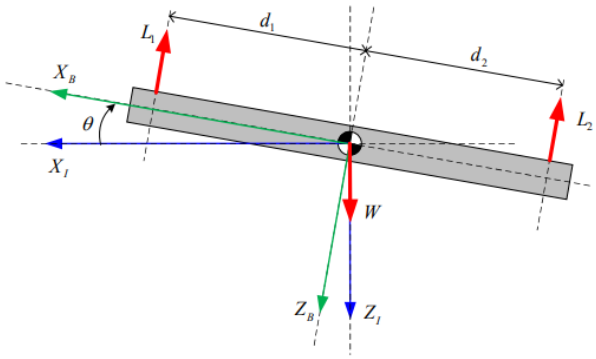


Fig. 2. Bicopter free body diagram.

$$
\begin{aligned}
m\ddot{x} &= -(L_1 + L_2)\sin\theta \\
m\ddot{z} &= mg - (L_1 + L_2)\cos\theta \\
I\ddot{\theta} &= d(L_1 - L_2).
\end{aligned}
$$

Here, m is bicopter mass, I is the inertia of the bicopter, x and z are the bicopter position, θ is the pitch angle, L1 and L2 are respectively thrust generated by motors.

Initial equilibrium equations are given by-

$$
\begin{aligned}
0 &= -(L_{10} + L_{20})\sin\theta_0 \\
0 &= mg - (L_{10} + L_{20})\cos\theta_0 \\
0 &= d(L_{10} - L_{20})
\end{aligned}
$$

Initial values of thrust are equal to mg/2, and initial value of theta is 0. We define the control input as : $u = L_1 + L_2$

$$\ddot{z} = g - \frac{u}{m}$$

Let the error between z and initial z be given as $\tilde{z} = z - z_0$.

$$\begin{aligned} \dot{\tilde{z}} &= \dot{z} \\ \ddot{\tilde{z}} &= -\frac{\tilde{u}}{m} \end{aligned}$$

The additional thrust given to motors is: $\tilde{u} := u - u_0$ .This additional thrust is derived from Lyapunov stability analysis. The stability conditions for a given Lyapunov function V(x) are – V(X) >0 for all X ∈ U\ {0}

(dV/dt) ≤ 0 for all X ∈ U

V(0) = 0

For our system, we define the Lyapunov function as $V = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2$

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= -\frac{\tilde{u}}{m} \end{aligned}$$

$$\dot{V} = e_1 e_2 - e_2 \frac{\tilde{u}}{m}$$

So the feedback control after calculations is $\tilde{u} = m\left[(1 + k_{21})\tilde{z} + k_{22}\dot{\tilde{z}}\right]$

The above equation is used to design the altitude controller. Here the additional control lift is the control input.

In MATLAB we define these functions as -

```
function ua = fcn(zerror,zerror_dot)
m = 2;
g = 9.8;
d = 0.2;
k3 = 3;
k4 = 3;
ubar = m*((1+k3)*zerror+k4*zerror_dot);
ua = ubar/2.0;
```

Roll Control

θ represents the angular motion of the bicopter that is known as the pitch/roll motion. The lift of the front propeller is L1, the lift of the rear propeller is L2, and the weight of the bicopter is W.

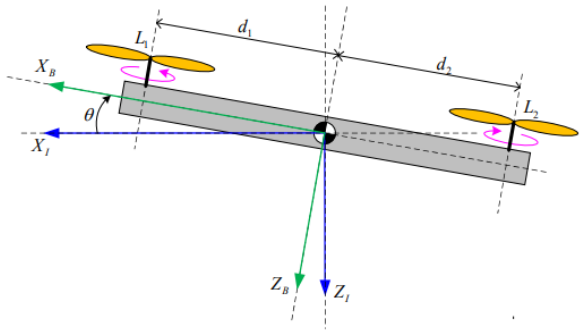


Fig. 1. Bicopter model.

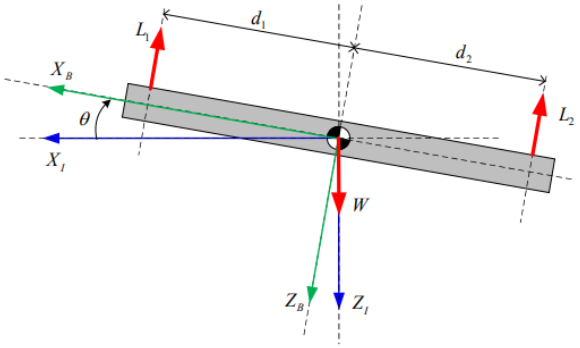


Fig. 2. Free-body diagram of the bicopter model.

The free body equations are :

$$\begin{aligned} \Sigma F_x &= -(L_1 + L_2)\sin\theta \\ \Sigma F_z &= W - (L_1 + L_2)\cos\theta \\ \Sigma M_y &= L_1 d_1 - L_2 d_2 \end{aligned}$$

Bicopter dynamics are as follows :

$$\begin{aligned} m\ddot{x} &= -(L_1 + L_2)\sin\theta \\ m\ddot{z} &= mg - (L_1 + L_2)\cos\theta \\ I_y\ddot{\theta} &= d(L_1 - L_2) \end{aligned}$$

Simplifying above equations we get:

$$\begin{aligned} \ddot{x} &= -\frac{1}{m}(L_1 + L_2)\sin\theta \\ \ddot{z} &= g - \frac{1}{m}(L_1 + L_2)\cos\theta \\ \ddot{\theta} &= \frac{d}{I_y}(L_1 - L_2). \end{aligned}$$

The initial thrusts given to each motor are L_{10} and L_{20} respectively.

So at equilibrium, the double derivatives would be zero.

$$\begin{aligned} \ddot{x}_0 &= -(L_{10} + L_{20})\sin\theta_0 = 0 \\ \ddot{z}_0 &= mg - (L_{10} + L_{20})\cos\theta_0 = 0 \\ \ddot{\theta}_0 &= d(L_{10} - L_{20}) = 0. \end{aligned}$$

Solving above equations, we get $\theta_0 = 0$ and $L_{10} = L_{20} = mg/2$

We define the error in roll angle as : $\tilde{\theta} = \theta - \theta_r$

From the above equations, we get

$\ddot{\tilde{\theta}} = \frac{d}{I_y}[(L_1 - L_{1_r}) - (L_2 - L_{2_r})]$, where $L_{1_r} = L_1 + u_p$ and u_p is the control input.
$L_{2_r} = L_2 - u_p$

So finally we have : $\ddot{\tilde{\theta}} = -\frac{2d}{I_y}u_p$

Define the state variables as $e_1 = \tilde{\theta}$ and $e_2 = \dot{\tilde{\theta}}$ and $\dot{e}_1 = e_2$
$\dot{e}_2 = -\frac{2d}{I_y}u_p$

Similar to altitude control, we will use the Lyapunov stability criteria to define the control input.
The Lyapunov function used here is : $V = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 \quad \dot{V} = e_1e_2 - \frac{2d}{I_y}e_2u_p$,

Define the control input as

$$u_p = \frac{I_y}{2d}[(1+k_1)e_1 + k_2e_2]$$

After substituting and solving, we get

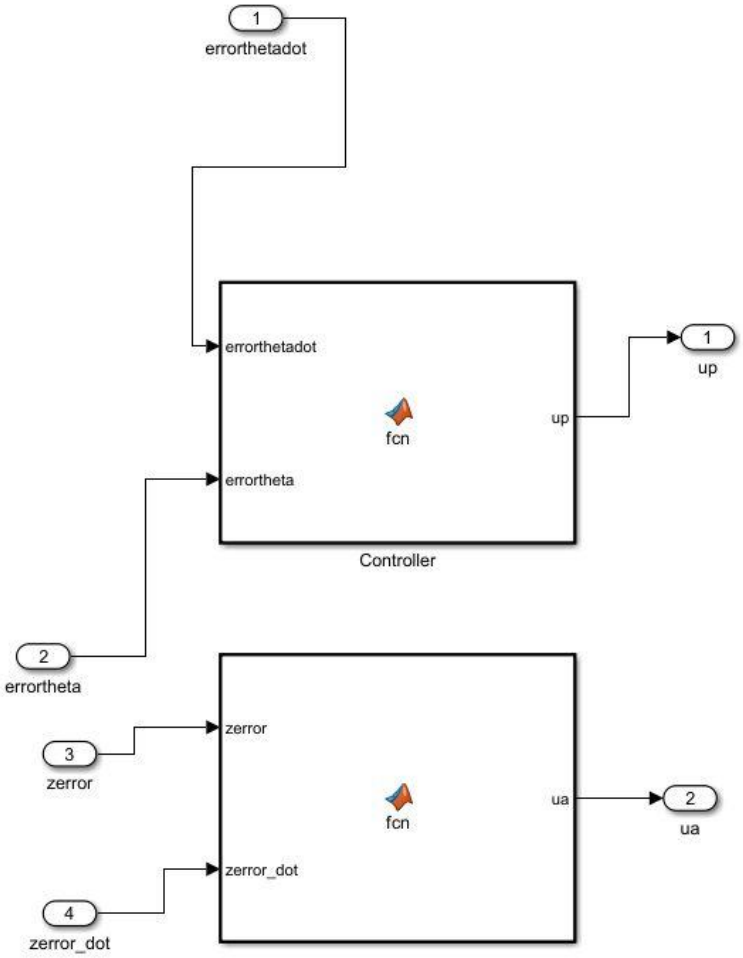
$$\begin{aligned}\dot{V} &= -k_2e_2^2 - k_2e_1e_2 \\ &\leq -k_2e_2^2 - k_1\left(\frac{e_1^2}{2} + \frac{e_2^2}{2}\right) \\ &= -\frac{k_1}{2}e_1^2 - \left(k_2 + \frac{k_1}{2}\right)e_2^2\end{aligned}$$

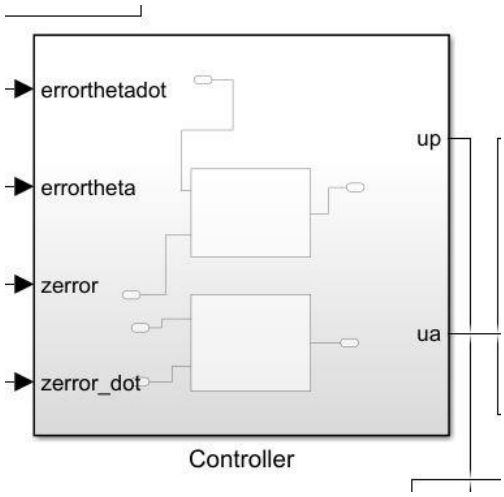
So, the feedback control after calculations is $u_p = \frac{I_y}{2d}\left[(1+k_1)\tilde{\theta} + k_2\dot{\tilde{\theta}}\right]$

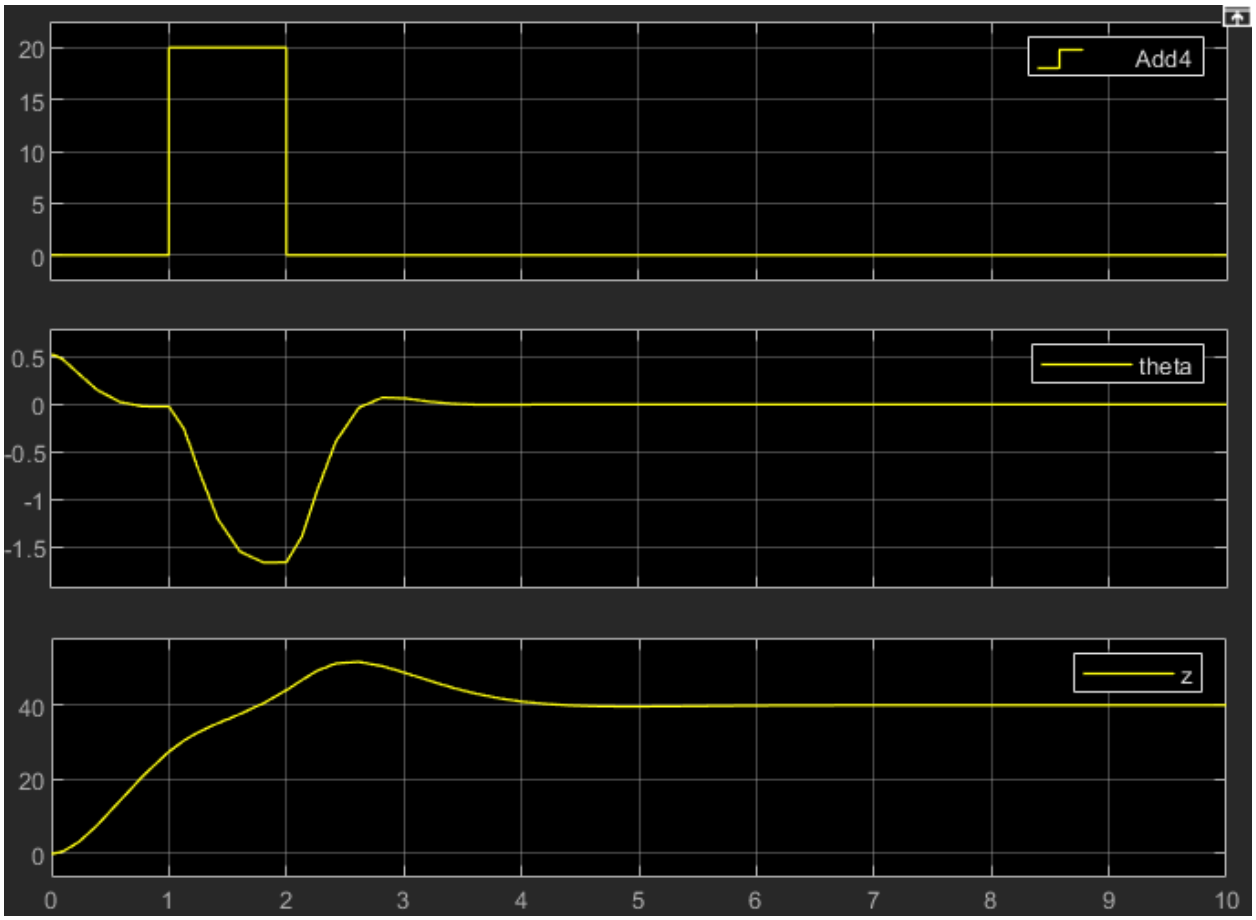
In MATLAB we define these functions as -

```
function up  = fcn(errorthetadot, errortheta)
m = 2;
Iy = 0.2;
g = 9.8;
d = 0.2;
k1 = 24;
k2 = 7;
up1 = (Iy/(2*d))*((1+k1)*errortheta + k2*errorthetadot);
up = -up1;
```

Simulink block diagram for altitude and roll controller is shown below:







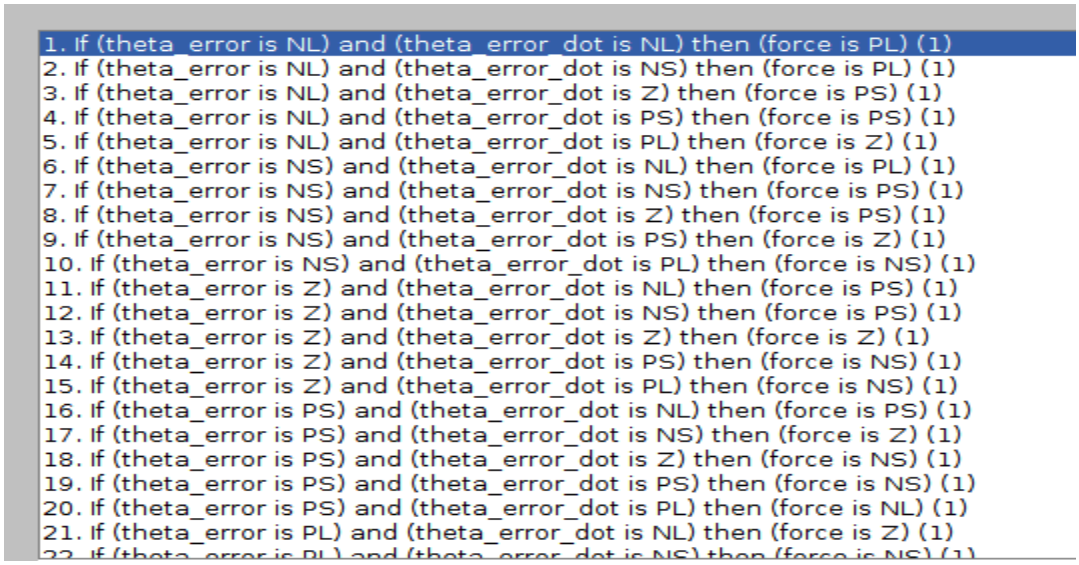
The theta_intial is pi/6 = 0.52, so the theta graph starts from 0.52 and since the theta_reference is 0 so it finally stabilises at 0. We have kept the z reference to be 40, so it gets stabilized at 40.

Fuzzy

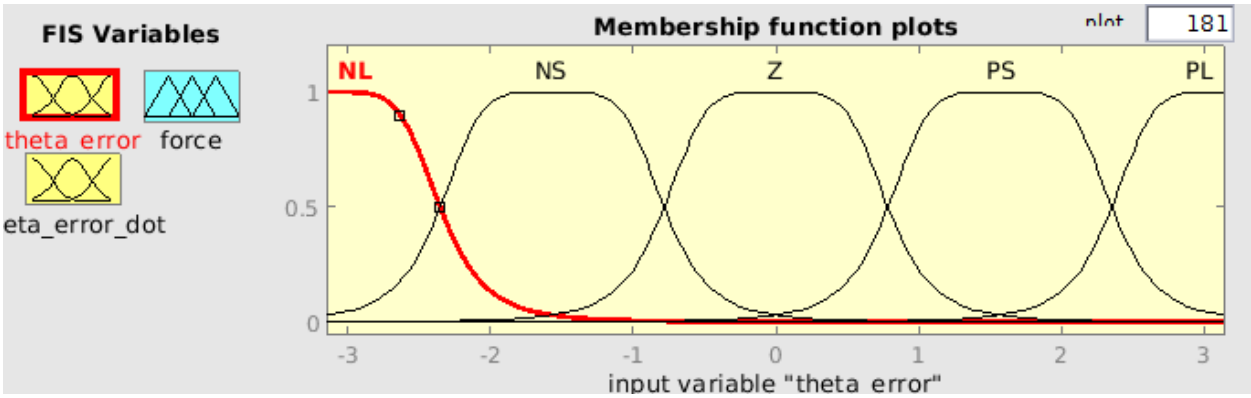
Fuzzy basically means things which are vague or not clear. Fuzzy logic uses common human decision making methodology. There is no perfect boolean value for the decision. For example, if someone says “they are good”, here good is a relative term and we can’t exactly determine the “value” of good. To determine this “value” we define membership functions which give us the values we need. We don’t require any precise inputs. Fuzzy is also easy to understand. So Fuzzy gives better overall results as compared to PID.

Altitude and Roll Control

The input is converted into a fuzzy variable whose value is then determined using the fuzzy logic rules. In our bicopter case, we are passing 2 inputs - theta_error and theta_errordot and the output to be the force. We have used the gbell type membership function. We defined 5 membership functions - NL (Negative Large), NS (Negative Small), Z (Zero), PS (Positive Small) and PL (Positive Large). We also defined the following rules -



The membership function is shown below, we can also define the range of each input according to our need and how much we need the range to be to stabilize the final output.

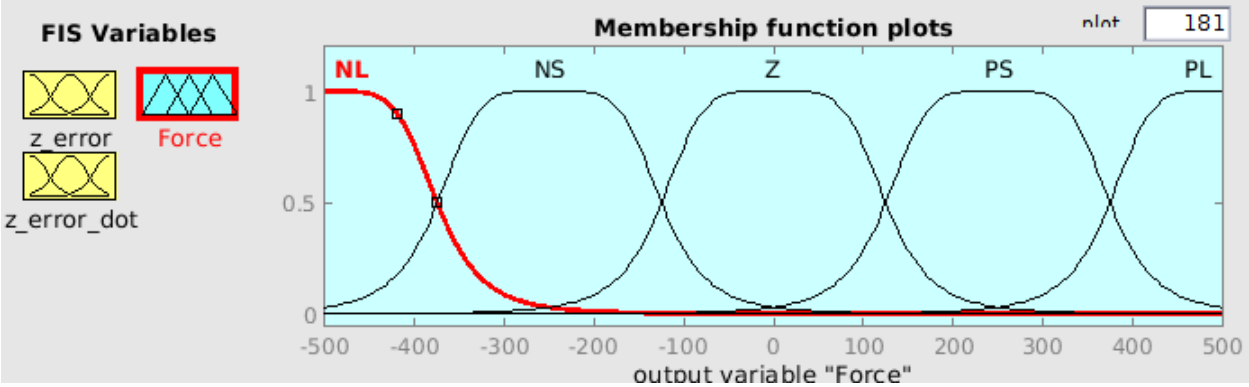


Theta_error has range : [-3.14 3.14]

Theta_error_dot has range: [-25 25]

Force has range : [-500 500]

We can see that the range of force is quite large, this is because we had given an impulse of 100. So to stabilize we needed a much larger force. We have kept the range of theta_error to be in between -pi and pi because, if we go higher than that the bicopter will make a flip which we don't want.



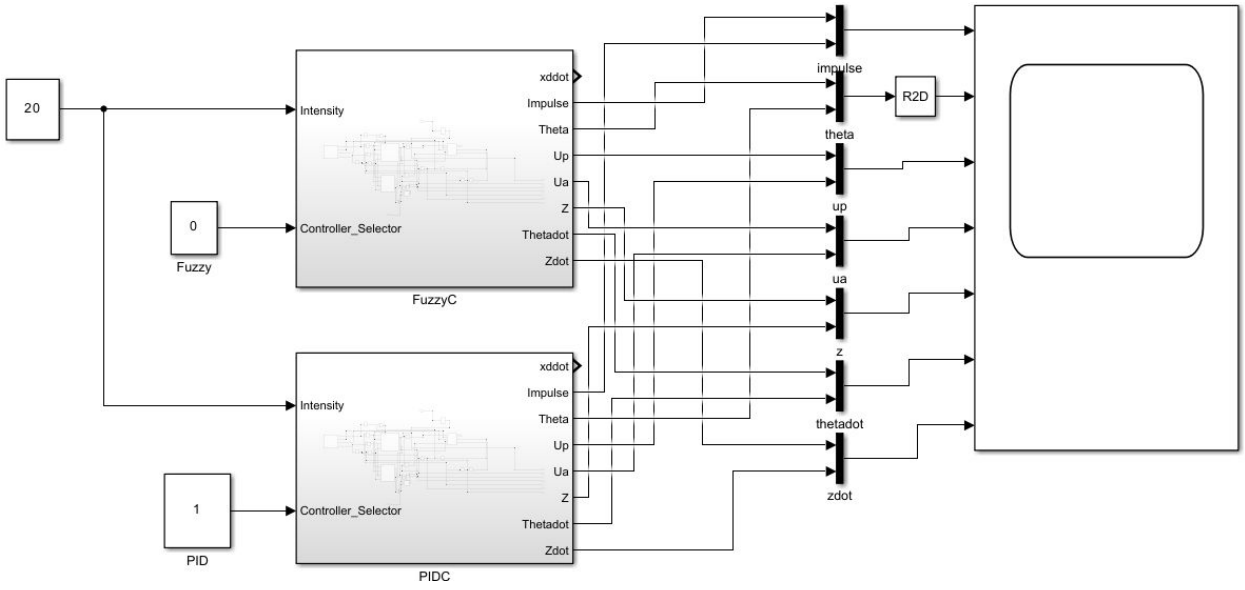
We also made another fuzzy controller for controlling the altitude which has the same membership functions as the roll controller but the range values are different.

Z_error has range: [-50 50]

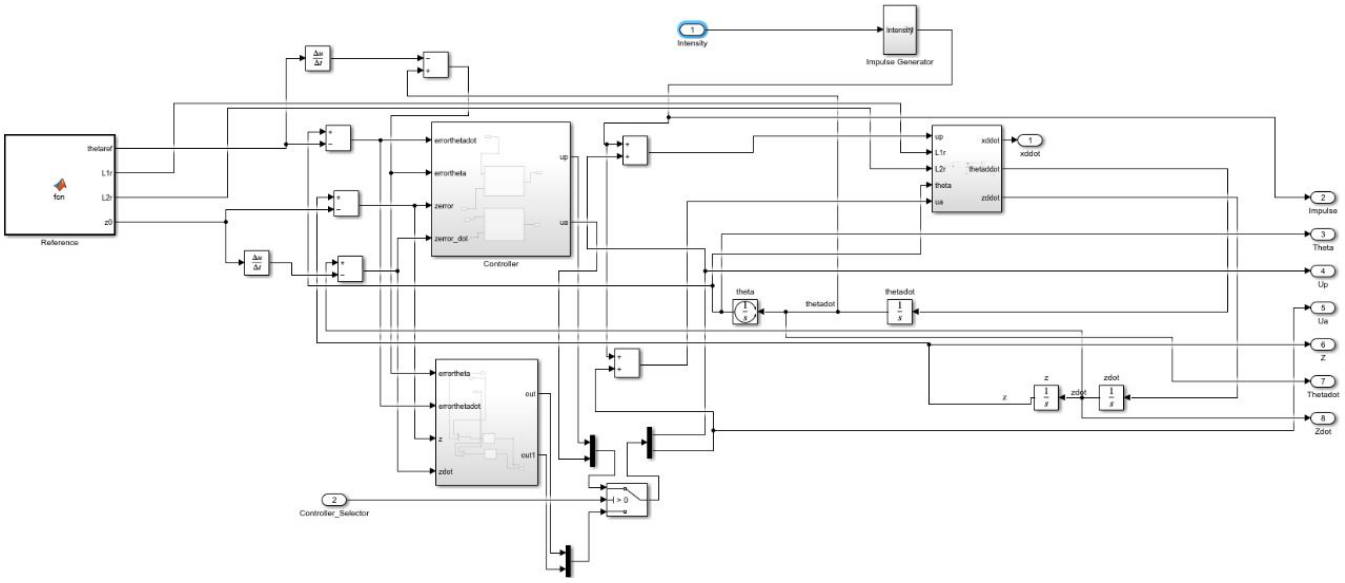
Z_error_dot has range: [-500 500]

Force has range: [-500 500]

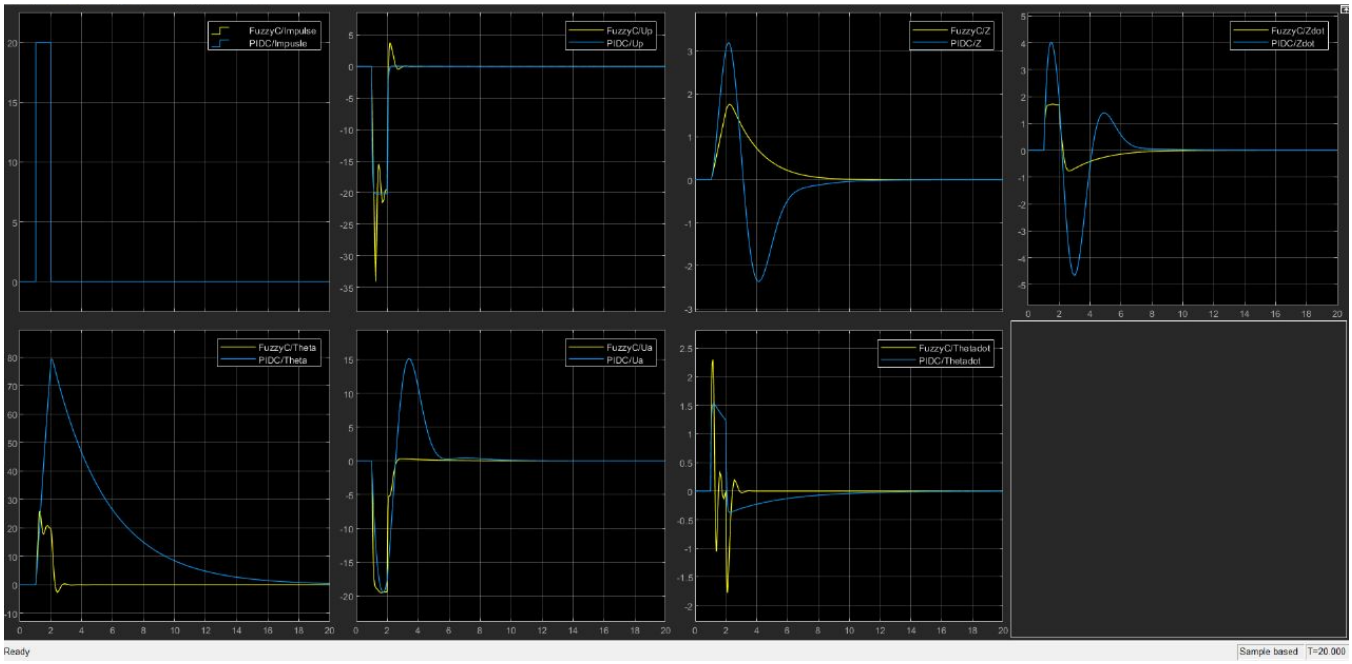
Since we are giving a large impulse the z_error_dot will be high as well so the range is high and same is the case with force.



We made this model of our bicopter, we have made 2 subsystems in which one is using PID and the other is using a fuzzy controller. In the scope we will be able to compare the 2 results and we can clearly see that the fuzzy controller is better than PID. The time required for the output to stabilize is less for fuzzy when compared to PID



This is what is inside both the subsystems. We have used a switching logic to shift between PID and Fuzzy.

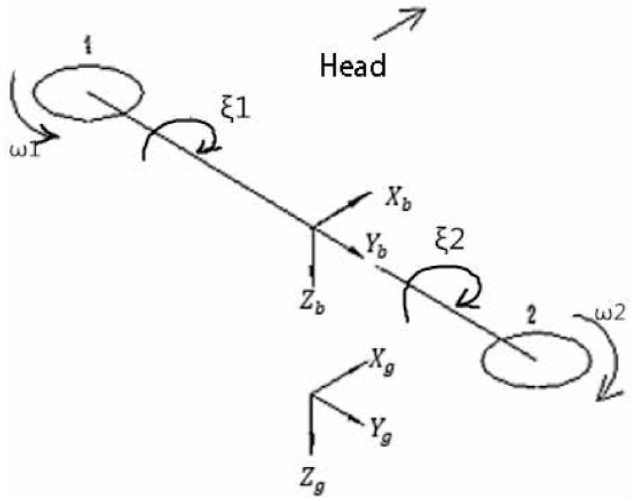


Here the blue one is PID and the yellow is fuzzy. We can see that the z and theta are getting stabilized earlier by the fuzzy controller.

3D System

Modeling

In 2D systems, we only considered 2 controllable parameters, which was, left thrust and right thrust. Now we will consider controlling the angle of thrust (angle of servos). We considered one ground frame and one frame for bicopter.



For considering orientation of bicopter frame with respect to ground frame, we used Euler angles to define its attitude -

- Yaw **ψ** : The angle between the axis of the body in the horizontal plane and the ground axis, the right side is positive.
- Pitch angle **θ**: The angle between the body axis and the horizontal plane, to rise is positive.
- Roll angle **Φ**: The angle that symmetry plane rotate around the body axis, right roll is positive.

In bicopter, we considered pitch to be negligible as we can't control the pitch, it will only depend upon position of centre of mass.

Now,
W.R.T. frame of bicopter, the dynamics equation are the following -

$$\frac{d^2 x_b}{dt^2} = [L_1 sin(\xi_1) + L_2 sin(\xi_2)]/m$$

$$\frac{d^2 z_b}{dt^2} = -[L_1 cos(\xi_1) + L_2 cos(\xi_2)]/m$$

$$\frac{d^2\theta_{x,b}}{dt^2} = [L_1 cos(\xi_1) - L_2 cos(\xi_2)]/I_x$$

$$\frac{d^2\theta_{z,b}}{dt^2} = [L_1 sin(\xi_1) - L_2 sin(\xi_2)]/I_z$$

Let,

$$U_1 = L_1 cos(\xi_1) + L_2 cos(\xi_2)$$
$$U_2 = L_1 cos(\xi_1) - L_2 cos(\xi_2)$$
$$U_3 = L_1 sin(\xi_1) + L_2 sin(\xi_2)$$
$$U_4 = L_1 sin(\xi_1) - L_2 sin(\xi_2)$$

W.R.T. ground frame, the dynamic model equation will look like -

$$\frac{d^2x}{dt^2} = -\frac{[sin(\phi)sin(\psi) + cos(\phi)sin(\theta)cos(\psi)]U_1 - [cos(\theta)cos(\psi)]U_3}{m}$$

$$\frac{d^2y}{dt^2} = \frac{[sin(\phi)cos(\psi) - cos(\phi)sin(\theta)sin(\psi)]U_1 + [cos(\theta)sin(\psi)]U_3}{m}$$

$$\frac{d^2z}{dt^2} = \frac{-[cos(\phi)cos(\theta)]U_1 - [sin(\theta)]U_3}{m} + g$$

$$\ddot{\phi} = \frac{lU_2}{I_x}$$

$$\ddot{\theta} = \frac{hU_3}{I_y}$$

$$\ddot{\psi} = \frac{lU_4}{I_z}$$

Fuzzy approach

On applying the same rules of the 2D fuzzy model but changing the ranges of the input and output variables. In the 3D system, we intended to control yaw angular velocity and linear speed (that is, ydot and yawdot wrt bicopter frame). As we were altering left and

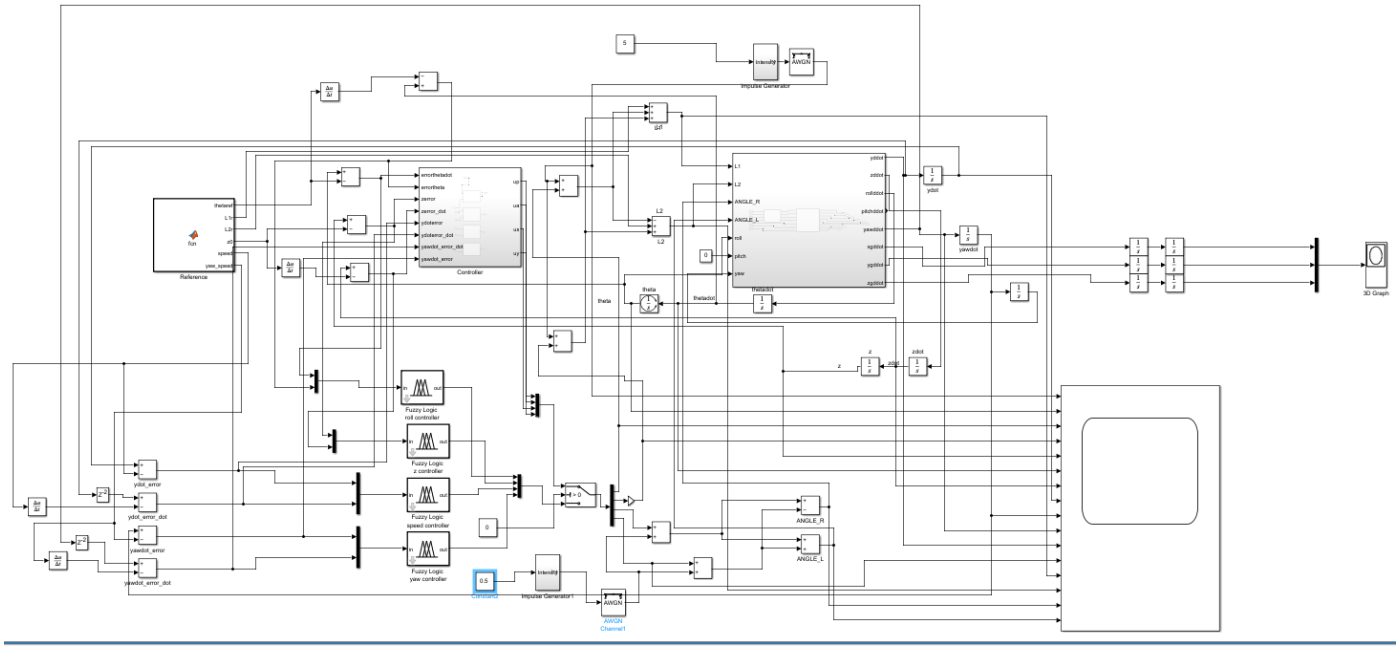
right thrust for controlling roll and altitude, similarly, we will alter thrust angles to control yaw speed and linear speed.
Suppose **us** and **uy** are control inputs for controlling linear speed and yaw speed respectively.
For linear speed, **us** will be added to both left and right angle, and for yaw speed, **uy** will be added to left angle and subtracted to right angle.
Just like altitude and roll control, we took **ydot** and **yddot** for generating **us**, and **yawdot**, **yawddot** for generating **uy**

Simulation

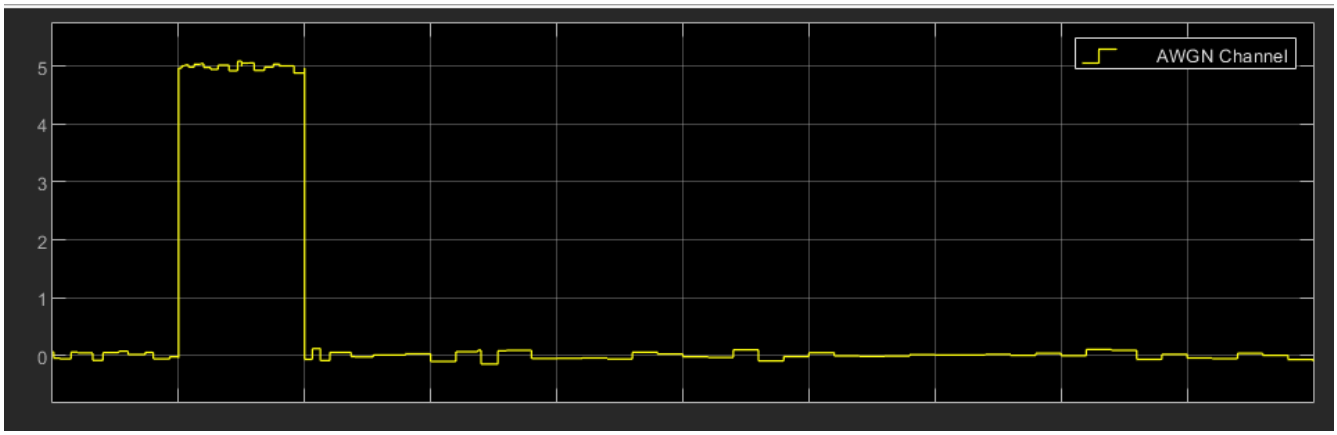
For simulation, we set the following parameters

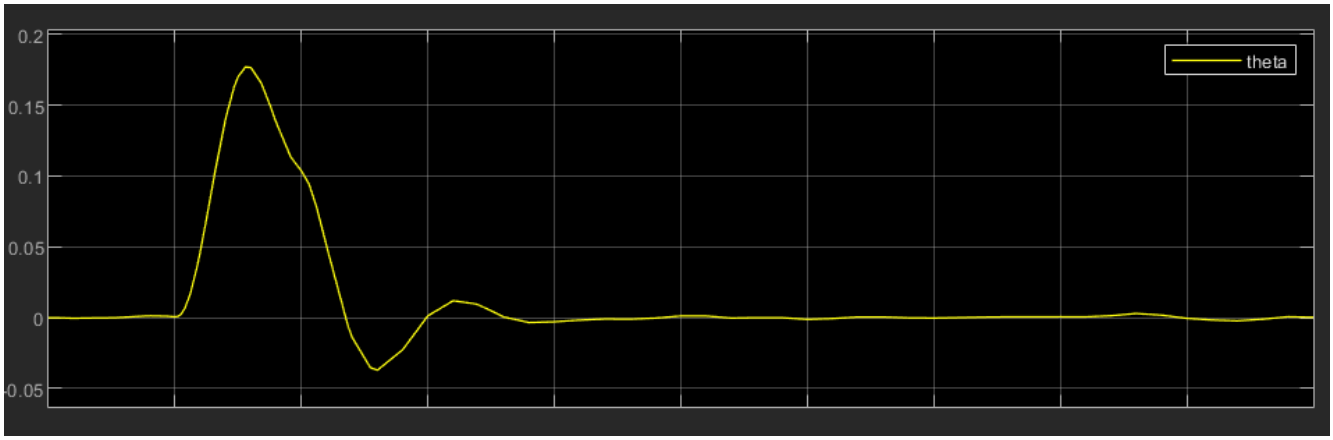
Parameter	Physical meaning	Value	Company
m	Quality	0.87	Kg
g	Acceleration of gravity	9.81	m/s^2
ρ	Air density	1.225	Kg/m^3
h	The vertical distance between the center of gravity and the plane of propellers	0.085	m
l	Horizon distance between rotor center and center of mass	0.175	m
K_T	Lift coefficient	6.46	Ns^2/rad^2
Ix	Inertia on the X axis	0.0043	$kg{\cdot}m^2$
I_y	Inertia on the Y axis	0.0142	$kg{\cdot}m^2$
I_z	Inertia on the Z axis	0.0176	$kg{\cdot}m^2$
I_{xz}	Inertia product on XZ plane	0.0001	$kg{\cdot}m^2$

We gave an impulse of 1 rad on left angle for 0.1 seconds, and added white gaussian noise of SNR value 50 dB to angle and 25 dB to thrust so that the model will be more realistic. Our simulink diagram was as follows -

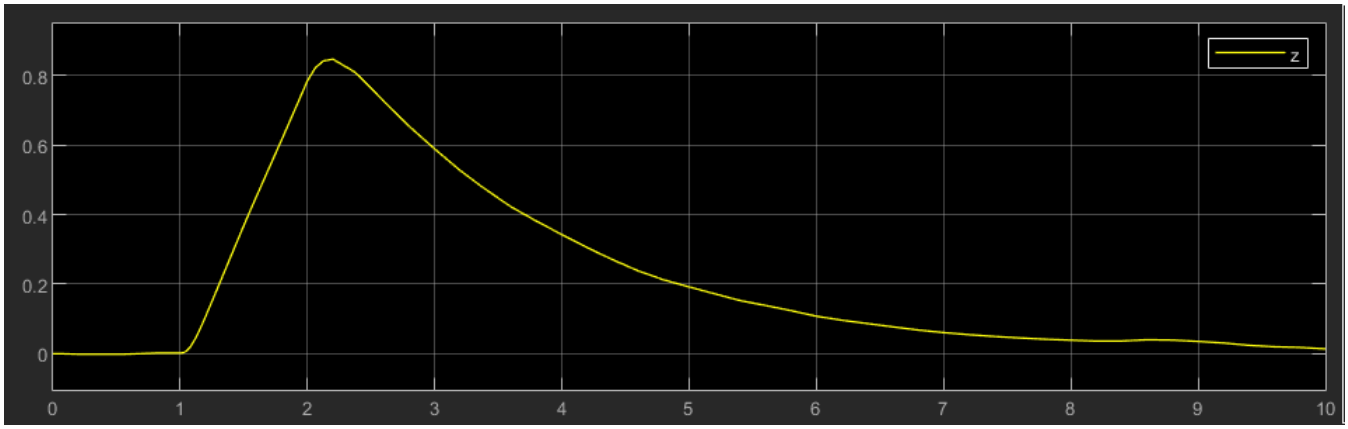


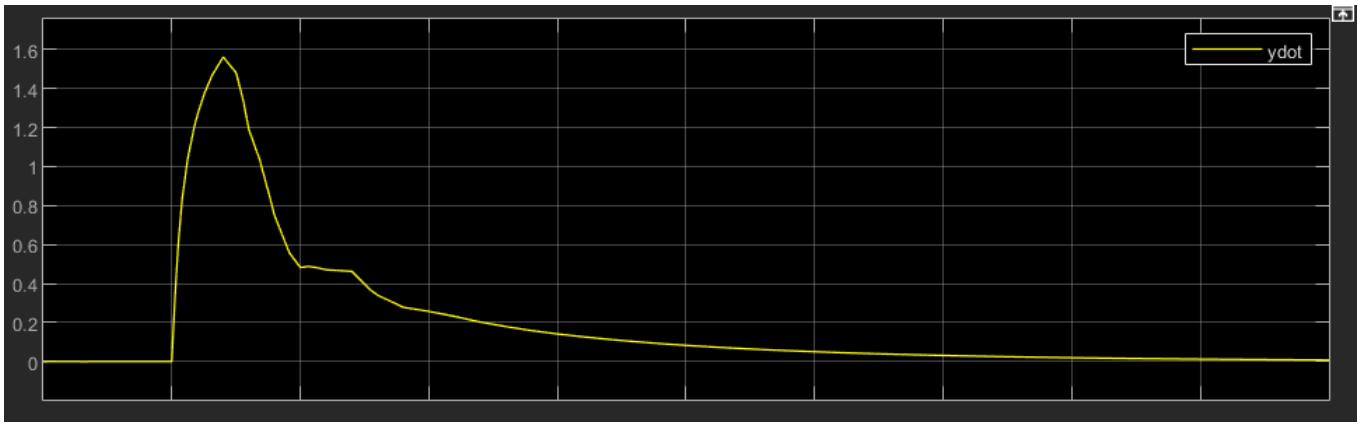
The simulation output were as follows -

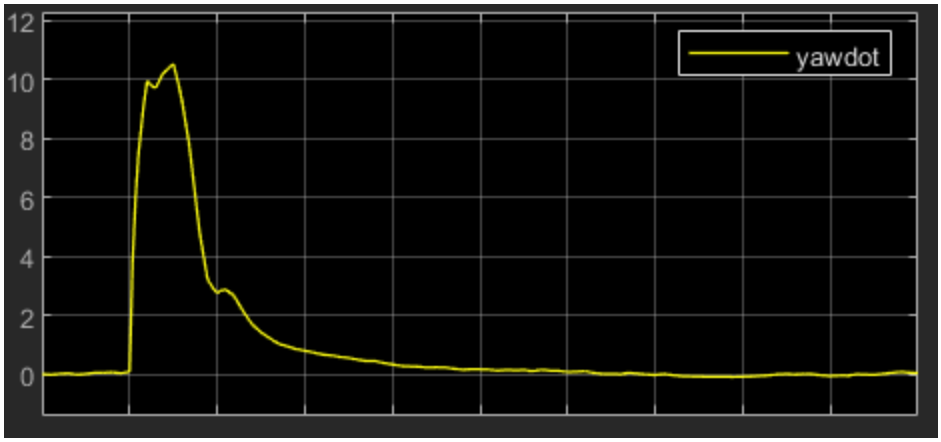




We can see that the theta is getting stabilized very well, the max theta it goes to after the impulse is less than 0.2 rad = 11.4592 degrees

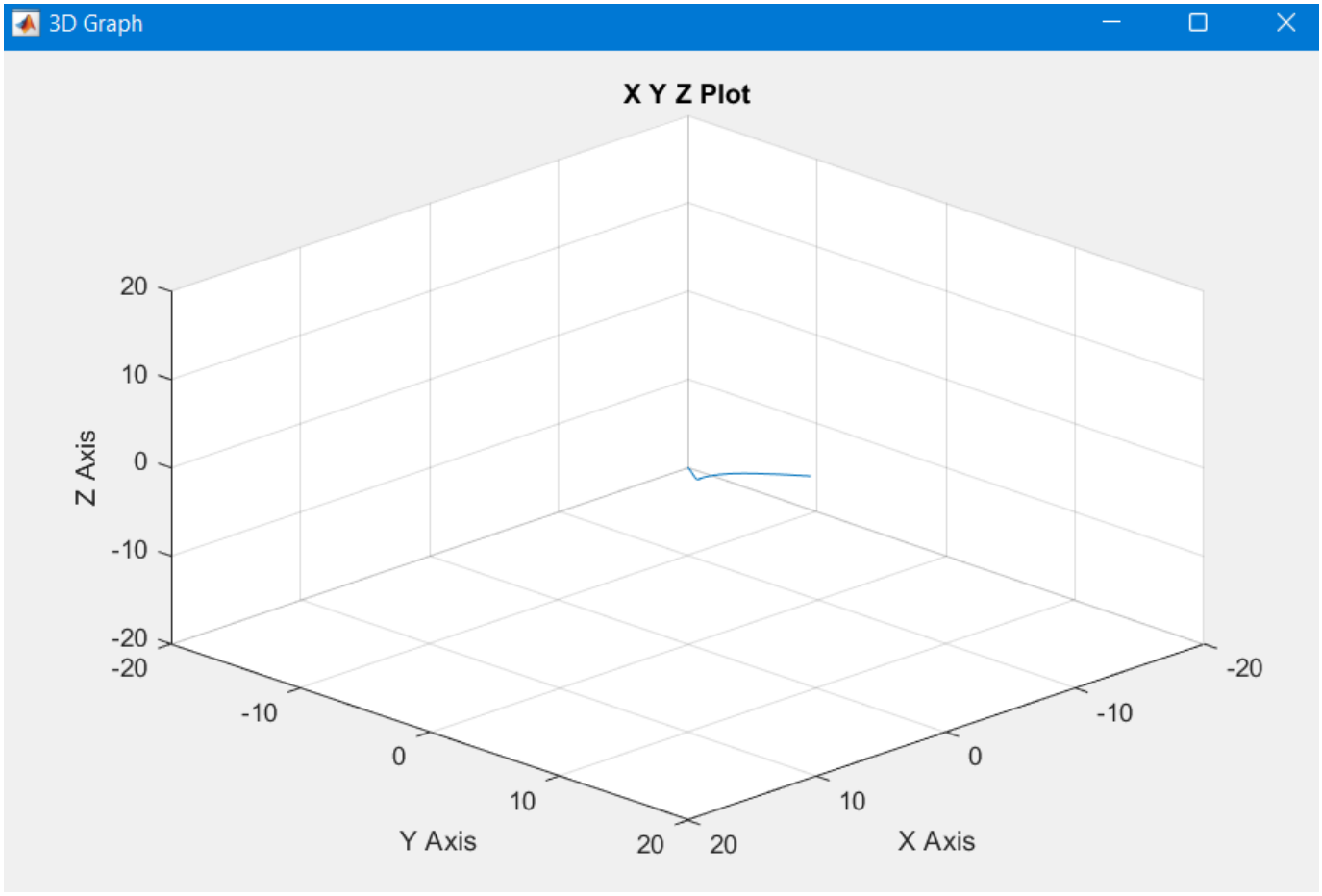






3D PLOT-

We installed another product known as “3DScope” which helps us plot on 3 axes. We used it to get the (x,y,z) coordinates of the bicopter till the simulation runs. We can change the angle of the camera to adjust how we want to look. Here we chose the camera coordinates to be (20,20,20)



As we can see, due to impulse, it changes altitude and tries to move forward, but our controller made bicopter to comeback to its original altitude and also we can see bicopter moving forward, which eventually stops as **ydot** comes to zero

CONCLUSION

We have derived the dynamic equations of Bicopter and used them to make PID and Fuzzy controllers for altitude, roll, pitch and yaw stabilization. The resulting plots show that Fuzzy controllers provide much better stabilization as compared to PID controllers. From the 3D plot we can see that our bicopter gets stabilized after disturbance in form of impulse is applied.

REFERENCES

1. N. Uddin, H. G. Harno and R. A. Sasongko, "Altitude Control System Design of Bicopter Using Lyapunov Stability Approach," *2021 International Symposium on Electronics and Smart Devices (ISESD)*, 2021, pp. 1-6, doi: 10.1109/ISESD53023.2021.9501544.
2. N. Uddin, H. G. Harno, A. Manurung and M. Nasucha, "A Pitch Control System Design for Bicopter UAVs," *2021 8th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, 2021, pp. 57-62, doi: 10.1109/ICITACEE53184.2021.9617535.
3. Achour, Zina & REZOUG, Amar & Hamerlain, Mustapha. (2015). Adaptive PID + Fuzzy PID Controller for Birotor UAV System.
4. Ö. B. Albayrak, Y. Ersan, A. S. Bağbaşı, A. Turgut Başaranoğlu and K. B. Arıkan, "Design of a Robotic Bicopter," 2019 7th International Conference on Control, Mechatronics and Automation (ICCMA), 2019, pp. 98-103, doi: 10.1109/ICCMA46720.2019.8988694.