

Simulating **MONOCOPTER**

Team

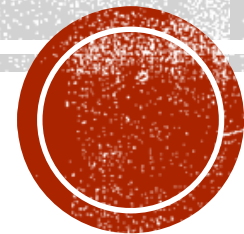
Maha Ezzat

Shrouk Shalaby

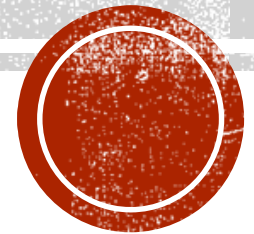
Yomna Sherif

Mai Tarek

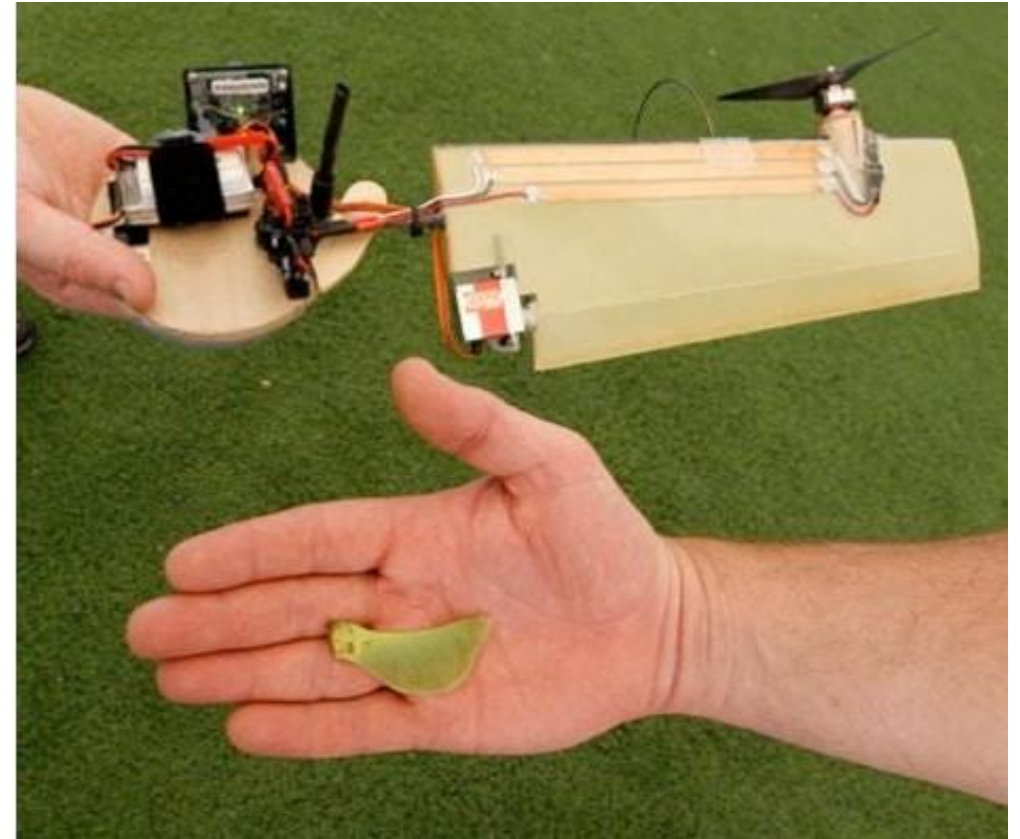
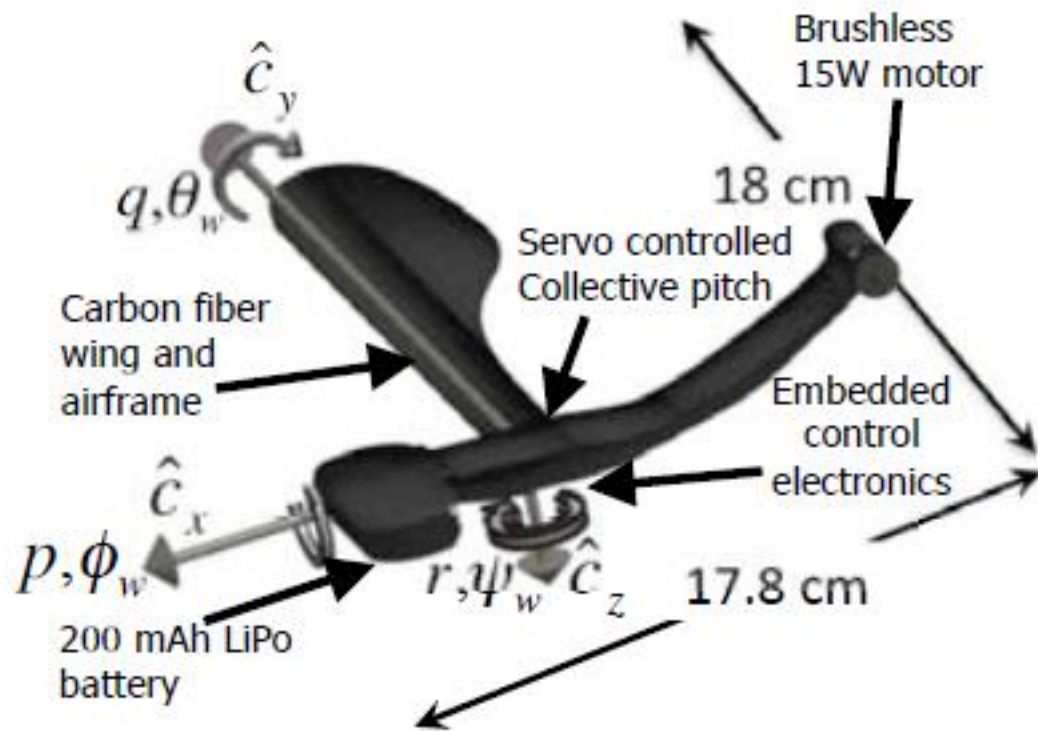
Eman Abulmagd



INTRODUCTION



MONOCOPTER



DERIVING STATE SPACE



STATE SPACE

A =

1.0e+03 *

0.0006	0.0012	0	0
0	0	0.0002	-0.0410
0	0	-0.0029	1.0116
0	0	-0.0000	0

C =

1	0	0	0
---	---	---	---

B =

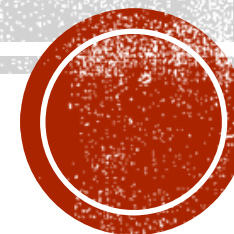
114.3500
38.9420
0
-2.3455

D =

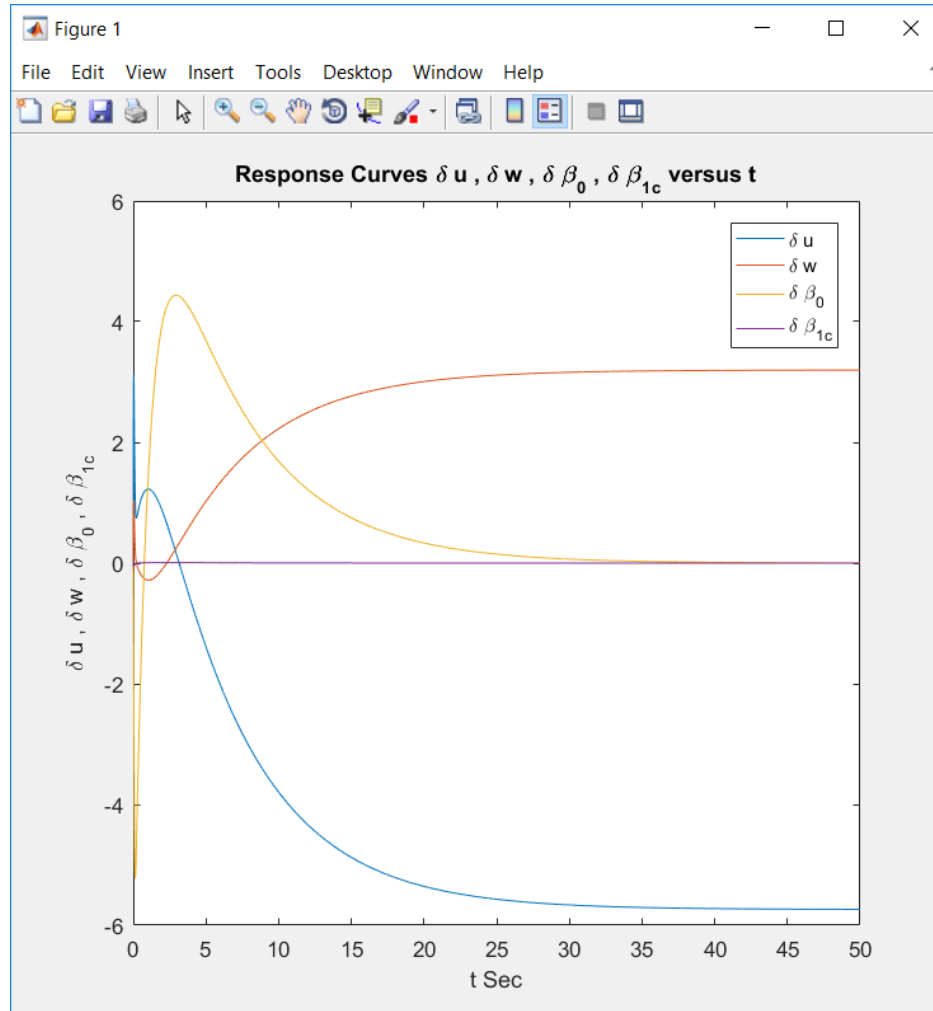
0



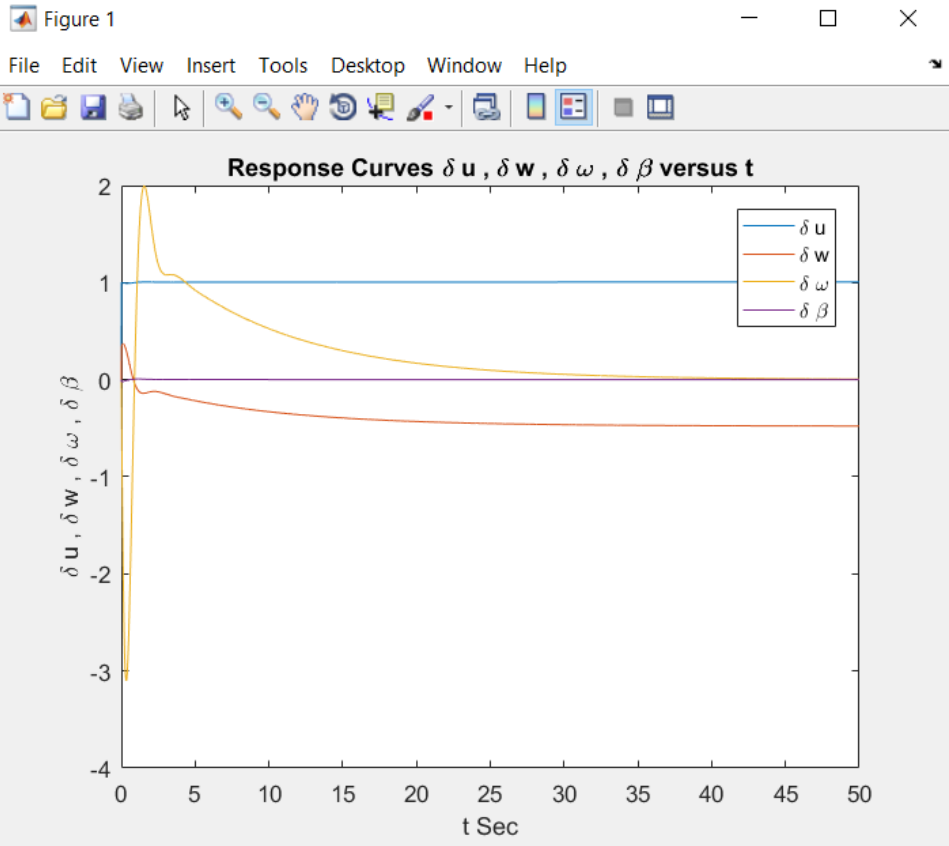
LQR CONTROL



UN-TUNED LQR PARAMETERS



TUNED LQR PARAMETERS



$Q =$

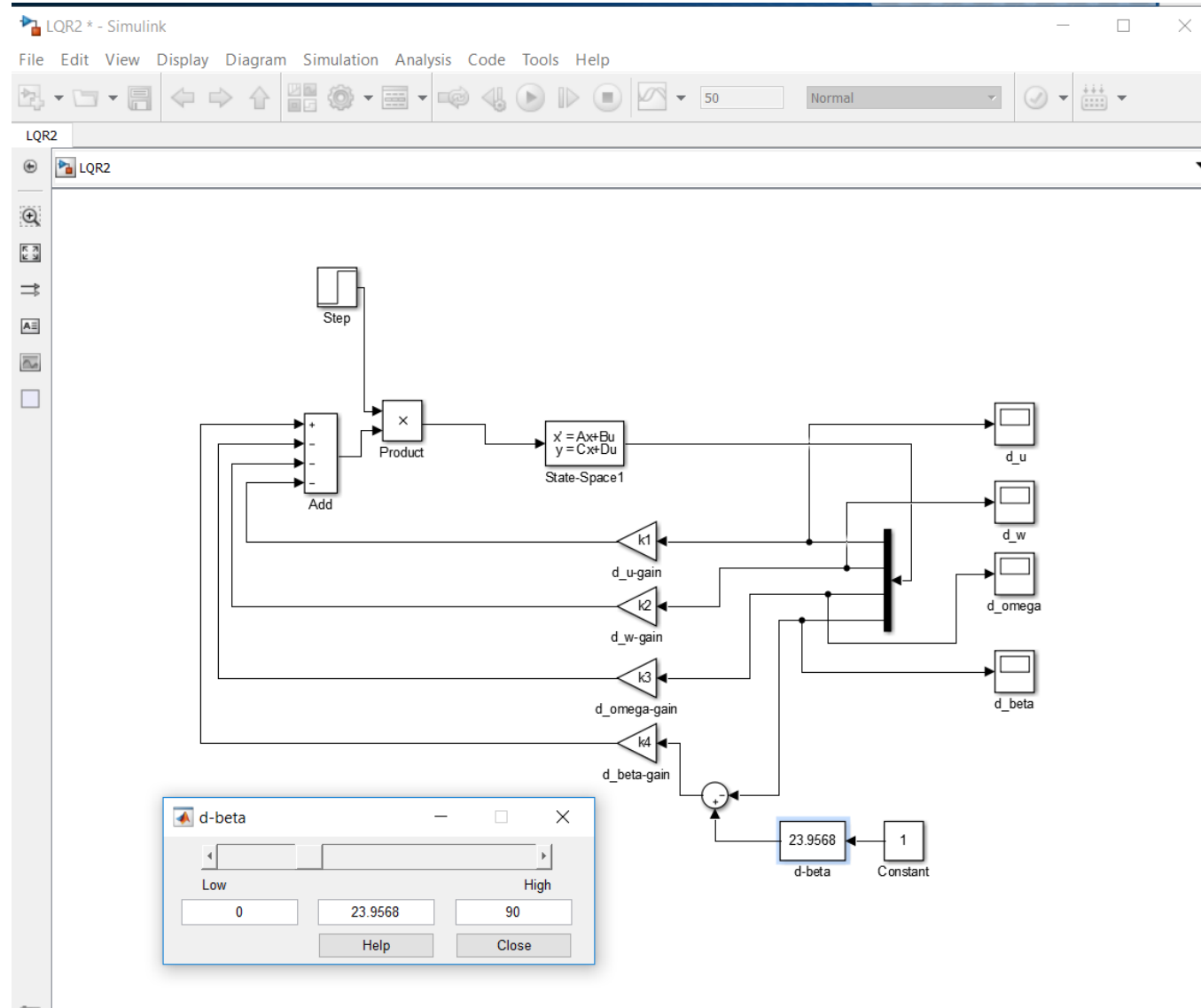
100.0000	0	0	0
0	0.0100	0	0
0	0	0.0100	0
0	0	0	0.0100

$R =$

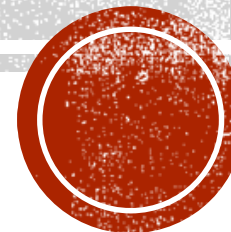
1

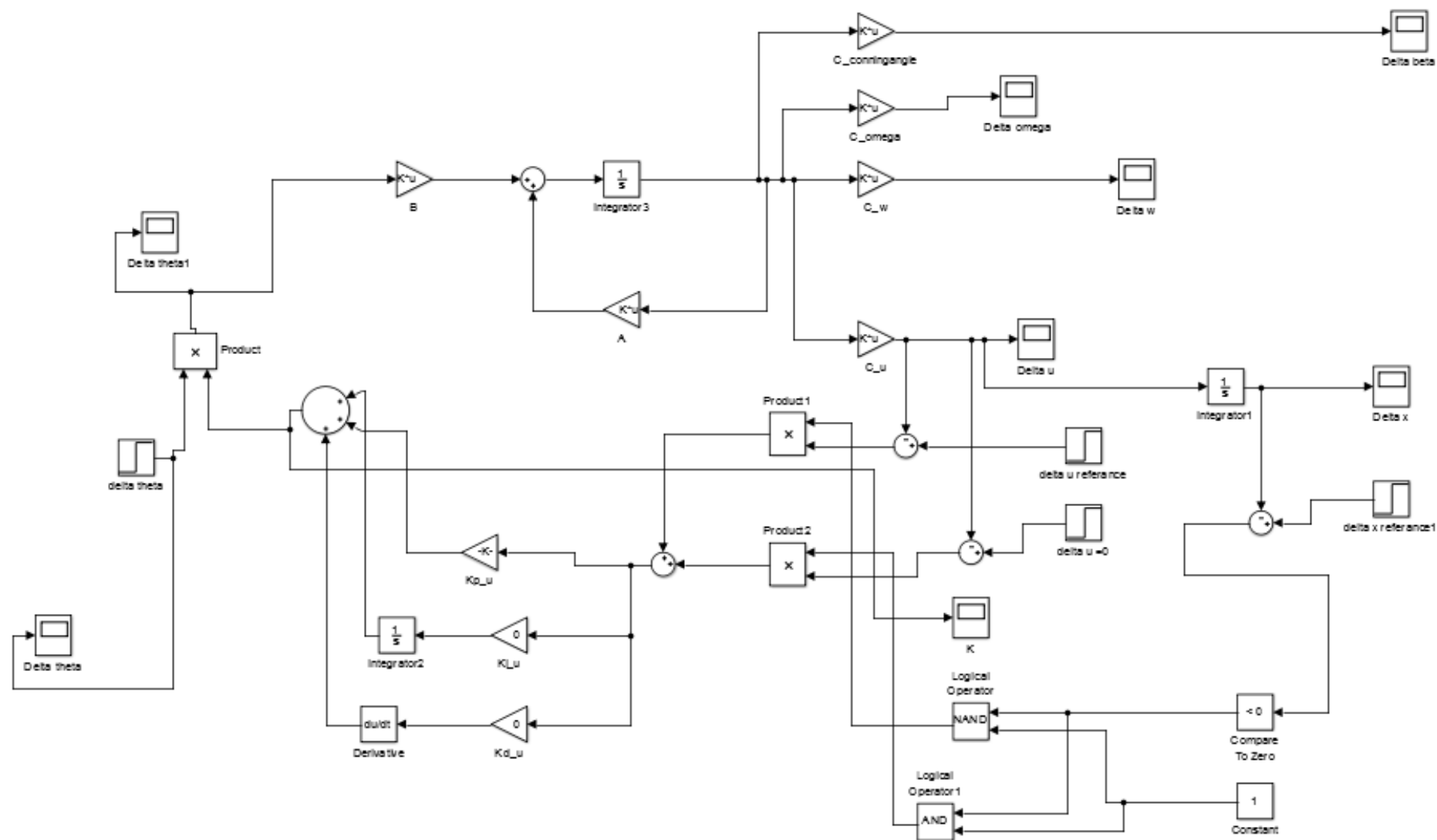


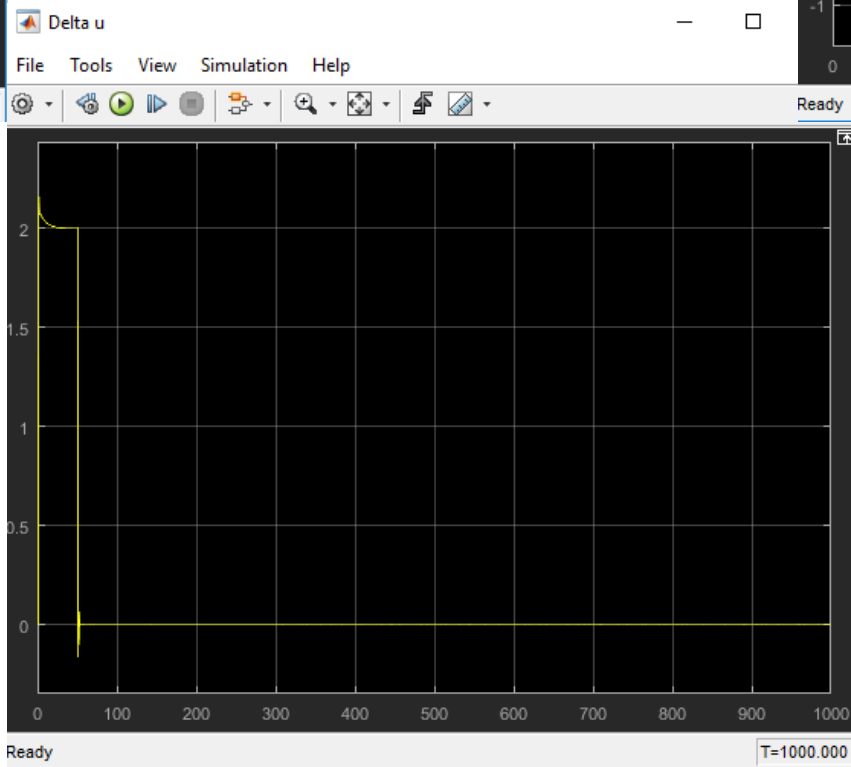
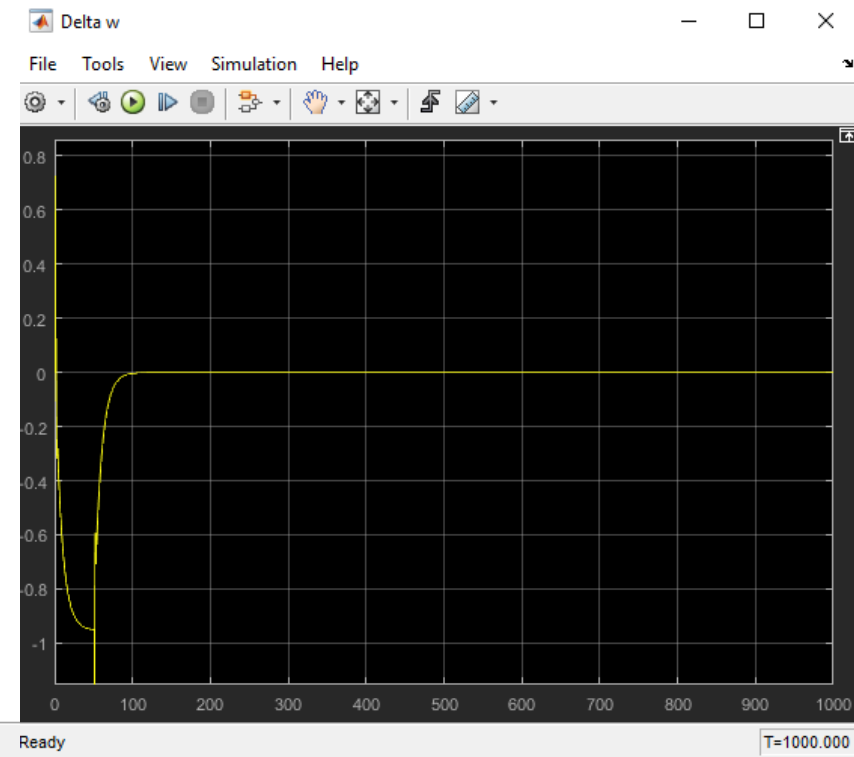
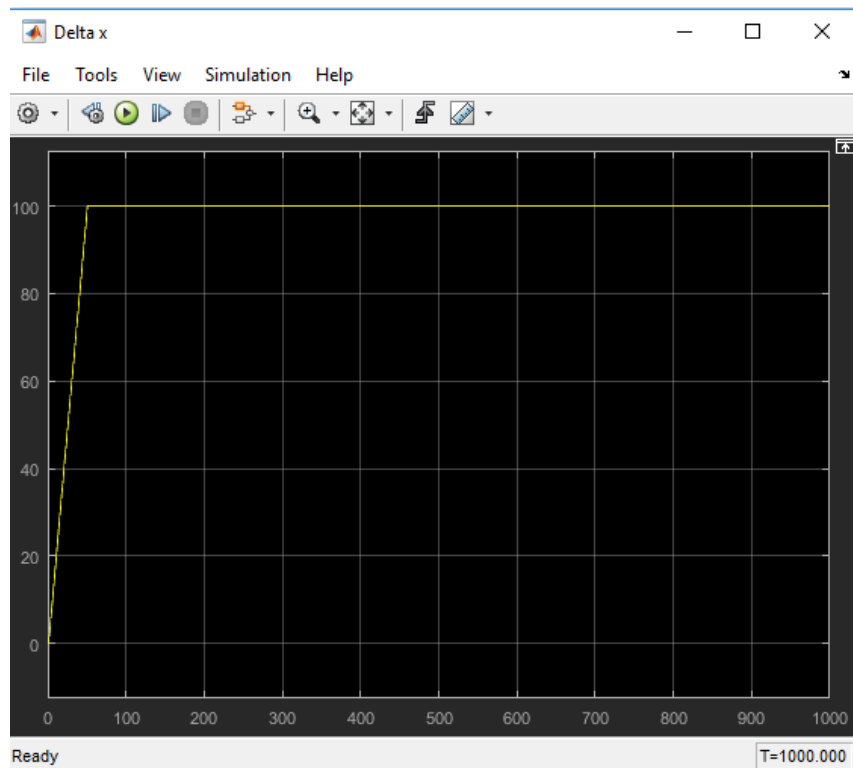
TUNED LQR PARAMETERS

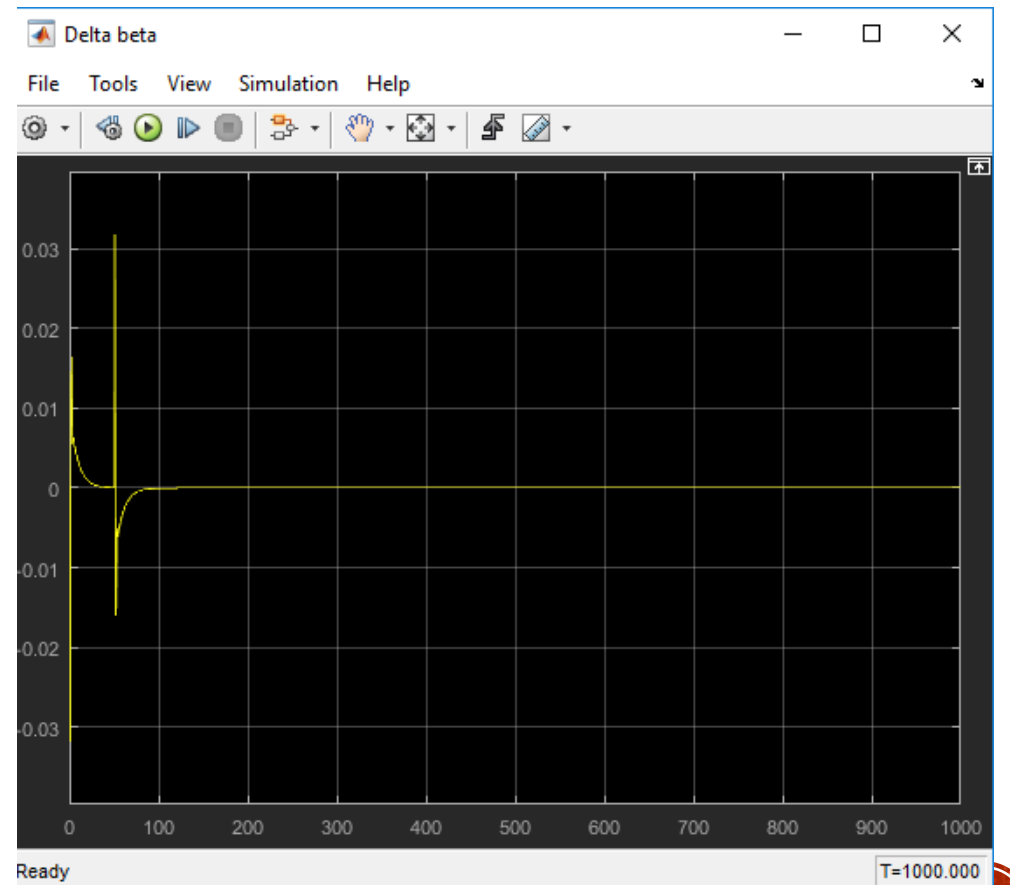
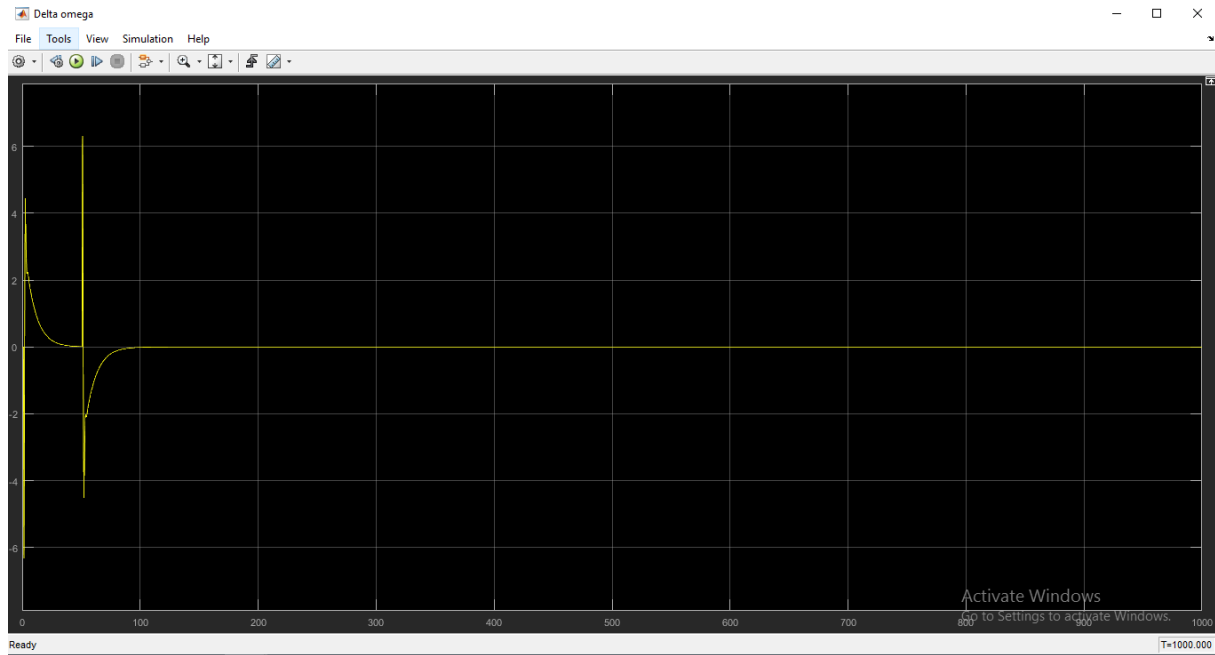


PID CONTROL

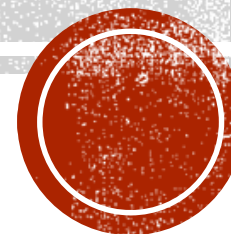




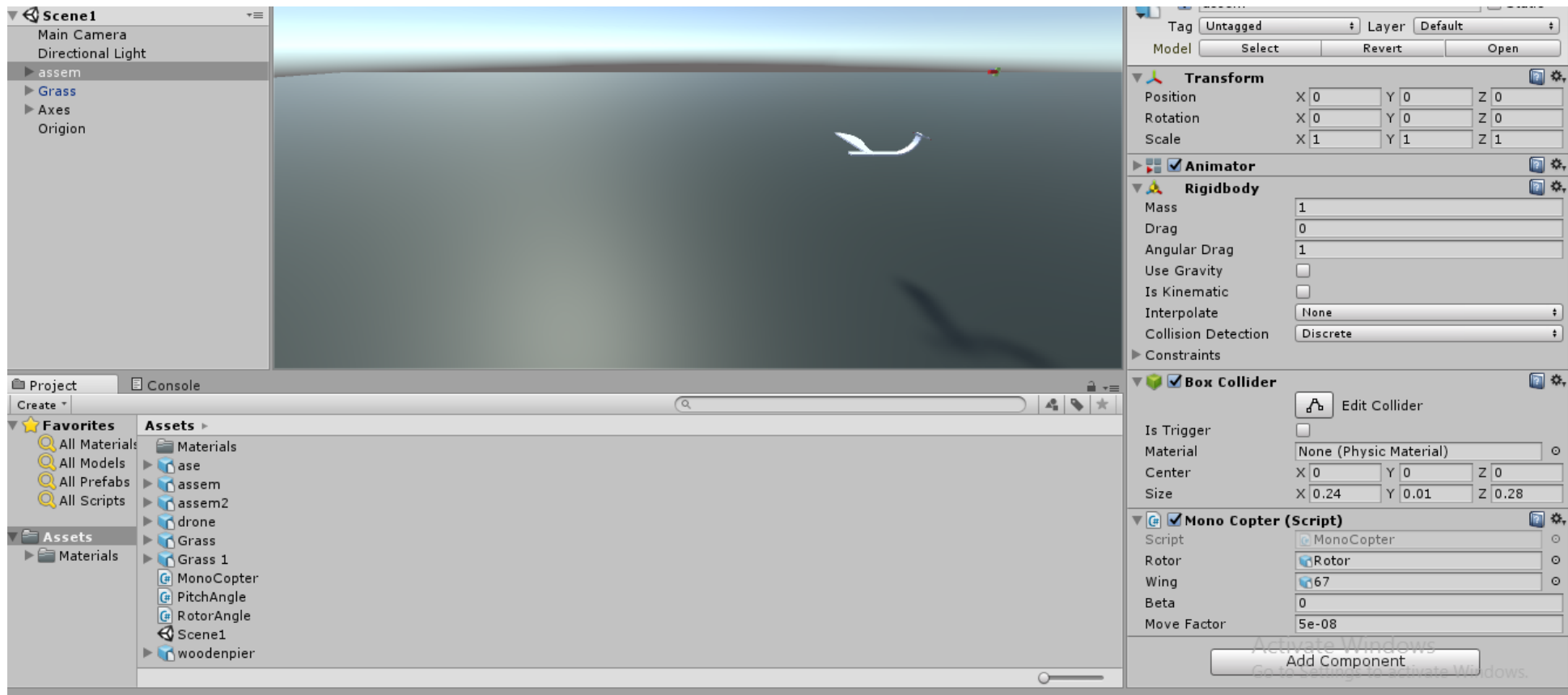




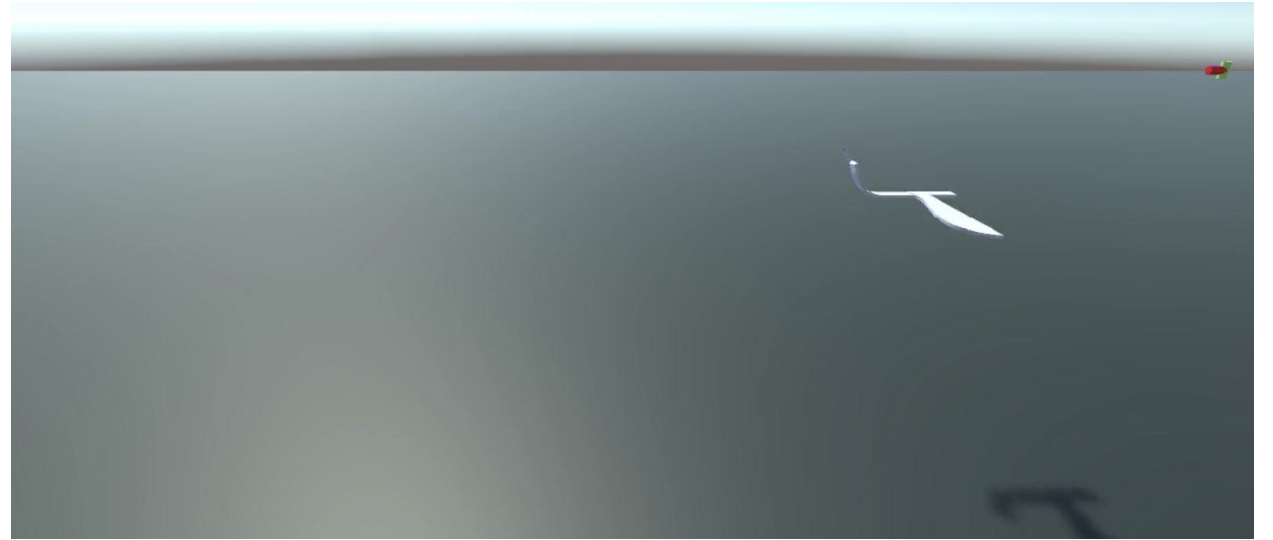
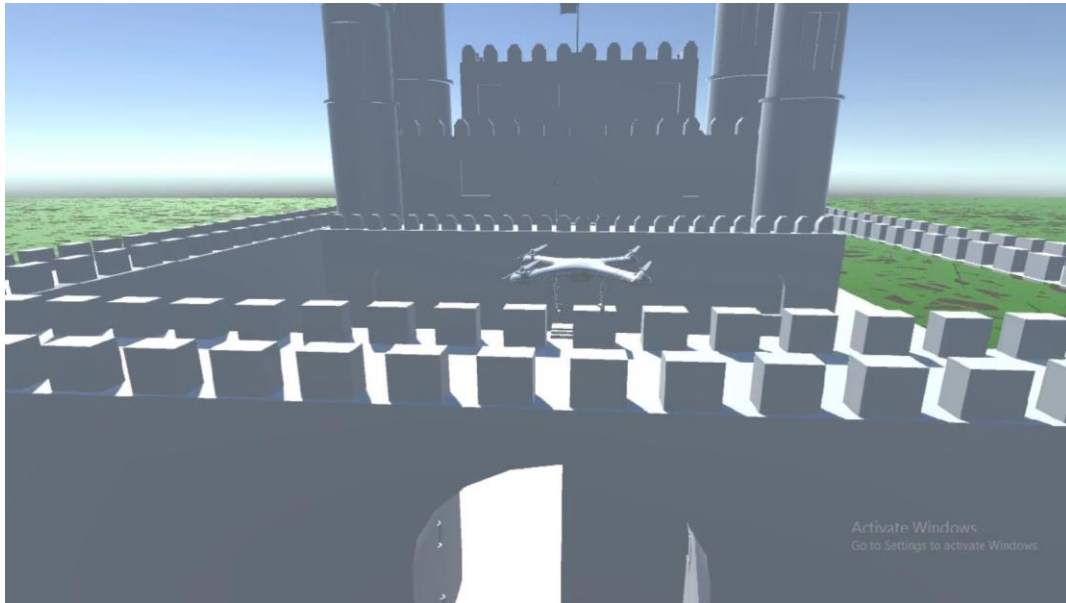
UNITY SIMULATION



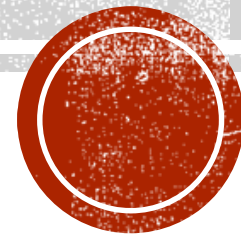
UNITY 3D



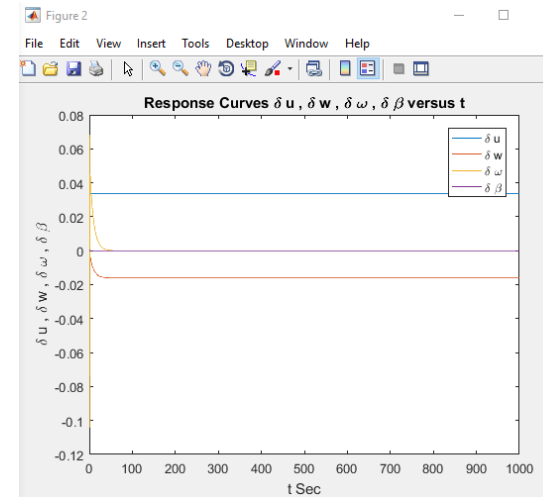
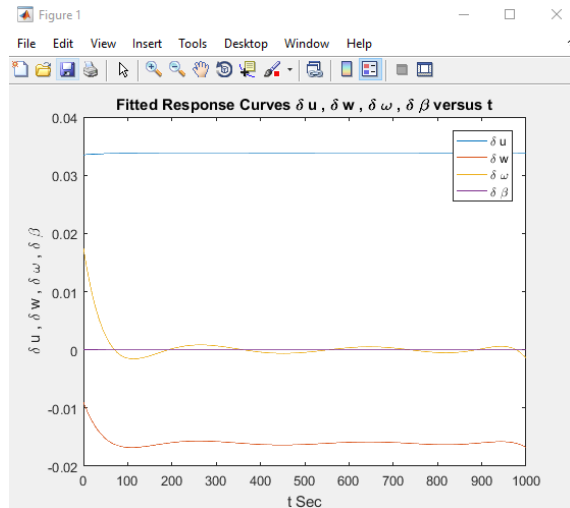
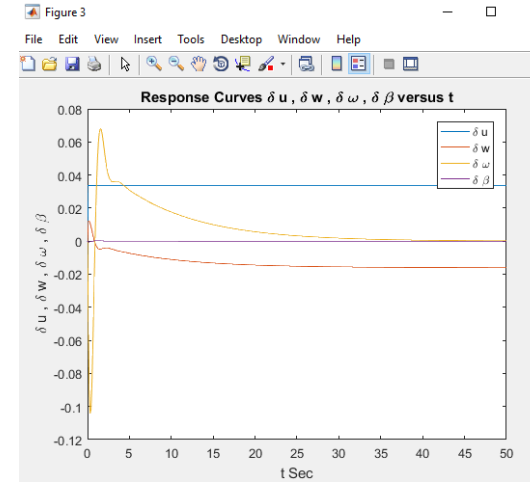
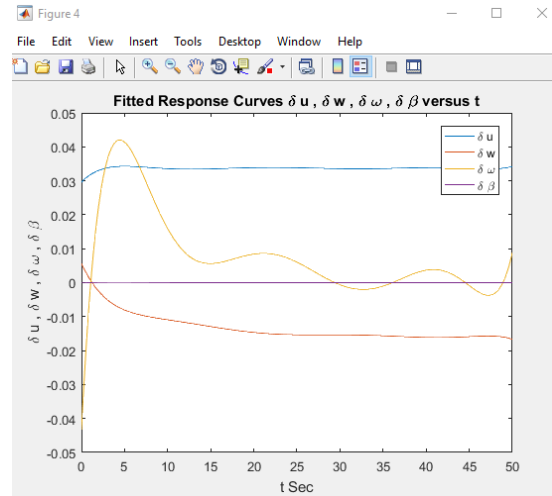
QUADCOPTER & MONOCOPTER SIMULATION



SIMULATION CODE C#



LQR TO C#



C# CODE

```
void AddMotorForce()
{
    Vector3 propellerUp = quadcopterRB.transform.up;

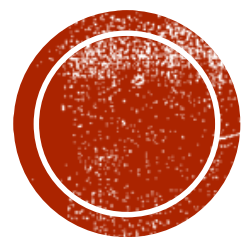
    Vector3 propellerPos = quadcopterRB.transform.position;

    quadcopterRB.AddTorque(transform.up * Omega);
    quadcopterRB.AddForceAtPosition(Vector3.up * moveFactor*deltaW, propellerPos); //force in y
    quadcopterRB.AddForceAtPosition(Vector3.right * moveFactor*deltaU, propellerPos); //force in x
}

void StatesCal()
{
    if (k == 0)
    {
        deltaU0 = 0;
        deltaW0 = 0;
        t = Time.realtimeSinceStartup;
    }
    else
    {
        t = Time.realtimeSinceStartup - t1;

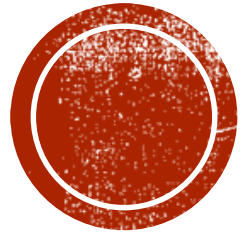
        deltaU = (1.7491f * Mathf.Pow(10f, -22f)) - (6.5305f * Mathf.Pow(10f, -19f)) * t + (9.7967f * Mathf.Pow(10f, -16f)) * Mathf.Pow(t, 2f) - (7.5367f * Mathf.Pow(10f, -13f)) * Mathf.Pow(t, 3f) + (2.5367f * Mathf.Pow(10f, -10f)) * Mathf.Pow(t, 4f);
        deltaW = (-4.5908f * Mathf.Pow(10f, -21f)) + (1.7273f * Mathf.Pow(10f, -17f)) * t - (2.6170f * Mathf.Pow(10f, -14f)) * Mathf.Pow(t, 2f) + (2.0402f * Mathf.Pow(10f, -11f)) * Mathf.Pow(t, 3f) - (0.5367f * Mathf.Pow(10f, -8f)) * Mathf.Pow(t, 4f);
        deltaOmega = (-1.0143f * Mathf.Pow(10f, -20f)) + (3.8333f * Mathf.Pow(10f, -17f)) * t - (5.8407f * Mathf.Pow(10f, -14f)) * Mathf.Pow(t, 2f) + (4.5873f * Mathf.Pow(10f, -11f)) * Mathf.Pow(t, 3f) - (0.5367f * Mathf.Pow(10f, -8f)) * Mathf.Pow(t, 4f);
        Omega = Omega + deltaOmega * Time.fixedDeltaTime;
        t1 = t;
    }
}
```





QUESTIONS TIME !





THANK YOU

