F-16 Airplane MIMO System for Flight Conditions-2

1. Discretization (T_s=0.1 sec) and design of analog and digital tracking control laws (lqr and dlqr);
2. Simulations of analog and discrete systems with y= [θ ϕ ψ]

Continuous Time-System:

Initially the airplane had 8 states and 6 controls with which the flight systems can be stabilized and controlled. The states space form is as follows $\dot{x}(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$ from which the following A and B matrices can be formed as shown below

$$
\begin{bmatrix} \dot{\theta}_P \\ \dot{u}_P \\ \dot{\alpha}_P \\ \dot{q}_P \\ \dot{\phi}_P \\ \dot{\beta}_P \\ \dot{p}_P \\ \dot{r}_P \end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\acute{X}_\theta & \acute{X}_u & \acute{X}_\alpha & \acute{X}_q & 0 & 0 & 0 & 0 \\
\acute{Z}_\theta & \acute{Z}_u & \acute{Z}_\alpha & \acute{Z}_q & 0 & 0 & 0 & 0 \\
\acute{M}_\theta & \acute{M}_u & \acute{M}_\alpha & \acute{M}_q & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & \acute{Y}_\phi & \acute{Y}_\beta & \acute{Y}_p & \acute{Y}_r \\
0 & 0 & 0 & 0 & 0 & \acute{L}_\beta & \acute{L}_p & \acute{L}_r \\
0 & 0 & 0 & 0 & 0 & \acute{N}_\beta & \acute{N}_p & \acute{N}_r
\end{bmatrix}
\begin{bmatrix} \theta_P \\ u_P \\ \alpha_P \\ q_P \\ \phi_P \\ \beta_P \\ p_P \\ r_P \end{bmatrix} +
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0.5\acute{X}_{\delta_e} & 0.5\acute{X}_{\delta_e} & 0.5\acute{X}_{\delta_f} & 0.5\acute{X}_{\delta_f} & 0 & 0 \\
0.5\acute{Z}_{\delta_e} & 0.5\acute{Z}_{\delta_e} & 0.5\acute{Z}_{\delta_f} & 0.5\acute{Z}_{\delta_f} & 0 & 0 \\
0.5\acute{M}_{\delta_e} & 0.5\acute{M}_{\delta_e} & 0.5\acute{M}_{\delta_f} & 0.5\acute{M}_{\delta_f} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0.5\acute{Y}_{\delta_{DT}} & -0.5\acute{Y}_{\delta_{DT}} & 0.5\acute{Y}_{\delta_a} & -0.5\acute{Y}_{\delta_a} & Y_{\delta_c} & Y_{\delta_r} \\
0.5\acute{L}_{\delta_{DT}} & -0.5\acute{L}_{\delta_{DT}} & 0.5\acute{L}_{\delta_a} & -0.5\acute{L}_{\delta_a} & L_{\delta_c} & L_{\delta_r} \\
0.5\acute{N}_{\delta_{DT}} & -0.5\acute{N}_{\delta_{DT}} & 0.5\acute{N}_{\delta_a} & -0.5\acute{N}_{\delta_a} & N_{\delta_c} & N_{\delta_r}
\end{bmatrix} \times
\begin{bmatrix} \delta_{e_R} \\ \delta_{e_L} \\ \delta_{f_R} \\ \delta_{f_L} \\ \delta_c \\ \delta_r \end{bmatrix}
$$

But for our control we had to rearrange and add an extra state as per the given problem. Now the state space model will change to 9 states and 6 controls as shown below

$$
\begin{bmatrix} \dot{V} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} =
\begin{bmatrix}
\acute{X}_v & \acute{X}_\alpha & \acute{X}_q & \acute{X}_\theta & 0 & 0 & 0 & 0 & 0 \\
\acute{Z}_v & \acute{Z}_\alpha & \acute{Z}_q & \acute{Z}_\theta & 0 & 0 & 0 & 0 & 0 \\
\acute{M}_v & \acute{M}_\alpha & \acute{M}_q & \acute{M}_\theta & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \acute{Y}_\beta & \acute{Y}_p & \acute{Y}_r & \acute{Y}\phi & 0 \\
0 & 0 & 0 & 0 & \acute{L}_\beta & \acute{L}_p & \acute{L}_r & 0 & 0 \\
0 & 0 & 0 & 0 & \acute{N}_\beta & \acute{N}_p & \acute{N}_r & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix} V \\ \alpha \\ q \\ \theta \\ \beta \\ p \\ r \\ \phi \\ \psi \end{bmatrix} +
\begin{bmatrix}
0.5\acute{X}_{\delta e} & 0.5\acute{X}_{\delta e} & 0.5\acute{X}_{\delta f} & 0.5\acute{X}_{\delta f} & 0 & 0 \\
0.5\acute{Z}_{\delta e} & 0.5\acute{Z}_{\delta e} & 0.5\acute{Z}_{\delta f} & 0.5\acute{Z}_{\delta f} & 0 & 0 \\
0.5\acute{M}_{\delta e} & 0.5\acute{M}_{\delta e} & 0.5\acute{M}_{\delta f} & 0.5\acute{M}_{\delta f} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0.5\acute{Y}_{\delta DT} & -0.5\acute{Y}_{\delta DT} & 0.5\acute{Y}_{\delta a} & -0.5\acute{Y}_{\delta a} & Y_{\delta C} & Y_{\delta r} \\
0.5\acute{L}_{\delta DT} & -0.5\acute{L}_{\delta DT} & 0.5\acute{L}_{\delta a} & -0.5\acute{L}_{\delta a} & L_{\delta C} & L_{\delta r} \\
0.5\acute{N}_{\delta DT} & -0.5\acute{N}_{\delta DT} & 0.5\acute{N}_{\delta a} & -0.5\acute{N}_{\delta a} & N_{\delta C} & N_{\delta r} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \times
\begin{bmatrix} \delta_{e_R} \\ \delta_{e_L} \\ \delta_{f_R} \\ \delta_{f_L} \\ \delta_c \\ \delta_r \end{bmatrix}
$$

Where V is forward velocity (m/sec), α is angle of attack (rad), q is pitch rate (rad/sec), θ is pitch angle (rad), β is sideslip angle (rad), p is roll rate (rad/sec), r is yaw rate (rad/sec), ϕ is bank angle (rad) and ψ is and the outputs are pitch angle, bank angle and . Similarly, the control inputs are δ_{eR} is right horizontal tail (rad), δ_{eL} is left horizontal tail (rad), δ_{fR} is right flaperon (rad), δ_{fL} is left flaperon (rad), δ_e is canards (rad) and δ_f is rudder (rad). Now based on the flight condition 2 the airplane is cruising at a speed of Mach 0.6 and at a height of 30,000 ft at which we can study the variation of elements within the model. After substituting the dimensional derivative elements in the obtained A and B matrices we get the following (since the given values are in ft/sec we need to convert it m/sec by dividing first row, first element of second and third of A with 3.28. Similarly, multiplying the first row of B matrix with 3.28)

```
A =

    0.0016    7.0245  -14.9020   -9.7840         0         0         0         0         0
   -0.0000   -0.5264    0.9972   -0.0044         0         0         0         0         0
   -0.0001    2.5271   -0.3419    0.0003         0         0         0         0         0
         0         0    1.0000         0         0         0         0         0         0
         0         0         0         0   -0.1541    0.0824   -0.9983    0.0538         0
         0         0         0         0  -19.2246   -0.8936    0.3188         0         0
         0         0         0         0    2.2958   -0.0009   -0.2787         0         0
         0         0         0         0         0    1.0000         0         0         0
         0         0         0         0         0         0    1.0000         0         0
```

```
B =

    5.1994    5.1994   -3.4416   -3.4416         0         0
   -0.0331   -0.0331   -0.0559   -0.0559         0         0
   -2.9311   -2.9311   -0.1059   -0.1059         0         0
         0         0         0         0         0         0
    0.0072   -0.0072    0.0002   -0.0002    0.0073    0.0212
   -6.7916    6.7916   -8.7234    8.7234    0.4145    3.9232
   -0.7527    0.7527   -0.1342    0.1342    1.5101   -1.9665
         0         0         0         0         0         0
         0         0         0         0         0         0
```

Similarly, the output matrix H is defined below

```
H =

     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     1
```

Now we need to find the extended state space model as shown Below

$$\mathbf{As} = \begin{bmatrix} A & 0 & 0 \\ -I_X H & 0 & 0 \\ -HA & I_E^X & -I_E \end{bmatrix} \qquad \mathbf{Bs} = \begin{bmatrix} B \\ 0 \\ -HB \end{bmatrix}$$

i.e. the following are written according to the problems matrix dimensions as follows

As = [A zeros(9,6); -IX*H zeros(3,6); -H*A zeros(3,3) -IE];

Bs = [B; zeros(3,6); -H*B];

And the matrices As, Bs and Hs are as shown below

```
As =

    0.0016    7.0245  -14.9020   -9.7840         0         0         0         0         0         0         0         0         0         0         0
   -0.0000   -0.5264    0.9972   -0.0044         0         0         0         0         0         0         0         0         0         0         0
   -0.0001    2.5271   -0.3419    0.0003         0         0         0         0         0         0         0         0         0         0         0
         0         0    1.0000         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0   -0.1541    0.0824   -0.9983    0.0538         0         0         0         0         0         0         0
         0         0         0         0  -19.2246   -0.8936    0.3188         0         0         0         0         0         0         0         0
         0         0         0         0    2.2958   -0.0009   -0.2787         0         0         0         0         0         0         0         0
         0         0         0         0         0    1.0000         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0    1.0000         0         0         0         0         0         0         0         0
         0         0         0   -1.0000         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0   -1.0000         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0   -1.0000         0         0         0         0         0         0
         0         0   -1.0000         0         0         0         0         0         0         0         0         0   -1.0000         0         0
         0         0         0         0         0   -1.0000         0         0         0         0         0         0         0   -1.0000         0
         0         0         0         0         0         0   -1.0000         0         0         0         0         0         0         0   -1.0000
```

Bs =

5.1994	5.1994	-3.4416	-3.4416	0	0
-0.0331	-0.0331	-0.0559	-0.0559	0	0
-2.9311	-2.9311	-0.1059	-0.1059	0	0
0	0	0	0	0	0
0.0072	-0.0072	0.0002	-0.0002	0.0073	0.0212
-6.7916	6.7916	-8.7234	8.7234	0.4145	3.9232
-0.7527	0.7527	-0.1342	0.1342	1.5101	-1.9665
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Hs =

0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

I_X, I_E and I_E^x specify the error state and tracking error dynamics and chosen as identity matrix as

IX =

1	0	0
0	1	0
0	0	1

IE =

1	0	0
0	1	0
0	0	1

In order to minimize the function below we must choose and vary Q and G value

$$J = \min_{u} \int_{t_0}^{t_f} \tfrac{1}{2}\left(\mathbf{x}^T Q \mathbf{x} + u^T G u\right) dt\ ,\ Q \in \mathbb{R}^{(n+2b)\times(n+2b)},\ Q \geq 0,\ G \in \mathbb{R}^{m\times m},\ G > 0,$$

Q and G values are chosen randomly by checking the continuous time signal i.e. how fast it follows the reference signal

Q =

1.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1.0000	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1.0000	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1.0000	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1.0000	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1.0000	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1.0000	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0.5000	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1.0000	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1.0000	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1.0000	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	100.0000	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1.0000	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1.0000	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.0000

G =

0.0100	0	0	0	0	0
0	0.0100	0	0	0	0
0	0	0.0100	0	0	0
0	0	0	0.0100	0	0
0	0	0	0	0.0100	0
0	0	0	0	0	0.0100

Now we need to find the optimal control law which can be found using LQR (Linear-Quadratic Regulator) command in MATLAB which is as shown below

[K,S,Eig]=lqr(As,Bs,Q,G);

And this command gives feedback gain matrix (K), solution matrix i.e. solution for the Riccati equation (S) and the eigen value matrix, the output of lqr command is as shown below for this problem.

K =

4.8924	-1.1485	-5.6273	-10.0069	1.0697	-4.1879	-2.7301	-7.0070	-11.1857	5.1053	4.2002	23.5377	0.5996	0.5549	0.0724
4.8924	-1.1485	-5.6273	-10.0069	-1.0697	4.1879	2.7301	7.0070	11.1857	5.1053	-4.2002	-23.5377	0.5996	-0.5549	-0.0724
-5.1056	-2.3577	-3.5106	-8.3476	0.4773	-5.4426	0.0906	-9.0381	0.2737	4.8924	5.4104	-0.5365	0.5586	0.7203	-0.0018
-5.1056	-2.3577	-3.5106	-8.3476	-0.4773	5.4426	-0.0906	9.0381	-0.2737	4.8924	-5.4104	0.5365	0.5586	-0.7203	0.0018
0.0000	-0.0000	0.0000	0.0000	4.7579	0.1421	6.5362	-0.0827	32.8017	-0.0000	0.0927	-56.9031	0.0000	-0.0125	-0.1720
0.0000	-0.0000	-0.0000	0.0000	1.7447	2.6024	-8.7988	4.1727	-35.9859	0.0000	-2.4824	75.1892	-0.0000	-0.3424	0.2317

S =

0.0146	-0.0043	0.0093	0.0315	0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0162	0.0000	0.0000	-0.0016	-0.0000	0.0000
-0.0043	0.7066	-0.0116	-0.6287	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	0.2124	0.0000	0.0000	0.0104	0.0000	-0.0000
0.0093	-0.0116	0.0358	0.0971	0.0000	0.0000	-0.0000	-0.0000	-0.0000	-0.0485	-0.0000	0.0000	-0.0051	-0.0000	0.0000
0.0315	-0.6287	0.0971	3.2288	0.0000	-0.0000	-0.0000	0.0000	-0.0000	-1.5033	-0.0000	0.0000	0.3451	-0.0000	0.0000
0.0000	-0.0000	0.0000	0.0000	2.6965	-0.0008	0.0186	-0.1869	2.6853	-0.0000	0.1355	-0.5735	-0.0000	0.0035	-0.0003
-0.0000	-0.0000	0.0000	-0.0000	-0.0008	0.0063	-0.0008	0.0104	-0.0034	0.0000	-0.0062	0.0064	0.0000	-0.0008	0.0000
-0.0000	-0.0000	-0.0000	-0.0000	0.0186	-0.0008	0.0434	-0.0025	0.2051	0.0000	0.0017	-0.3758	-0.0000	0.0001	-0.0011
-0.0000	-0.0000	-0.0000	0.0000	-0.1869	0.0104	-0.0025	2.0532	-0.1963	0.0000	-1.1550	0.0123	0.0000	0.3555	-0.0000
-0.0000	-0.0000	-0.0000	-0.0000	2.6853	-0.0034	0.2051	-0.1963	8.7470	0.0000	0.1427	-12.6066	0.0000	0.0042	0.4621
-0.0162	0.2124	-0.0485	-1.5033	-0.0000	0.0000	0.0000	0.0000	0.0000	1.9958	-0.0000	0.0000	-0.1159	-0.0000	-0.0000
0.0000	0.0000	-0.0000	-0.0000	0.1355	-0.0062	0.0017	-1.1550	0.1427	-0.0000	1.6776	-0.0137	-0.0000	-0.1330	0.0000
0.0000	0.0000	0.0000	0.0000	-0.5735	0.0064	-0.3758	0.0123	-12.6066	0.0000	-0.0137	50.9913	-0.0000	-0.0031	-0.3062
-0.0016	0.0104	-0.0051	0.3451	-0.0000	0.0000	-0.0000	0.0000	0.0000	-0.1159	-0.0000	-0.0000	0.4933	-0.0000	-0.0000
-0.0000	0.0000	-0.0000	-0.0000	0.0035	-0.0008	0.0001	0.3555	0.0042	-0.0000	-0.1330	-0.0031	-0.0000	0.4911	-0.0000
0.0000	-0.0000	0.0000	0.0000	-0.0003	0.0000	-0.0011	-0.0000	0.4621	-0.0000	0.0000	-0.3062	-0.0000	-0.0000	0.4995

```
Eig =

   1.0e+02 *

  -0.0024 + 0.0000i
  -0.0057 + 0.0033i
  -0.0057 - 0.0033i
  -0.0069 + 0.0048i
  -0.0069 - 0.0048i
  -0.0088 + 0.0000i
  -0.0101 + 0.0000i
  -0.0142 + 0.0000i
  -0.0186 + 0.0000i
  -0.0239 + 0.0207i
  -0.0239 - 0.0207i
  -0.2350 + 0.0000i
  -0.2668 + 0.0000i
  -0.9459 + 0.0000i
  -1.6128 + 0.0000i
```

As observed in the above matrices the eigen values are all negative. Now to find the Kfeedback matrix we need to use KF=inv(G)*Bs'*S and the output is as follows

KF =

4.8924	-1.1485	-5.6273	-10.0069	1.0697	-4.1879	-2.7301	-7.0070	-11.1857	5.1053	4.2002	23.5377	0.5996	0.5549	0.0724
4.8924	-1.1485	-5.6273	-10.0069	-1.0697	4.1879	2.7301	7.0070	11.1857	5.1053	-4.2002	-23.5377	0.5996	-0.5549	-0.0724
-5.1056	-2.3577	-3.5106	-8.3476	0.4773	-5.4426	0.0906	-9.0381	0.2737	4.8924	5.4104	-0.5365	0.5586	0.7203	-0.0018
-5.1056	-2.3577	-3.5106	-8.3476	-0.4773	5.4426	-0.0906	9.0381	-0.2737	4.8924	-5.4104	0.5365	0.5586	-0.7203	0.0018
-0.0000	-0.0000	-0.0000	-0.0000	4.7579	0.1421	6.5362	-0.0827	32.8017	0.0000	0.0927	-56.9031	-0.0000	-0.0125	-0.1720
-0.0000	-0.0000	0.0000	0.0000	1.7447	2.6024	-8.7988	4.1727	-35.9859	0.0000	-2.4824	75.1892	0.0000	-0.3424	0.2317

Simulink For Continuous Time System with Control Law:

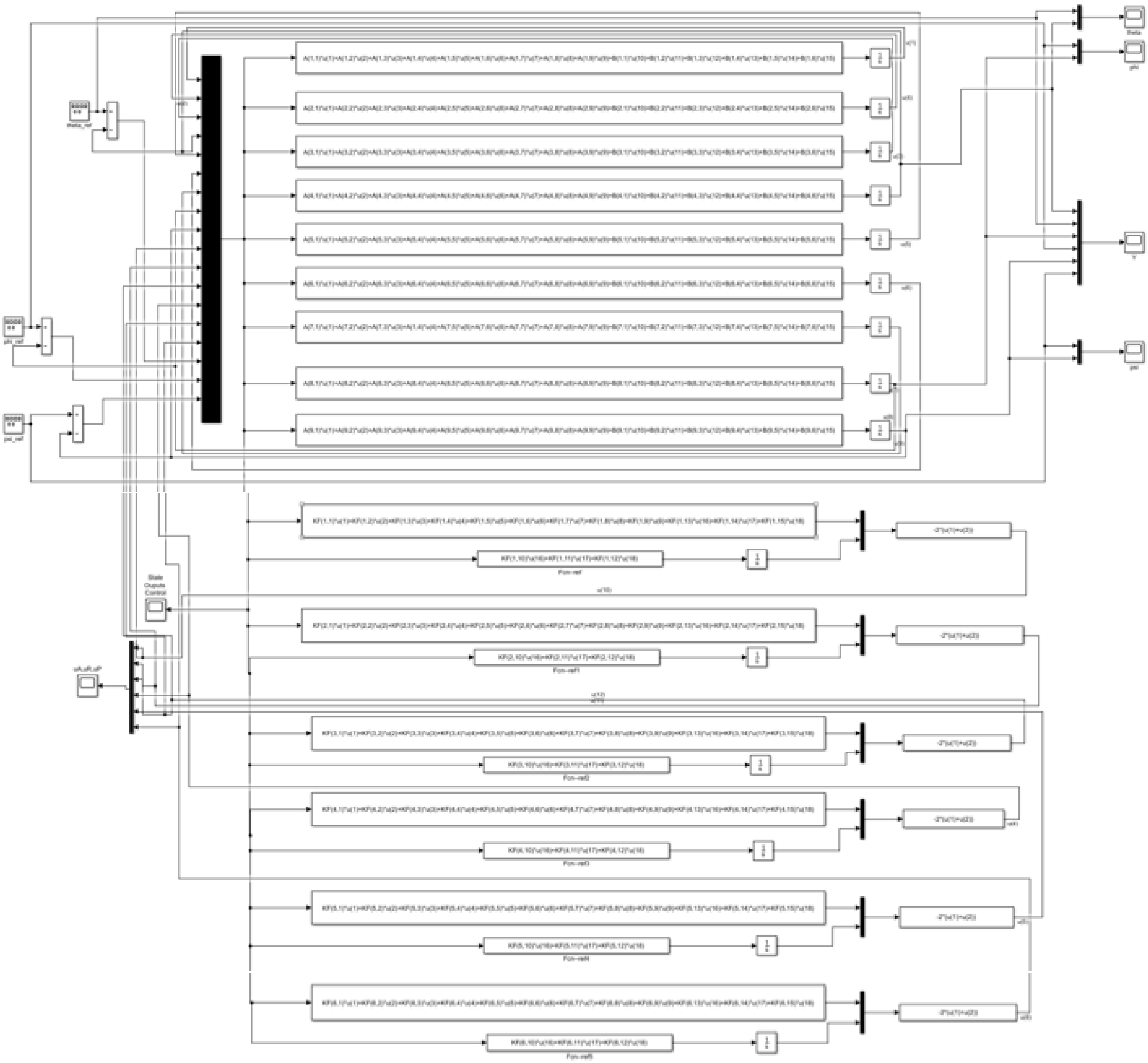

Fig 1: Simulink Diagram

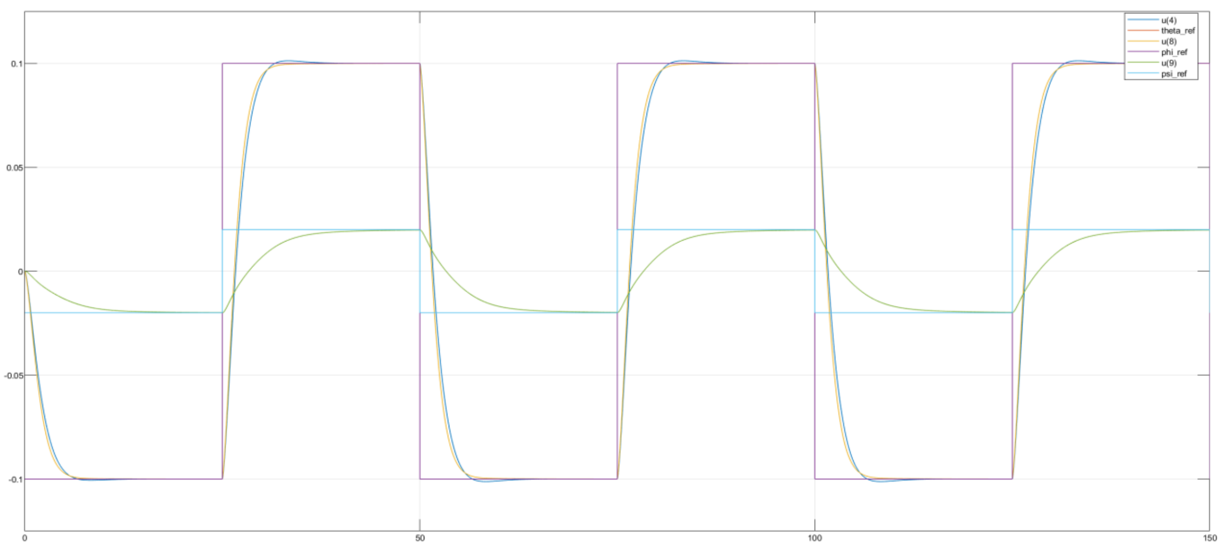

Fig 2: Analog output following the references

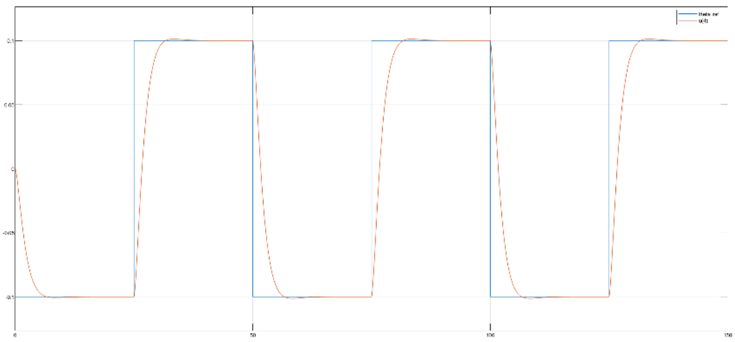

Fig 3: Theta following the reference signal ± 0.1 rad

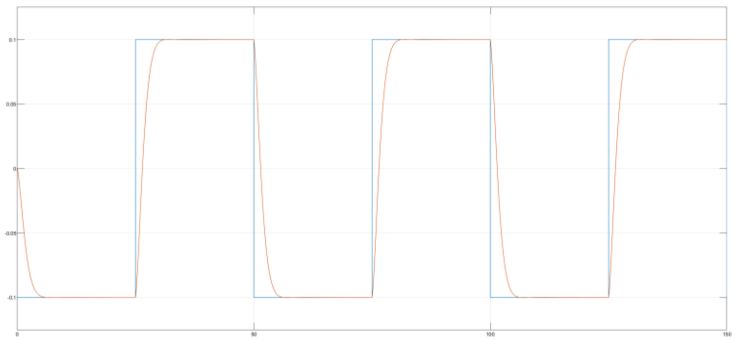

Fig 4: Phi following the reference signal ± 0.1 rad

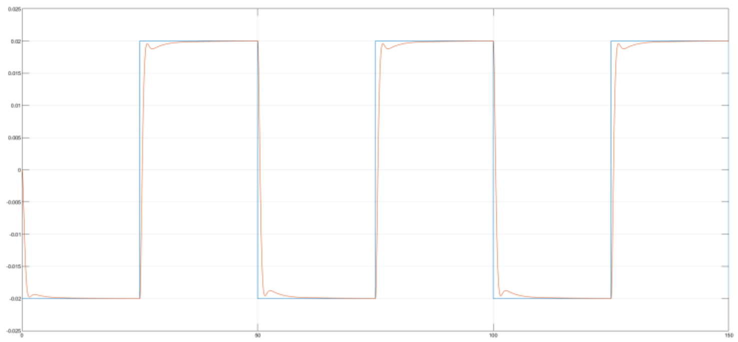

Fig 5: Psi following the reference signal ± 0.02 rad

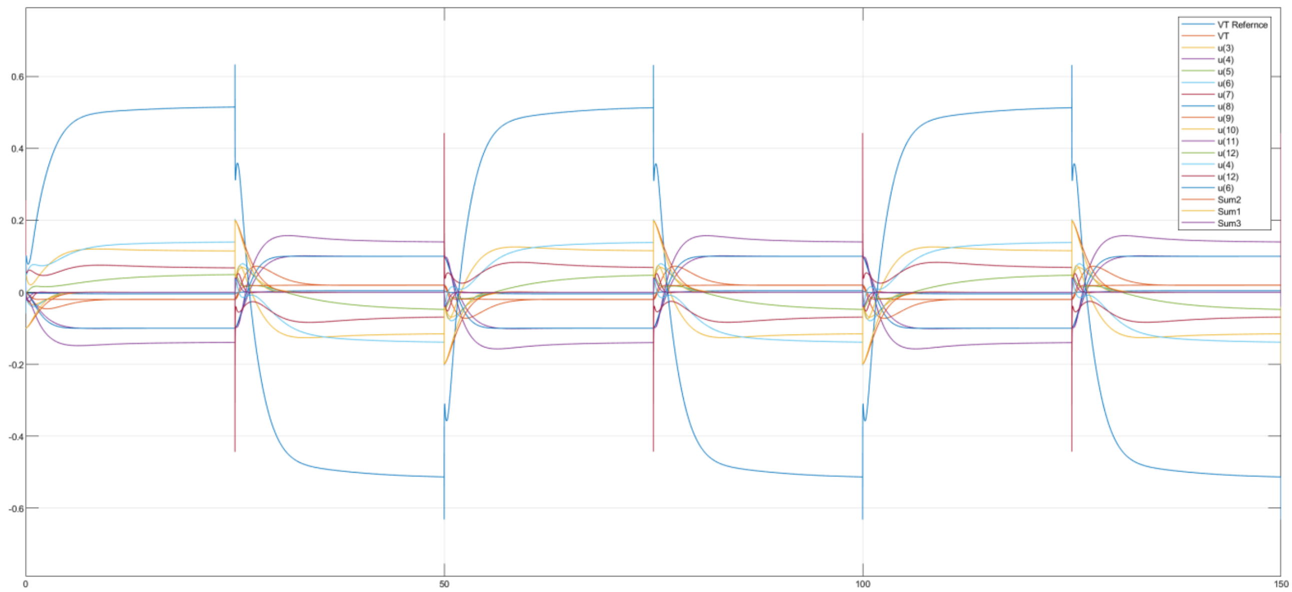

Fig 6: State Output Controls

Discrete Time-System:

After computing the continuous time system, we need to build the Discrete state space model which can be done using c2d command in MATLAB which is shown below with Sampling time Ts=0.1 sec.

sys=ss(As,Bs,Hs,D) % Continuous-time system

Ts=0.1; % Sampling time

sys_d=c2d(sys,Ts,'tustin')

And the outputs of this command are a discrete system which has matrices similar to As, Bs, Hs and Ds, represented as shown below

```
Ak =

    1.0002    0.5012   -1.4888   -0.9786         0         0         0         0         0         0         0         0         0         0         0
   -0.0000    0.9605    0.0961   -0.0004         0         0         0         0         0         0         0         0         0         0         0
   -0.0000    0.2436    0.9783   -0.0000         0         0         0         0         0         0         0         0         0         0         0
   -0.0000    0.0122    0.0989    1.0000         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0    0.9661    0.0080   -0.0967    0.0053         0         0         0         0         0         0         0
         0         0         0         0   -1.8056    0.9071    0.1189   -0.0049         0         0         0         0         0         0         0
         0         0         0         0    0.2227    0.0008    0.9616    0.0006         0         0         0         0         0         0         0
         0         0         0         0   -0.0903    0.0954    0.0059    0.9998         0         0         0         0         0         0         0
         0         0         0         0    0.0111    0.0000    0.0981    0.0000    1.0000         0         0         0         0         0         0
    0.0000   -0.0006   -0.0049   -0.1000         0         0         0         0         0    1.0000         0         0         0         0         0
         0         0         0         0    0.0045   -0.0048   -0.0003   -0.1000         0         0    1.0000         0         0         0         0
         0         0         0         0   -0.0006   -0.0000   -0.0049   -0.0000   -0.1000         0         0    1.0000         0         0         0
    0.0000   -0.0116   -0.0942    0.0000         0         0         0         0         0         0         0         0    0.9048         0         0
         0         0         0         0    0.0860   -0.0908   -0.0057    0.0002         0         0         0         0         0    0.9048         0
         0         0         0         0   -0.0106   -0.0000   -0.0934   -0.0000         0         0         0         0         0         0    0.9048
```

```
Bk =

    0.7373    0.7373   -0.3377   -0.3377         0         0
   -0.0173   -0.0173   -0.0060   -0.0060         0         0
   -0.2903   -0.2903   -0.0112   -0.0112         0         0
   -0.0145   -0.0145   -0.0006   -0.0006         0         0
    0.0016   -0.0016   -0.0028    0.0028   -0.0064    0.0132
   -0.6527    0.6527   -0.8326    0.8326    0.0478    0.3605
   -0.0740    0.0740   -0.0135    0.0135    0.1482   -0.1925
   -0.0326    0.0326   -0.0416    0.0416    0.0024    0.0180
   -0.0037    0.0037   -0.0007    0.0007    0.0074   -0.0096
    0.0007    0.0007    0.0000    0.0000         0         0
    0.0016   -0.0016    0.0021   -0.0021   -0.0001   -0.0009
    0.0002   -0.0002    0.0000   -0.0000   -0.0004    0.0005
    0.0138    0.0138    0.0005    0.0005         0         0
    0.0311   -0.0311    0.0396   -0.0396   -0.0023   -0.0172
    0.0035   -0.0035    0.0006   -0.0006   -0.0071    0.0092
```

```
Hk =

   -0.0000    0.0061    0.0495    1.0000         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0   -0.0451    0.0477    0.0030    0.9999         0         0         0         0         0         0         0
         0         0         0         0    0.0056    0.0000    0.0490    0.0000    1.0000         0         0         0         0         0         0
```

```
Dk =

   -0.0073   -0.0073   -0.0003   -0.0003         0         0
   -0.0163    0.0163   -0.0208    0.0208    0.0012    0.0090
   -0.0019    0.0019   -0.0003    0.0003    0.0037   -0.0048
```

The references are chosen are $\theta = \pm\, 0.1$ rad, $\phi = \pm\, 0.1$ rad and $\psi = \pm\, 0.02$ rad. I have chosen the I_X, I_E, I_E^X, Q and G values same as that of continuous time system. The expanded system dynamics are as follows

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{x}_e(t) \\ \dot{e}(t) \end{bmatrix} = \mathbf{A}\mathbf{x} + A_r r + \mathbf{B}u = \begin{bmatrix} A & 0 & 0 \\ -I_X H & 0 & 0 \\ -HA & I_E^X & -I_E \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ I_X \\ 0 \end{bmatrix} \begin{bmatrix} r \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ -HB \end{bmatrix} u$$

where $\mathbf{A} \in \mathbb{R}^{(n+2b)\times(n+2b)}$, $A_r \in \mathbb{R}^{(n+2b)\times 2b}$ and $\mathbf{B} \in \mathbb{R}^{(n+2b)\times m}$ are the matrices.

In order to design a control law for discrete time system, we can use dlqr command in MATLAB

[Kfeedback,Kn,Eigenvalues]=dlqr(Ak,Bk,Q,G);

Which in turn returns values similar to lqr, which was used for continuous time signal and the outpiuts of this command are as follows

```
Kfeedback =

    0.1394   -0.5402   -1.6310   -2.5556    1.1589   -0.2534   -0.9416   -0.4707   -2.6066    1.2278    0.2763    5.8042    0.1323    0.0293    0.0169
    0.1394   -0.5402   -1.6310   -2.5556   -1.1589    0.2534    0.9416    0.4707    2.6066    1.2278   -0.2763   -5.8042    0.1323   -0.0293   -0.0169
   -1.1344   -1.9196   -1.3189   -4.0487    0.6350   -0.3490    0.0581   -0.5661    0.3591    2.6336    0.3217   -0.6353    0.2824    0.0390   -0.0018
   -1.1344   -1.9196   -1.3189   -4.0487   -0.6350    0.3490   -0.0581    0.5661   -0.3591    2.6336   -0.3217    0.6353    0.2824   -0.0390    0.0018
   -0.0000    0.0000   -0.0000   -0.0000    4.7520   -0.0232    2.3251   -0.3430   13.0728    0.0000    0.2443  -15.9109   -0.0000    0.0085   -0.0431
   -0.0000   -0.0000    0.0000    0.0000    1.5712    0.2139   -3.1130    0.1762   -9.4088   -0.0000   -0.0750   20.0267    0.0000   -0.0213    0.0581
```

```
Kn =

    1.0325   -0.0500    0.0364    0.2073    0.0000   -0.0000    0.0000   -0.0000    0.0000   -0.1047   -0.0000   -0.0000   -0.0093    0.0000    0.0000
   -0.0500    7.5807   -0.0801   -6.2119    0.0000   -0.0000    0.0000   -0.0000    0.0000    2.0862    0.0000   -0.0000    0.0993   -0.0000   -0.0000
    0.0364   -0.0801    1.1630    1.4629   -0.0000   -0.0000   -0.0000    0.0000   -0.0000   -0.7281   -0.0000    0.0000   -0.0771    0.0000   -0.0000
    0.2073   -6.2119    1.4629   33.7078   -0.0000   -0.0000   -0.0000    0.0000   -0.0000  -15.4874   -0.0000    0.0000    3.3927    0.0000   -0.0000
    0.0000    0.0000   -0.0000   -0.0000   27.4681    0.0063    0.1827   -1.8455   26.8352    0.0000    1.3410   -5.7012    0.0000    0.0330   -0.0033
   -0.0000   -0.0000   -0.0000   -0.0000    0.0063    1.0492   -0.0032    0.8639   -0.0012    0.0000   -0.5038    0.0024    0.0000   -0.0641    0.0000
    0.0000    0.0000   -0.0000   -0.0000    0.1827   -0.0032    1.3033   -0.0184    3.4868    0.0000    0.0130   -6.2679    0.0000    0.0007   -0.0186
   -0.0000   -0.0000    0.0000    0.0000   -1.8455    0.8639   -0.0184   22.0459   -1.9146   -0.0000  -12.2845    0.0322    0.0000    3.4530   -0.0005
    0.0000    0.0000   -0.0000   -0.0000   26.8352   -0.0012    3.4868   -1.9146   94.7874    0.0000    1.4001 -137.9880    0.0000    0.0383    4.5756
   -0.1047    2.0862   -0.7281  -15.4874    0.0000    0.0000    0.0000   -0.0000    0.0000   20.6948    0.0000   -0.0000   -1.1301   -0.0000    0.0000
   -0.0000    0.0000   -0.0000   -0.0000    1.3410   -0.5038    0.0130  -12.2845    1.4001    0.0000   17.7170   -0.0860   -0.0000   -1.2713    0.0003
   -0.0000   -0.0000    0.0000    0.0000   -5.7012    0.0024   -6.2679    0.0322 -137.9880   -0.0000   -0.0860  585.1458   -0.0000   -0.0242   -2.9814
   -0.0093    0.0993   -0.0771    3.3927    0.0000    0.0000    0.0000    0.0000    0.0000   -1.1301   -0.0000   -0.0000    5.4486    0.0000    0.0000
    0.0000   -0.0000    0.0000    0.0000    0.0330   -0.0641    0.0007    3.4530    0.0383   -0.0000   -1.2713   -0.0242    0.0000    5.4317   -0.0000
    0.0000   -0.0000   -0.0000   -0.0000   -0.0033    0.0000   -0.0186   -0.0005    4.5756    0.0000    0.0003   -2.9814    0.0000   -0.0000    5.5081
```

```
Eigenvalues =

   0.9763 + 0.0000i
   0.9444 + 0.0308i
   0.9444 - 0.0308i
   0.9320 + 0.0446i
   0.9320 - 0.0446i
   0.9161 + 0.0000i
   0.9043 + 0.0000i
   0.8678 + 0.0000i
   0.8297 + 0.0000i
   0.7712 + 0.1635i
   0.7712 - 0.1635i
   0.1223 + 0.0000i
   0.1128 + 0.0000i
   0.0124 + 0.0000i
   0.0038 + 0.0000i
```

The initial conditions and the Ark values are chosen as shown below

x0=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]'; % Initial conditions

Ark=[zeros(9,3); IX;zeros(3,3)]*Ts;%Input

```
Ark =

         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
    0.1000         0         0
         0    0.1000         0
         0         0    0.1000
         0         0         0
         0         0         0
         0         0         0
```

For closed loop system

$$x_{n+1} = \left[A_n - B_n\left(G_n + B_n^T K_{n+1} B_n\right)^{-1} B_n^T K_n A_n\right] x_n$$

And the optimal control law is given by

$$u_n = -\left(G_n + B_n^T K_n B_n\right)^{-1} B_n^T K_n A_n x_n$$

These are computed by the following MATLAB commands

```
r=[-0.1*square(pi*f*t,50);-0.1*square(pi*f*t,50);-0.02*square(pi*f*t,50)]; % references
x(:,1)=(Ak-Bk*Kfeedback)*x0+Ark*r(:,1);
for k=1:N
        x(:,k+1)=(Ak-Bk*Kfeedback)*x(:,k)+Ark*r(:,k);
        u(:,k)=-Kfeedback*x(:,k);
end
```

The outputs of the following are determined using the following

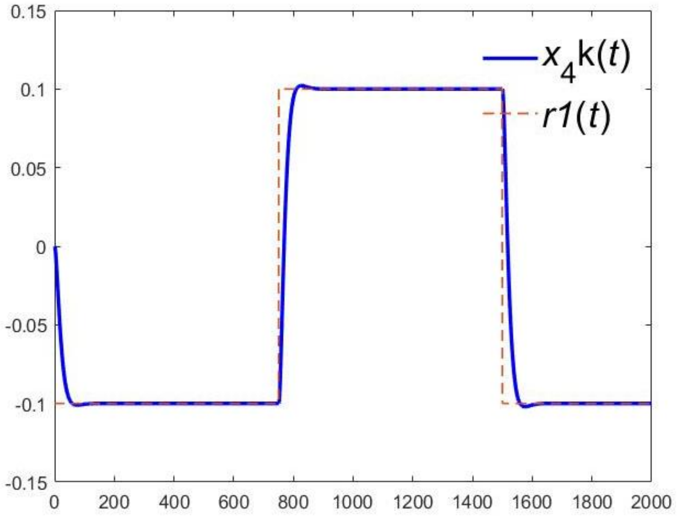

Fig 7: Discrete time Theta following the reference signal ± 0.1 rad

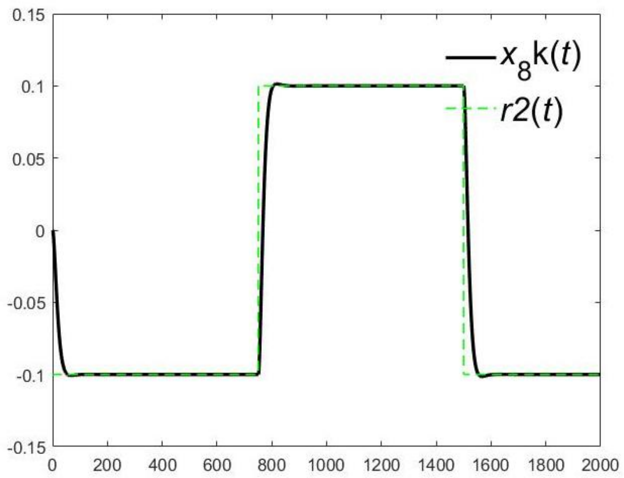

Fig 8: Discrete time phi following the reference signal ± 0.1 rad

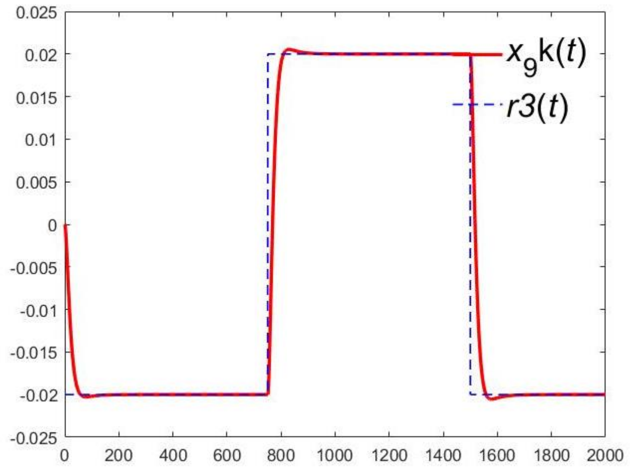

Fig 9: Discrete time psi following the reference signal ± 0.02 rad

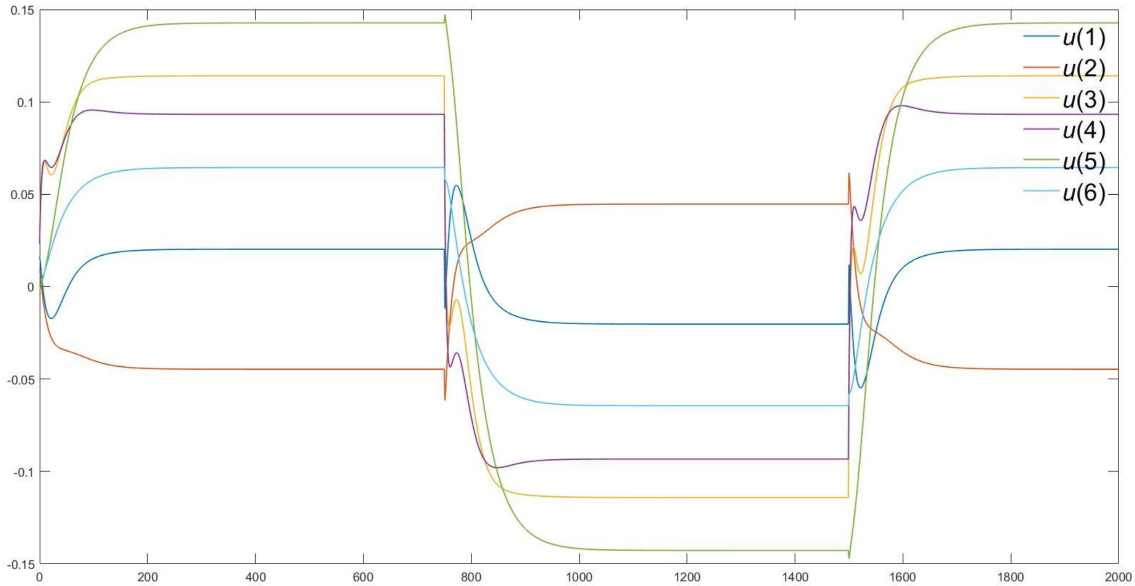

Fig 9: Discrete time controls

The following signals have y axis as Time (sec) and Dynamics of system states, outputs and controls.

MATLAB CODE:

```
clear all;clc;
%Matrices of A and B of the model Paramters
A=[ 0.005142  23.0402  -48.8785  -32.0915  0 0 0 0 0;
   -0.000109 -0.526422  0.997184 -0.004425 0 0 0 0 0;
   -0.000337  2.52708  -0.341902  0.000313 0 0 0 0 0;
    0 0 1 0 0 0 0 0 0;
    0 0 0 0 -0.154099 0.082387 -0.998322 0.05376 0;
    0 0 0 0 -19.2246 -0.893601  0.318845 0 0;
    0 0 0 0  2.29583 -0.000888 -0.278676 0 0;
    0 0 0 0 0 1 0 0 0;
    0 0 0 0 0 0 1 0 0];
A(1,:)=A(1,:)/3.28; A(2,1)=A(2,1)/3.28; A(3,1)= A(3,1)/3.28;
B=[  1.585175  1.585175 -1.049275  -1.049275  0 0;
    -0.033078 -0.033078 -0.0558555 -0.0558555 0 0;
    -2.93107  -2.93107  -0.1058865 -0.1058865 0 0;
     0 0 0 0 0 0  ;
     0.007199 -0.007199  0.0001785 -0.0001785 0.00735  0.021165;
    -6.7916    6.7916   -8.7234     8.7234    0.414519 3.92325;
    -0.752735  0.752735 -0.1342015  0.1342015 1.51008 -1.96651;
     0 0 0 0 0 0;
     0 0 0 0 0 0];
 B(1,:)=B(1,:)*3.28;
%H matrix: Output of Euler angles
H=[ 0 0 0 1 0 0 0 0 0;
    0 0 0 0 0 0 0 1 0;
    0 0 0 0 0 0 0 0 1];
D=0;
%Extended Model with tracking error dynamics
IX=eye(3,3);
IE=eye(3,3);
As=[A zeros(9,6); -IX*H zeros(3,6); -H*A zeros(3,3) -IE];
Bs=[B; zeros(3,6); -H*B];
Hs=[ 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0;
     0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;
     0 0 0 0 0 0 0 0 1 0 0 0 0 0 0];
Q=1*eye(15,15);
Q(6,6)=1;Q(8,8)=0.5;
Q(10,10)=1;
Q(12,12)=100; Q(14,14)=1;
G=0.01*eye(6,6); %minimum value
[K,S,Eig]=lqr(As,Bs,Q,G)
KF=inv(G)*Bs'*S
%%
sys=ss(As,Bs,Hs,D) % Continuous-time system
Ts=0.1; % Sampling time
sys_d=c2d(sys,Ts,'tustin') % Discrete-time system
G_z=tf(sys_d) % Transfer function G(z)
poles=zpk(G_z) % Zeros, poles and gain
% Discrete-time system: State-space model
Ak=sys_d.a, Bk=sys_d.b, Hk=sys_d.c, Dk=sys_d.d;
[Kfeedback,Kn,Eigenvalues]=dlqr(Ak,Bk,Q,G)
%%
x0=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]'; % Initial conditions
Ark=[zeros(9,3);IX;zeros(3,3)]*Ts;%Input
```

```
N=2000;
t=1:N;
f=1/750;
r=[-0.1*square(pi*f*t,50);-0.1*square(pi*f*t,50);-
0.02*square(pi*f*t,50)];

x(:,1)=(Ak-Bk*Kfeedback)*x0+Ark*r(:,1);
for k=1:N
x(:,k+1)=(Ak-Bk*Kfeedback)*x(:,k)+Ark*r(:,k);
u(:,k)=-Kfeedback*x(:,k);
end

k=0:N;
figure(1)
plot(k,x(4,:),'b',k,x(8,:),'k',k,x(9,:),'r','LineWidth',2);hold on
plot(r(1,:),'--','LineWidth',1);
plot(r(2,:),'g--','LineWidth',1);
plot(r(3,:),'b--','LineWidth',1);
hold off
legend('{\itx}_4k({\itt})','{\itx}_8k({\itt})','{\itx}_9k({\itt})','
{\itr1}({\itt})','{\itr2}({\itt})','{\itr3}({\itt})','FontSize',20);
legend boxoff;
%%
figure(2)
plot(k,x(4,:),'b','LineWidth',2);hold on
plot(r(1,:),'--','LineWidth',1); hold off
legend('{\itx}_4k({\itt})','{\itr1}({\itt})','FontSize',20); legend
boxoff;
figure(3)
plot(k,x(8,:),'k','LineWidth',2);hold on
plot(r(2,:),'g--','LineWidth',1); hold off
legend('{\itx}_8k({\itt})','{\itr2}({\itt})','FontSize',20); legend
boxoff;
figure(4)
plot(k,x(9,:),'r','LineWidth',2);hold on
plot(r(3,:),'b--','LineWidth',1);hold off
legend('{\itx}_9k({\itt})','{\itr3}({\itt})','FontSize',20); legend
boxoff;
%%
figure(5)
l=0:N-1;
plot(l,u,'LineWidth',1);
legend('{\itu}(1)','{\itu}(2)','{\itu}(3)','{\itu}(4)','{\itu}(5)','
{\itu}(6)','FontSize',20); legend boxoff;
```

References

[1] S. H. ˙Zak, Systems and Control, Oxford University Press, NY, 2003.