ELL-801 : Non-linear Control

Assignment - 1

Ashwath Ram . S

2023ME28467

Q1) System:

$$\left.\begin{aligned}\dot{x}_1 &= -x_1 + x_2 \\ \dot{x}_2 &= x_1 - x_2 - x_1 x_3 + u \\ \dot{x}_3 &= x_1 + x_1 x_2 - 2x_3 \\ y &= x_1\end{aligned}\right\} \rightarrow \text{at origin } (\overset{x}{0}, \overset{u}{0}) \quad \dot{x}(x,u) = 0$$

Now since the controller should use output feedback and not the state feedback, we must design an observer to estimate the states from the output to pass it as feedback in the input. Since C is not invertible, thus estimation is the only available means.

Now before building the estimator, we must linearize the system about origin to make controller which can stabilize it locally. Global stabilization can be seen or investigated after local stabilization.

Now $f = \begin{bmatrix} -x_1 + x_2 \\ x_1 - x_2 - x_1 x_3 + u \\ x_1 + x_1 x_2 - 2x_3 \end{bmatrix}$

$$A = \left. \begin{bmatrix} -1 & 1 & 0 \\ 1 - x_3 & -1 & -x_1 \\ 1 + x_2 & x_1 & -2 \end{bmatrix} \right|_{x=0,\, u=0} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -2 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} ; \quad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Thus our linearized system (at origin) becomes:

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

Now building the estimator for it

$$\dot{\hat{x}} = A\hat{x} + Bu$$

Estimated output $\hat{y} = C\hat{x}$

$e = x - \hat{x} \Rightarrow \dot{e} = \dot{x} - \dot{\hat{x}} = Ae$

Using output feedback on estimated state dynamics to make $e \to 0$

$$\dot{\hat{x}} = A\hat{x} + Bu - L(y - \hat{y})$$
$$= A\hat{x} + Bu - LCe$$

$$\dot{e} = \dot{x} - \dot{\hat{x}} = Ax + Bu - A\hat{x} - Bu + LCe$$
$$= Ae + LCe = (A + LC)e$$

Thus choosing L such that $(A + LC)$ is Hurwitz would be enough to make $e \longrightarrow 0$.

Now for the controller part :

$$\dot{\hat{x}} = A\hat{x} + Bu - LCe$$

$$u = -K\hat{x}$$

$$\Rightarrow \dot{\hat{x}} = (A - BK)\hat{x} - LCe$$

$$\dot{e} = \dot{x} - \dot{\hat{x}} \Rightarrow \dot{x} = \dot{e} + \dot{\hat{x}}$$

$$= \dot{e} + (A - BK)\hat{x} - LCe$$

$$= (A+LC)e - LCe + (A - BK)\hat{x}$$

$$\Rightarrow \dot{x} = Ae + (A - BK)\hat{x}$$

$$= Ae + (A - BK)(x - e)$$

$$= Ae + Ax - Ae - BKx + BKe$$

$$\dot{x} = (A - BK)x + BKe$$

Now if we make $(A - BK)$ hurwitz, then x will be stabilized at origin as well.

Our design Criteria : (i) K such that $(A-BK)$ is hurwitz.

(II) L such that $(A+LC)$ is hurwitz

(III) (K, L) such that $e \longrightarrow 0$ faster than $x \longrightarrow x(0) = 0$

$$L = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} \Rightarrow |(A+LC) - \lambda I| = 0 \;;\; LC = \begin{bmatrix} l_1 & 0 & 0 \\ l_2 & 0 & 0 \\ l_3 & 0 & 0 \end{bmatrix}$$

$$= \begin{vmatrix} l_1 - 1 - \lambda & 1 & 0 \\ l_2 + 1 & -1 + \lambda & 0 \\ 1 + l_3 & 0 & -2 + \lambda \end{vmatrix} = 0$$

$$\Rightarrow (l_1 - 1 - \lambda)\left[(\lambda - 1)(\lambda - 2)\right] - 1(l_2 + 1)(\lambda - 2) = 0$$

$$\Rightarrow (l_1 - 1 - \lambda)\left[\lambda^2 - 3\lambda + 2\right] - l_2\lambda + 2l_2 - \lambda + 2 = 0$$

if we choose $L = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$, we get eigen values as $[-2, -1, -2] \rightarrow$ verified using MATLAB. Similarly K was found for $|A - BK|$ using Ackerman's Pole placement. //

Q2) $\tau \ddot{\psi} + \dot{\psi} = ku \longrightarrow (1)$

If we take the states of system to be

$x_1 = \psi - \pi/2 \longrightarrow$ Since this is the desired eq point

$x_2 = \dot{\psi}$

$$\left.\begin{aligned} \dot{x}_1 &= \dot{\psi} = x_2 \\ \dot{x}_2 &= \ddot{\psi} = \frac{1}{\tau}(-x_2 + ku) \end{aligned}\right\} \longrightarrow \text{System is already linear}$$

Since it's a state feedback system, considering output to be the state vector

$$y = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \dot{x} = f(x, u) = Ax + Bu$$

where $A = \begin{bmatrix} 0 & 1 \\ 0 & -1/\tau \end{bmatrix}$; $B = \begin{bmatrix} 0 \\ k/\tau \end{bmatrix}$

Now the desired equilibrium state is : $x_{ss} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ ($\psi = \pi/2$, $\dot{\psi} = 0$)

Using these states in dynamic eqn (1) to find u_{ss} :

We get $u_{ss} = 0$

Now for the first case,

Considering a state feed back controller with no integral control : we take $u = -Kx$

$$\therefore \quad \dot{x} = (A - BK)x$$

Now on applying integral control by considering

$$r = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Augmented state space :

$$\dot{x} = Ax + Bu$$

$$\dot{\sigma} = y - r = Cx - r = e$$

$$u = -K^T \begin{bmatrix} x \\ \sigma \\ e \end{bmatrix} = -K_1 x - K_2 \sigma - K_3 e$$

Augmuted State matrices:

$$\mathcal{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} ; \mathcal{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} ; K = [K_1 \ K_2 \ K_3]$$

Now since our control variable is just ψ and stabilizing ψ to ψ_r will automatically make $\dot{\psi} = 0$, we can reduce r from $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ to just 0, thus reducing our state space.

since $x_1 = \psi - \pi/2 \rightarrow \psi_r$

After doing this change:

$$\dot{x} = Ax + Bu$$
$$\dot{\sigma} = y - r$$

where $r = 0$, $y = x_1$, thus $y = \underbrace{[1 \quad 0]}_{C} x$

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1/\tau & 0 \\ 1 & 0 & 0 \end{bmatrix}, \mathcal{B} = \begin{bmatrix} 0 \\ k/\tau \\ 0 \end{bmatrix}, C = [1 \quad 0]$$

$$\boxed{\text{Matrix } \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1/\tau & k/\tau \\ 1 & 0 & 0 \end{bmatrix} = T}$$

$\text{rank}(T) = 3$ if $k \neq 0, \tau \neq 0$

$= n + p$ as $n = 2, p = 1$ //

Thus $(\mathcal{A}, \mathcal{B})$ is controllable. //

Now we design K such that $(\mathcal{A} - \mathcal{B}K)$ is hurwitz.

Considering $K_3 = 0$ for now

$$K = [\underbrace{K_1}_{(1\times 2)\text{ vector}} \quad \overbrace{K_2}^{(1\times 1)\text{ vector}}]$$ where $K_2 \neq 0$

Using nominal values of I, k : $I = 1$, $k = 1$

we have $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

$$K = [K_1 \; K_2 \; K_3]$$

$\lambda(A - BK) < 0 \longrightarrow$ K can be modelled using pole placement.

$\downarrow$

Done using MATLAB.

Q3) $\dot{x}_1 = -x_1 + x_2 - x_3$

$\dot{x}_2 = -x_1 x_3 - x_2 + u$

$\dot{x}_3 = -x_1 + u$

$y = x_3$

(a) $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \Rightarrow \dot{x} = \underbrace{\begin{bmatrix} -x_1 + x_2 - x_3 \\ -x_1 x_3 - x_2 \\ -x_1 \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}}_{g(x)} u$

$y = x_3 = h(x)$

Now $n = 3$

- $L_g h(x) = \frac{\partial h}{\partial x} \cdot g(x) = \begin{bmatrix} \frac{\partial h}{\partial x_1} & \frac{\partial h}{\partial x_2} & \frac{\partial h}{\partial x_3} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}$

$= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 1$

This can also be shown using

$\dot{y} = \frac{\partial y}{\partial x_1} \dot{x}_1 + \frac{\partial y}{\partial x_2} \dot{x}_2 + \frac{\partial y}{\partial x_3} \dot{x}_3$

$= \dot{x}_3 = -x_1 + u \rightarrow L_g h = 1 \neq 0$

Therefore relative degree of system is

$\rho = 1$ in $\mathbb{R}^3$. //

This system is input - output linearizable on $\mathbb{R}^3$. //

(b) $\rho = 1$ in $\mathbb{R}^2$

$n = 3$

Thus $T(x) = \begin{pmatrix} h(x) \\ \phi_1(x) \\ \phi_2(x) \end{pmatrix}$ where $L_g \phi_i = 0$ so that $T(x)$ is a diffeomorphism

if we take $\phi_1(x) = x_1$, $\phi_2 = x_2 - x_3$

then, $L_g \phi_1 = \left[\frac{\partial \phi_1}{\partial x_1} \quad \frac{\partial \phi_2}{\partial x_2} \quad \frac{\partial \phi_3}{\partial x_3} \right] \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$

$= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 0$

Similarly $L_g \phi_2 = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 0$

Thus T is a diffeomorphism in $\mathbb{R}^3$ //

$$\therefore \begin{bmatrix} z \\ \eta \end{bmatrix} = \begin{bmatrix} x_3 \\ x_1 \\ x_2 - x_3 \end{bmatrix}$$

where $z = x_3$, $\eta = \begin{bmatrix} x_1 \\ x_2 - x_3 \end{bmatrix}$

$$\dot{\eta} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 - \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -x_1 + x_2 - x_3 \\ -x_1 x_3 - x_2 + \cancel{u} + x_1 - \cancel{u} \end{bmatrix}$$

$$= \begin{bmatrix} -x_1 + x_2 - x_3 \\ x_1(1 - x_3) - x_2 \end{bmatrix}$$

$$\dot{z} = \dot{x}_3 = -x_1 + u$$

$$\Rightarrow (1) \quad \left.\begin{aligned} \dot{\eta}_1 &= -\eta_1 + \eta_2 \\ \dot{\eta}_2 &= \eta_1(1 - z) - (z + \eta_2) \end{aligned}\right] \rightarrow f_0(\eta, z)$$

$$\dot{z} = -\eta_1 + u \quad //$$

$$y = z$$

(c) Zero dynamics is the dynamics of the states of the system which cannot be controlled by the input. So when controlled by the input. So when $z = 0$ which is the controlled state, then $\dot{\eta} = f_0(\eta, 0)$ is called zero dynamics of the system.

$$f_0(\eta, 0) = \begin{bmatrix} -\eta_1 + \eta_2 \\ \eta_1 - \eta_2 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}}\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$$

Eigen values are
$\lambda_1 = 0$
$\lambda_2 = -2$

Since $\lambda \nless 0$ as $\lambda_1 = 0$, it is not asymptotically stable. System is not minimum phase. //

MATLAB Code section

Qn1)

```
%Observer design
A = [-1 1 0;1 -1 0;1 0 -2];
B = [0;1;0];
C = [1 0 0];
L = [-1;-1;-1]
```

```
L = 3×1

    -1
    -1
    -1
```

```
eig(A+(L*C))    %Eigen values
```

```
ans = 3×1

    -2
    -1
    -2
```

```

%Controller design
A = [-1 1 0;1 -1 0;1 0 -2];
B = [0;1;0];
C = [1 0 0];
poles = [-1 -1 -1]
```

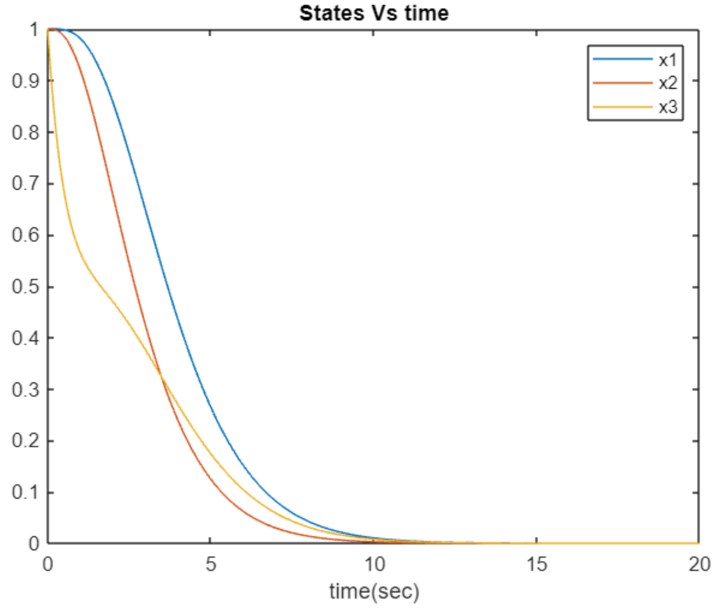
```
poles = 1×3

    -1    -1    -1
```

```
%Finding gain matrix using Ackerman pole placement
K = Ackermann(A,B,poles)
```

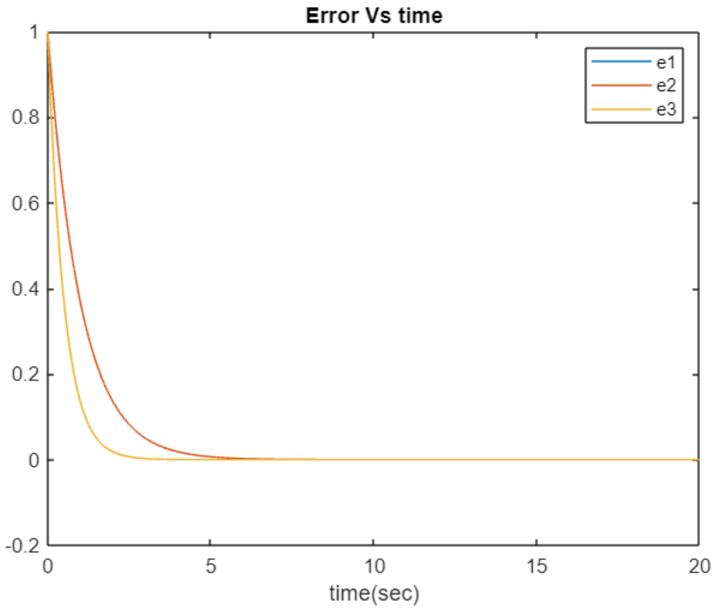
```
K = 1×3

     2    -1    -1
```

```

%Simulating the results

tstart = 0;
tend = 20;
dt = 0.01;
n = 1000;

tspan = transpose(linspace(tstart,tend,n));
xinit = [1 1 1 1 1 1]; %initial values of states and errors

[t,x] = ode45(@integratingfunc_SpringMass_system, tspan, xinit);
```

```
plot(t,x(:,1),t,x(:,2),t,x(:,3));
title("States Vs time")
legend(["x1","x2","x3"])
xlabel("time(sec)")
```




```
plot(t,x(:,4),t,x(:,5),t,x(:,6));
title("Error Vs time")
legend(["e1","e2","e3"])
xlabel("time(sec)")
```




```
function dxdt = integratingfunc_SpringMass_system(t,x)
%INTEGRATINGFUNC_SPRINGMASS_SYSTEM Summary of this function goes here
%   Detailed explanation goes here
A = [-1 1 0;1 -1 0;1 0 -2];
```

```
B = [0;1;0];
C = [1 0 0];
K = [2 -1 -1];
L = [-1;-1;-1];

dxdt = zeros(6,1);
%T = Tss - (k1*(pos-del)) - (k2*(v));
dxdt(1:3) = ((A-(B*K))*x(1:3)) + (B*K*x(4:6));
dxdt(4:6) = (A+(L*C))*x(4:6);
end


%Ackermann function for pole placement
function K = Ackermann(Ac,Bc,poles)

n = size(Ac,1);
m = size(Bc,2);
In = eye(n);
del_A = eye(n);
%Forming characteristic equation
for i=1:n
    del_A = del_A*(Ac-(poles(i)*In));
end
const = zeros(1,n);
const(1,n) = 1;

%Forming controllability matrix
iter = Ac*Bc;
    con = [Bc iter];
    for i = 1:n-2
        iter = Ac*iter;
        con = [con iter];
    end
K = const*(con\del_A);      % For SISO K = [0 0 ...1]*inv(con)*A_desired

end
```

Qn2)

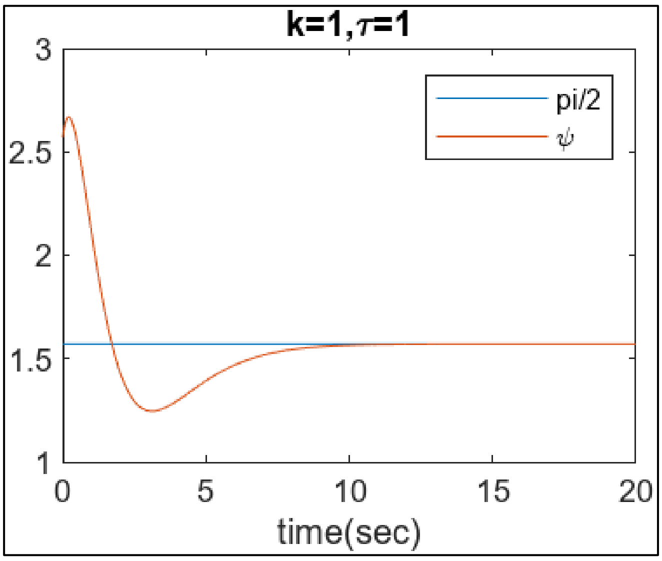
```
%Controller design
A = [0 1 0;0 -1 1;1 0 0];
B = [0;1;0];
poles = [-1 -1 -1]
```

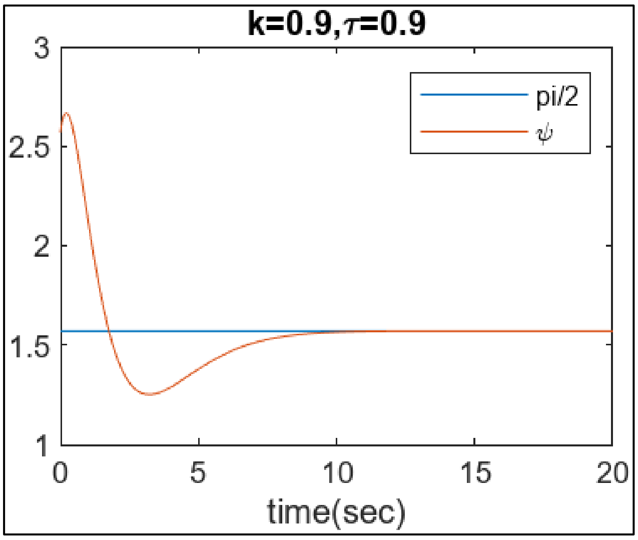
```
poles = 1×3

    -1     -1     -1
```

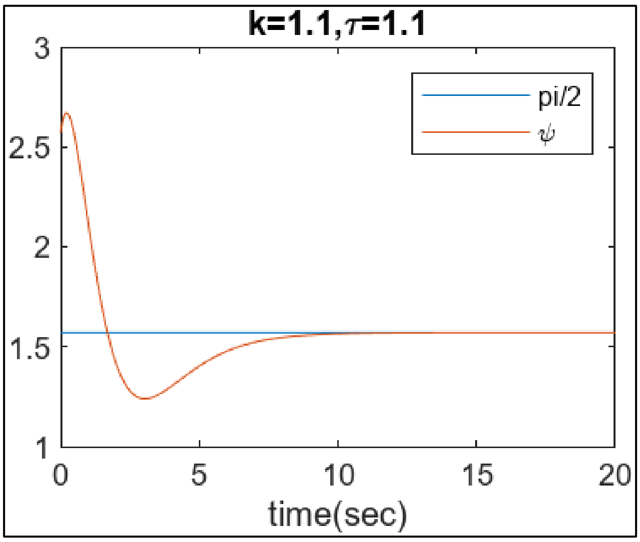
```
%Finding gain matrix using Ackerman pole placement
K = Ackermann(A,B,poles)
```

```
K = 1×3

     3     2     2
```

```
tstart = 0;
tend = 20;
dt = 0.01;
n = 1000;

tspan = transpose(linspace(tstart,tend,n));
xinit = [1 1 0];  %initial values of [x1 x2 sigma]

[t,x] = ode45(@integratingfunc_SpringMass_system, tspan, xinit);
plot(t,zeros(size(tspan))+(pi/2),t,x(:,1)+(pi/2))
title("k=1,\tau=1")
legend(["pi/2","\psi"])
xlabel("time(sec)")
```








```
function dxdt = integratingfunc_SpringMass_system(t,x)
%INTEGRATINGFUNC_SPRINGMASS_SYSTEM Summary of this function goes here
%   Detailed explanation goes here
k_act = 1;
tau_act = 1;
A = [0 1 0;0 (-1/tau_act) (k_act/tau_act);1 0 0];
B = [0;k_act/tau_act;0];
K = [3 2 2];

dxdt = (A-(B*K))*x;
end

%Ackermann function for pole placement
function K = Ackermann(Ac,Bc,poles)

n = size(Ac,1);
m = size(Bc,2);
In = eye(n);
del_A = eye(n);
%Forming characteristic equation
for i=1:n
    del_A = del_A*(Ac-(poles(i)*In));
end
const = zeros(1,n);
const(1,n) = 1;

%Forming controllability matrix
iter = Ac*Bc;
    con = [Bc iter];
    for i = 1:n-2
        iter = Ac*iter;
```

```
        con = [con iter];
    end
K = const*(con\del_A);      % For SISO K = [0 0 ...1]*inv(con)*A_desired

end
```