Large Scale Homework #1

Problem 1_Alef

Centralized design:

First we design centralize system using Centeralized_form1 m.file.

We define system:

```
s = tf('s');
G11 = (0.9*s+15.4)/(s^2+9.163*s+15.47);
G12 = -0.01/(s+6.931);
G21 = 0.025/(s+2.232);
G22 = (0.7549*s+13.9)/((s+2)*(s+6));
G = [G11 G12;G21 G22];

[A,B,C,D] = ssdata(G)
```

The state space of system is:

$$A = \begin{bmatrix} -9.1630 & -3.8675 & 0 & 0 & 0 & 0 \\ 4.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2.2320 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6.9310 & 0 & 0 \\ 0 & 0 & 0 & 0 & -8.0000 & -3.0000 \\ 0 & 0 & 0 & 0 & 4.0000 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 2.0000 & 0 \\ 0 & 0 \\ 0.1250 & 0 \\ 0 & 0.1250 \\ 0 & 2.0000 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.4500 & 1.9250 & 0 & -0.0800 & 0 & 0 \\ 0 & 0 & 0.2000 & 0 & 0.3775 & 1.7375 \end{bmatrix}$$

So there are six states and the size of Q matrix is 6*6:

We write LQR below the function

```
Q = eye(6);
R = eye(2);

K = lqr(A,B,Q,R);
u = -K*x;
xp = A*x+B*u;
```

The LQR gains are as follows:

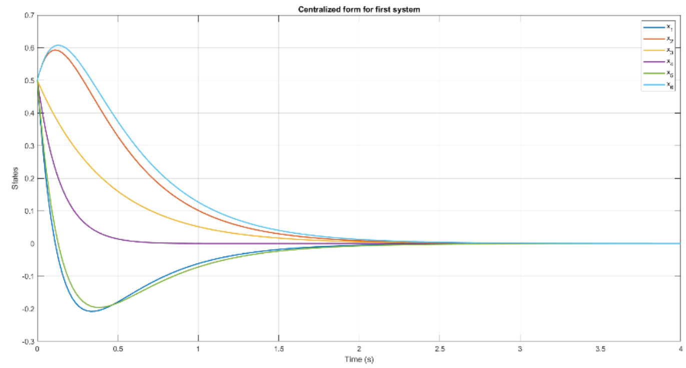
K =

0.2104	0.2431	0.0261	-0.0000	0.0000	0.0000
-0.0000	0.0000	-0.0000	0.0086	0.2674	0.3027

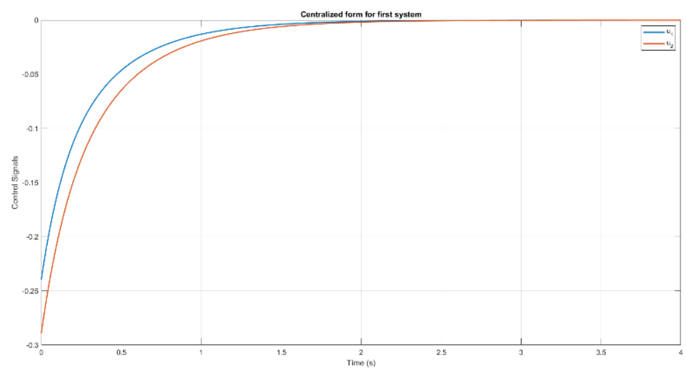
After that we solve state space we give initial condition by using ODE45:

```
tspan = [0 8];
x0 = [0.5;0.5;0.5;0.5;0.5;0.5];
[t,x] = ode45(@Cent1,tspan,x0);
% Solving with ODE45
x = x';
% For Dimensions
u = -K*x;
% State Feedback
```

The states of system are:



As can be seen all states divergent to the zero in 2.5 seconds.



we can see the control signals are tracked well.

For measuring performance, we have to achieve cost function:

$$P = \int_0^\infty x^T Q x dt$$

So, we write for loop for calculate the integral:

```
JJJ = [];
J(1,1)=0;
JJJ(1,:)=0;
tt = 0;
xT = x';
uT = u';


for i = 1:70
     dt=t(i)-tt;
     tt=t(i);
```

```
    S = (xT(i,:)*Q*x(:,i))*dt;
    % S = (xT*Q*x)+(uT*R*u)
    J = S+J;
    JJJ(i+1,:)=J;
End
```

P_cen equals to =

0.4362

Decentralized form

For designing Decentralized LQR we have to convert the system in Jordan form:

```
eig(sys_decent)
[T,Aj] = jordan(A)

Bj = inv(T)*B

Cj = C*T

Q = T'*eye(6)*T
```

The Jordan form of state space is:

$$A = \begin{bmatrix} -2.2320 & 0 & 0 & 0 & 0 & 0 \\ 0 & -6.9310 & 0 & 0 & 0 & 0 \\ 0 & 0 & -6.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & -6.9310 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2.2320 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.1250 & 0 \\ 0 & 0.1250 \\ 0 & -2.0000 \\ 0 & 2.0000 \\ -1.7025 & 0 \\ 1.7025 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & -0.0800 & 0 & 0 & 1.1453 & 1.6739 \\ 0.2000 & 0 & 1.1713 & 1.5488 & 0 & 0 \end{bmatrix}$$

So, we are able to derive G11 and G22:

$$A11 = \begin{bmatrix} -6.93 & 0 \\ 0 & -2 \end{bmatrix}$$

$$A22 = \begin{bmatrix} -6 & 0 \\ 0 & -2 \end{bmatrix}$$

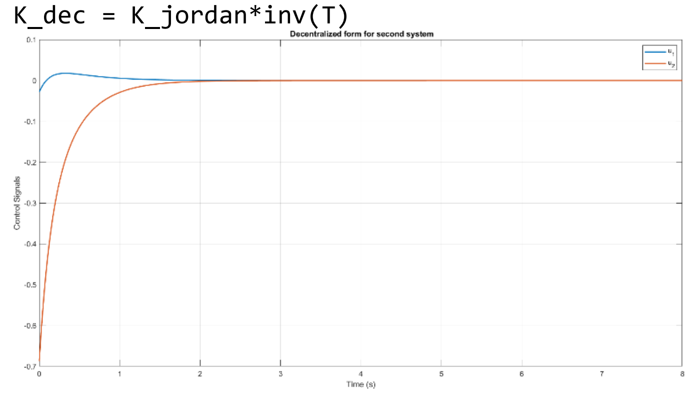
$$B1 = \begin{bmatrix} -1.73 \\ 1.73 \end{bmatrix}$$

$$B1 = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

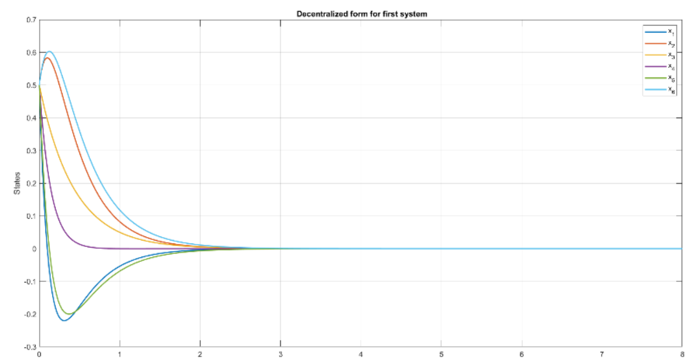
$$Q11 = \begin{bmatrix} 4 & 1.9669 \\ 1.9669 & 1.311 \end{bmatrix}$$

$$Q22 = \begin{bmatrix} 1.25 & 0 \\ 0 & 4.0024 \end{bmatrix}$$

```
K11 = lqr(A11,B11,Q11,R)
K22 = lqr(A22,B22,Q22,R)

K_jordan = [K11 0 0 0 0;...
            0 0 0 0 K22]
K_dec = K_jordan*inv(T)
```



The control signals are tracked well.



The states settled in 2.5 second.

The value of cost function is:

0.6160

Which is a bit higher than centralized form. This is normal since the performance of the decentralized control is lower.

$$Q_d \leq Q_c$$

Problem1_Be

Centralized design

Large Scale Homework #1

$$G = \begin{bmatrix} \dfrac{-0.805}{(0.3s+1)(1.6s+1)} & \dfrac{0.055}{(2.76s+1)(1.25s+1)} \\ \dfrac{0.465}{1.3s+1} & \dfrac{0.055}{3.3s+1} \end{bmatrix}$$

$$J = \int_0^{\infty} (x^T Q x + u^T R u)\,dt \quad , \quad Q \ \& \ R \equiv \ Identity\ matrix$$

First define the transfer function in Matlab, as follows:

First sub-systems:

```
s = tf('s');
g11=(-0.805/((0.3*s+1)*(1.6*s+1)));
g12=(0.055/((2.76*s+1)*(1.25*s+1)));
g21=(0.465/(1.3*s+1));
g22=(0.055/(3.3*s+1));
```

then cause the system is a MIMO system we define the system with a matrix containing the sub-systems:

```
G=[g11 g12;g21 g22]
```

Then we derive the state-space matrices A, B, C, D:

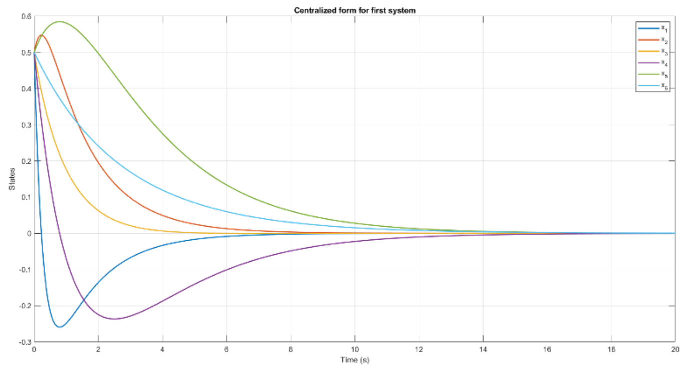
```
sys_decent=ss(G,'minimal')
A=sys_decent.A
B=sys_decent.B
C=sys_decent.C
D=sys_decent.D
```

Don’t forget we want the minimal form, for this question.

Matrices are, as follows:

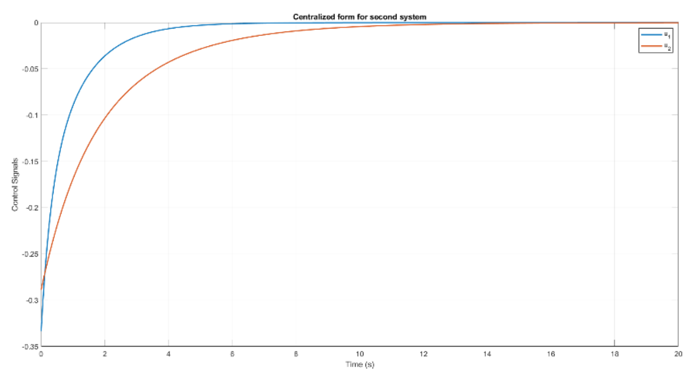
```
A =

   -3.9583   -2.0833         0         0         0
    1.0000         0         0         0         0
         0         0   -0.7692         0         0
         0         0         0   -1.1623   -0.5797
         0         0         0    0.5000         0
         0         0         0         0         0   -0.

B =

    1.0000         0
         0         0
    0.5000         0
         0    0.2500
         0         0
         0    0.1250

C =

         0   -1.6771         0         0    0.1275         0
         0         0    0.7154         0         0    0.1333

D =

     0     0
     0     0
```

This system has 2 inputs, 2 outputs, and 6 states.



The settling of centralized controlled system is 14 seconds.

The control signals are:



And the value of the cost function is:

2.1073

Decentralized design:

First thing first! Before designing an LQR controller first me need to get the Jordan form of the system. Cause the question has asked us to design the controllers for subsystems 11 and 22 and diagonalized, we have the below code:

```
[T,Aj] = jordan(A)
Bj = inv(T)*B
Cj = C*T
Q = T'*eye(6)*T
Q11 = [12.1111 3.0833;3.0833 1.3906]
Q22 = [1]
R = eye(1)
```

Minoo Jahanseir
Mohammad Alikhani

Large Scale Homework #1

```
A11 = [-3.3333 0;0 -0.6250]
B11=[-0.3692;0.3692]
A22=[ -0.3030]
B22=[0.1250]
```

The LQR Design process is quite simple in Matlab. Just use the LQR command:

```
K11 = lqr(A11,B11,Q11,R)
K22 = lqr(A22,B22,Q22,R)
K_jordan = [K11 0 0 0 0; 0 0 0 0 0 
K22]
K_decentralized = K_jordan*inv(T)
tspan = [0 20];
x0 = [0.5;0.5;0.5;0.5;0.5;0.5];
```

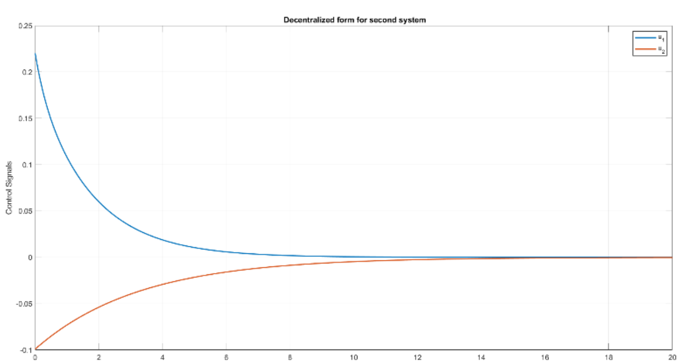
Beware that the initial conditions for this system are 0.5.

```
K11 = lqr(A11,B11,Q11,R)
K22 = lqr(A22,B22,Q22,R)
K_jordan = [K11 0 0 0 0; 0 0 0 0 0 
K22]
K_decentralized = K_jordan*inv(T)
tspan = [0 20];
x0 = [0.5;0.5;0.5;0.5;0.5;0.5];
```

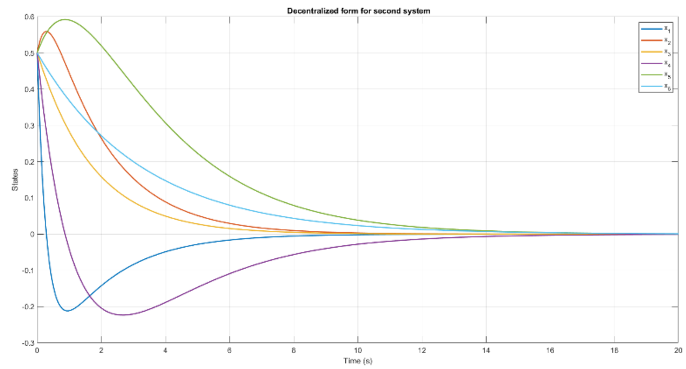
The LQR gains are:

```
K11 =

   -0.3715    0.1152

K22 =

    0.1982
```

The input signals are:



And the states are:



As can be seen the centralized form are quite similar to decentralized form.

Finally, the value of decentralized cost function is:

2.4577

This cost function is higher than pervious cost function the thing is interventions of system are high and in decentralized form we ignore the important part of system.

Results:

Now that we have computed all the costs, we can now write the controllers' performance with respect to their cost values (highest to lowest), which are as follows:

1. Decentralized Controller for interacted system.
2. Centralized Controller for interacted system.
3. Decentralized Controller for weakly coupled system.
4. Centralized Controller for weakly coupled system.

Problem 2:

First, we define system:

```
A=[0 1 0;
    100 -10 -300;
    0 0 -5];
B=[0 0.5 0;
    -0.05 0.5 7
    ;0 0 0];
C=eye(3);
D=zeros(3);
```

And check the controllability:

```
p = ctrb(A,B)
rank(p)
if(rank(p)==length(A))
    disp('system  is controllable');
```

```
else disp('system is not controllable')
end
```

The system is not controllable so we are obliged to check the stabilization of the system. So we calculate the eigen values:

```
sys = ss(A,B,C,D)
[p z]=pzmap(sys)
```

This system has 3 eigen values:

6.1803

-16.1803

-5.0000

One of them is unstable so we have to check whether is controllable or not:

```
A1=-6.1803*eye-A;
P1=[A1 B]
rank(P1)
```

This mode of system is controllable.

After that we design LQR and check if this controller can stabilize the system or cannot:

```
Q = eye(3);
R = eye(3);
K = lqr(A,B,Q,R)
closed_loop_eig=eig(A-B*K)
```

The closed loop eigen values are:

-6.3864

-17.5703

-5.0000

The gain of LQR is:

K =

-0.0824	-0.0066	0.1463
14.0549	0.8905	-23.5059
11.5409	0.9262	-20.4781

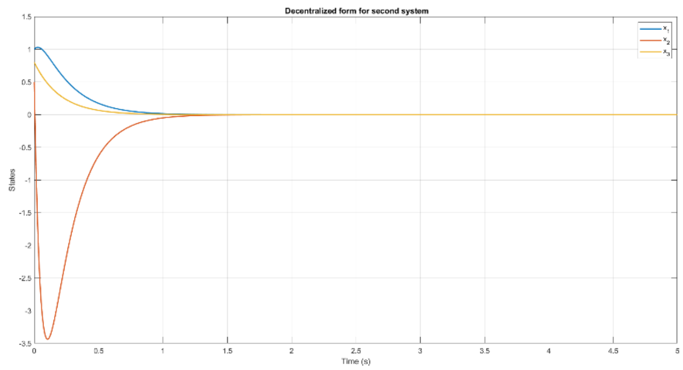
This closed loop system is stable

Then we check the observability:

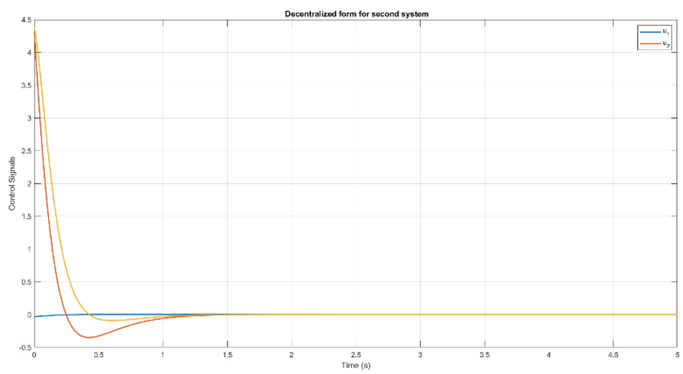
```
P2=obsv(A,C);
if(rank(P2)==length(A))
 disp('the system is observable');
else disp('the system is not observable')
end
```

the system is observable.

After that we plot the state response:



The control signals are:



For last section we want to place poles at the left of y=-3 axis, Therefore we solve the below Riccati:

$$(\mathbf{A}' - d\mathbf{I})\widetilde{\mathbf{P}} + \widetilde{\mathbf{P}}(\mathbf{A} - d\mathbf{I}) - \widetilde{\mathbf{P}}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}'\widetilde{\mathbf{P}} + \mathbf{Q} = \mathbf{0}$$

We use “care” command in MATLAB:

```
A_new=A-3*eye(3);
E=eye(3);
S=zeros(3);
[P_tild,L,K_new] = care(A_new,B,Q,R,S,E)
L=eig(A_new-B*K_new)
```

We set undetermined E to identity matrix and S to zero matrix. The new poles are:

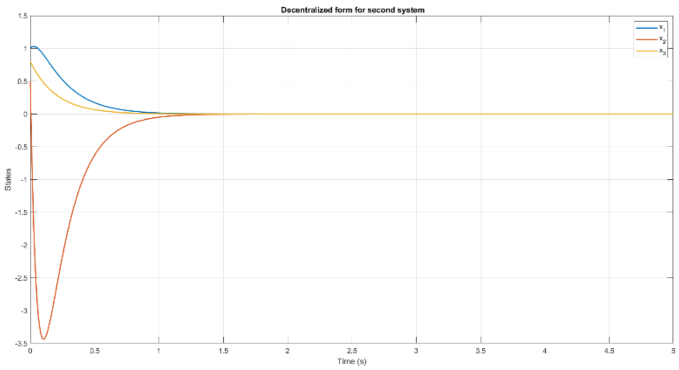
-4.1064

-8.0000

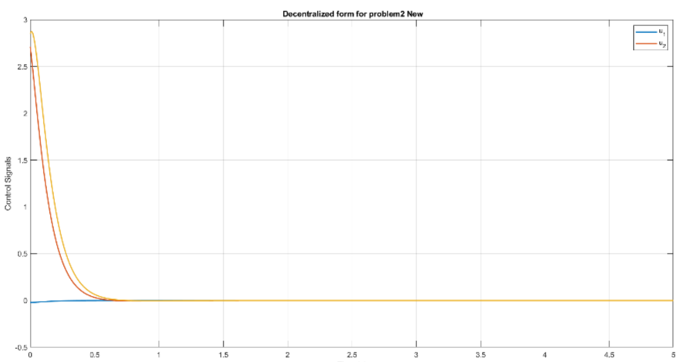
-20.2642

For observing the states please run Exercise2_New.mfile:

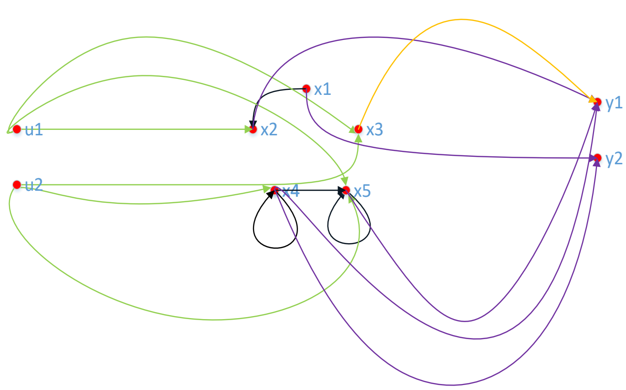
The state response is:



The control signals are:



Problem 3:



This is the corresponding graph of the system.

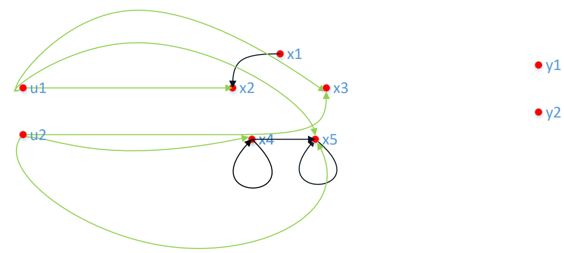
3.1

For different requirements, different conditions must be satisfied.

Input-connectable:

There must be a path from all inputs to all states.

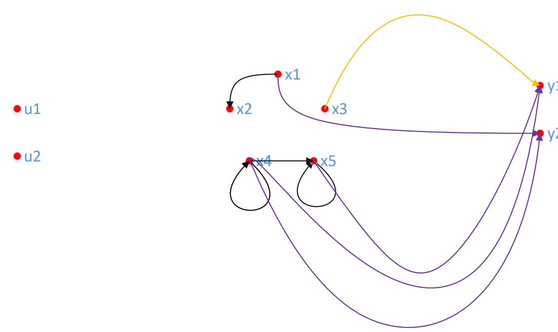
This system **isn't** input-connectable.



Output-connectable:

All states must be connected to the outputs via at least one path.

Grant that a = 1, system **is output-connectable**.



3.2

Structurally controllable:

Two conditions must be satisfied:

Firstly, the system must be Input-Connectable.

Second, $(S_A \;\; S_B)$ must be full rank(S_A and S_B are structural matrices and $(S_A \;\; S_B)$ is a block matrix)

This system, isn't Input-Connectable, first requirement is not met, so **not Structurally controllable:**

Structurally Observable:

Two conditions must be satisfied:

First thing first, system must Output-Connectable.

Second, rank$\begin{pmatrix} s_a \\ s_b \end{pmatrix}$ must be full rank(block matrix).

Cause rank$\begin{pmatrix} s_a \\ s_b \end{pmatrix}$ = 5, system **is Structurally observable**

3.3

Fixed Mode

There are two types of fixed modes:

Type 1: State **must not be** Structurally Observable or Structurally Controllable.

Type 2: The cycle family **must not be** full rank.

X_1 is a **type 1 fixed mode**.

X_3 cause it's output-controllability depends on the value of a, for a = 0 which is not output-controllable, $\boldsymbol{X_3}$ **is type 1 fixed mode.**