SSY281 Model Predictive Control

Assignment 2

Yongzhao Chen

February 15, 2023

1 Question 1 Set-point calculate

1.1 a

For this question, for dimensions of u and $y_s p$ are the same, so this situation is $p = m$, which means the system is totally controllable, and $y_s p$ can be settled.

```
ys=[pi/18;-pi];
temp1=[eye(length(A))-A, -B;C, zeros(2,2)];
[~,n1]=size(temp1);
temp1state=inv(temp1)*[zeros(n1-2,1);ys]
xs1=temp1state(1:4,:)
us1=temp1state(5:end,:)
```

Output information:

```
xs1 = ×41
    0.1745
    0.0722
   -3.1416
    0.0008

us1 = ×21
   -0.0153
   -0.1427
```

1.2 b

Now the outputs is more than manipulated input (p¿m situation), so it cannot be tracked, also cannot settle at y_s.
But it can be minimized. Calculate the $xaug2$ minimize $(C * Xs - Ysp)' * Q * (C * Xs - Ysp)$, then verify it depending on the difference, which is not zero, so it can not be settled as $y_s p$.

```
xaug2=quadprog(H,f,[],[],Aeq,Beq,[],[],[],options);
xs2=xaug2(1:4,:)
us2=xaug2(5:end,:)

[eye(4)-A -B2; C C*B*0]*xaug2 - [zeros(4,1) ; ys]

```

```
Output:

xs2 = ×41
     0.0000
     0.0000
    -3.1416
          0

us2 = -1.2096e-13

ans = ×61
    -0.0000
    -0.0000
    -0.0000
    -0.0000
    -0.1745
     0.0000
```

1.3 c

In this question, this situation changes into $p < m$, then it can be settle and can minimise the uncontrollable outputs.

```
xs3 = ×41
     0.1745
     0.0722
          0
     0.0008

us3 = ×21
    -0.0153
    -0.1427
```

2 Question 2: Control of a ball and wheel system

2.1 a

Using the part below to tell whether the system is detectable or not.

```
% detectable criterion:  equation 13
CriterionMatrix=rref([eye(na)-A,-Bd;C,Cd])
if rank(CriterionMatrix)==na+nd
    sprintf('The system %c is detectable!',subcase)
else
    sprintf('The system %c is not detectable!',subcase)
end

Output:
For case1:
CriterionMatrix = ×65
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
     0     0     0     0     0
ans = 'The system 2 is detectable!'

For case 2:
CriterionMatrix = ×66
     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     1
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     0
ans = 'The system 2 is not detectable!'

CriterionMatrix = ×66
     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
ans = 'The system 3 is detectable!'
```

So the system 1 and 3 are detectable while the system 2 is not detectable.

2.2 b

L is the matrix we want.

```
Q=eye(na+nd);
R=eye(p);
[P,K] = idare(Aaug',Caug',Q,R,[],[])
L=Aaug\K'
```

So for detectable system 1 and 3, their L are listed below:

```
Output:
system1:

L = ×52
    1.0912   -0.4025
    8.7809   -3.1649
    0.4246    0.2332
    0.3672   -0.0835
   -0.3508    0.4763



system3:
L = ×62
    1.2496    0.0099
   10.4751    0.2567
    0.0111    0.2279
    0.2721    1.8818
   -0.4815   -0.0090
   -0.0102    0.5109
```

2.3 c

from page 50, set $Zsp = 0$, we can findout the way to calculate Mss.
The output of my code is here:

```
For system 1:
```

```
Mss = ×51
     0
     0
    -1
     0
     0

For sytem 3:
Mss = ×52
         0   -0.0000
         0    0.0000
         0   -1.0000
         0   -0.0000
         0   -1.0000
```

2.4 d

I make mistakes about M, so the result is not correct, but here is the code that I tried. The logic is *calculate xs and us*, then *calculate du and get u*, then *update states and outputs*, finally *update observer*, and loop.

```
% consider zero as the intial condition, ys=0
Ttf=1000;
ys=zeros(4,tf);
n=4;%dimension of x
N=50;
M=40;
R=0.1;
Q=diag([5 2 0.5 0.1]);
Pf=Q;
X0=[pi/36,0,0,0,0]'%augment u

dk=[zeros(50,1); 0.2*ones(tf-51+1,1)]';
x_state=zeros(4,tf);
%x_state(:,1) = X0;
x_ob=zeros(5,tf+1);
x_ob(:,1)=X0;
x_ob(:,2)=X0;
u_state=zeros(1,tf);
y_out=zeros(2,tf);

for i= 1 :tf
```

```
    % get us
    pk=dk(i);
    xaug=Mss*pk;
    xs=xaug(1:na);
    us=xaug(na+1:end);

    % from the lecturenotes 48, the input for RHC should be x_ob-xs
    dx=x_ob(:,i)-xaug;
    %function [Z,VN,x,u,u0]=URHC(A,B,N,Q,R,Pf,x0,n)
    [Z,VN,x,u,du]=URHC(A,B,N,Q,R,Pf,dx(1:na,:),n);
    %the output is du, u for plant should be du+us
    u_state(:,i)=du+us;

    % use (1) to update x and y
    y_out(:,i)=C*x_state(:,i);
    x_state(:,i+1)=A*x_state(:,i)+B*u_state(:,i)+Bp*pk;

    %update observer
    x_ob_temp=x_ob(:,i)+L*(y_out(:,i)-Caug*x_ob(:,i));
    x_ob(:,i+1)=Aaug*x_ob_temp+Baug*u_state(:,i);
end
```