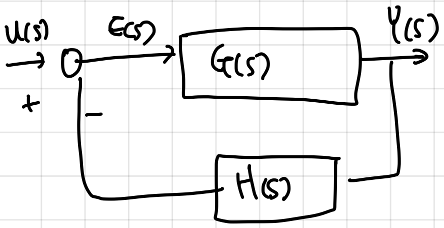
Jiyoon Park (andrew id : jiyoonp)

Fall 2022 16-642 Manipulation, Estimation, and Control Problem Set 2

#1

$$G(s) = \frac{200}{s^3+22s^2+141s+2}$$



a)

$Y(s) = G(s)E(s)$

$E(s) = U(s) - H(s)Y(s)$

$Y = G(U - HY) = GU - GHY$

$(Y + GHY) = GU$

$Y(1+GH) = GU$

$Y = \frac{G}{1+GH}U \Rightarrow T = \frac{G}{1+GH}$ so our closed loop transfer function is $T(s) = \frac{G(s)}{1+G(s)H(s)}$

we put $H(s)=1$ since we have negative unity feedback

$$T = \frac{G}{1+G} \Rightarrow T(s) = \frac{G(s)}{1+G(s)} = \frac{\frac{200}{s^3+22s^2+141s+2}}{1+\frac{200}{s^3+22s^2+141s+2}} = \frac{\frac{200}{s^3+22s^2+141s+2}}{\frac{s^3+22s^2+141s+202}{s^3+22s^2+141s+2}}$$

$$= \frac{200}{s^3+22s^2+141s+202}$$

$$\therefore T(s) = \frac{200}{s^3+22s^2+141s+202}$$

b) The "zeros" are roots of the numerator of $T(s)$ in our case, there are no "zeros".

The "poles" are the roots of the denominator of $T(s)$

$s^3+22s^2+141s+202 = 0$

$$\begin{array}{r|rrrr} -2 & 1 & 22 & 141 & 202 \\ & & -2 & -40 & -202 \\ \hline & 1 & 20 & 101 & \boxed{0} \end{array}$$

$(s+2)(s^2+20s+101)$

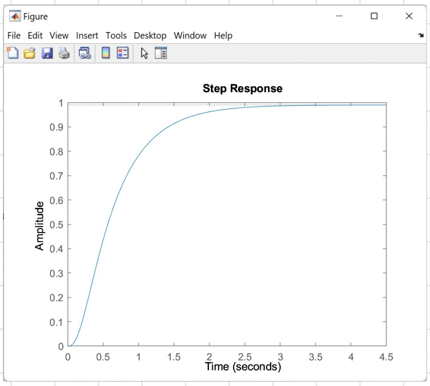
$s = -2, -10 \pm \sqrt{100-101}$

$s = -2, -10 \pm i$

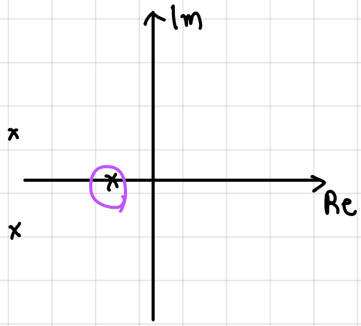
poles of T are $-2, -10 \pm i$

$\therefore$ There are no zeros and poles are $[-2, -10+i, -10-i]$

c)



The poles of the system are [-2, -10+i, -10-i] in the complex plane.



-2 is the dominant pole since it is closer to the imaginary axis. And when you look at the step response, we can see that the system looks like a overdamped system since it does not get effected by the underdamped $-10\pm i$ and it acts like a first order system with dominant pole -2.

∴ pole -2 dominates the step response

d) To get the steady state value,

$$Y_s = \lim_{t\to\infty} y(t) = \lim_{s\to 0} T(s) = \lim_{s\to 0} \frac{200}{s^3+22s^2+141s+202} = \frac{200}{202} = \frac{100}{101}$$

∴ The steady state value is $\frac{100}{101}$

#2

req: 1) rise time 0.5s
2) max % overshoot < 5%
3) steady state error = 0

$$G(s) = \frac{s+10}{s^4+71s^3+1070s^2+1000s}$$

let $G'(s) = \frac{K_d s^2 + K_p s + K_i}{s}$ since we are using a PID controller.

1) $|p| > \frac{1.8}{0.5}$

2) $\operatorname{atan}\left(\frac{-\mathrm{Re}(p_1)}{\mathrm{Im}(p_1)}\right) > 45°$

let $G(s) = \frac{B_G}{A_G}$ $G'(s) = \frac{B_G'}{A_G'}$ then,

$$T_G = \frac{B_G B_G'}{A_G A_G' + B_G B_G'} = \frac{(s+10)(K_d s^2 + K_p s + K_i)}{s(s^4+71s^3+1070s^2+1000s) + (s+10)(K_d s^2+K_p s+K_i)}$$

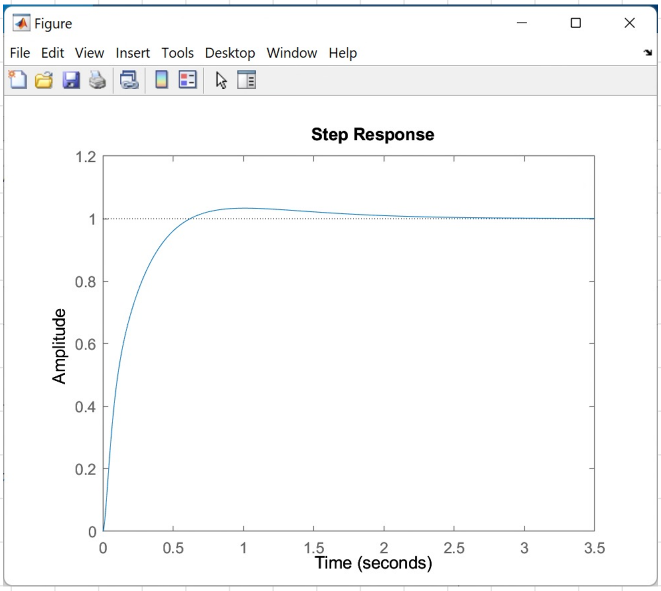
$$= \frac{K_d s^3 + K_p s^2 + K_i s + 10K_d s^2 + 10K_p s + 10K_i}{s^5+71s^4+1070s^3+1000s^2 + K_d s^3 + K_p s^2 + K_i s + 10K_d s^2 + 10K_p s + 10K_i}$$

$$= \frac{K_d s^3 + (K_p+10K_d)s^2 + (K_i+10K_p)s + 10K_i}{s^5+71s^4+(1070+K_d)s^4+(1000+(K_p+10K_d))s^2+(K_i+10K_p s)+10K_i}$$

I implemented this T_G on matlab and tuned K_p, K_d, K_i according to the steps given in the lecture notes (p 20)

The final gains I used is $K_p = 800$ $K_d = 500$ $K_i = 1$

Resulting closed loop step response



To check if the req were met,

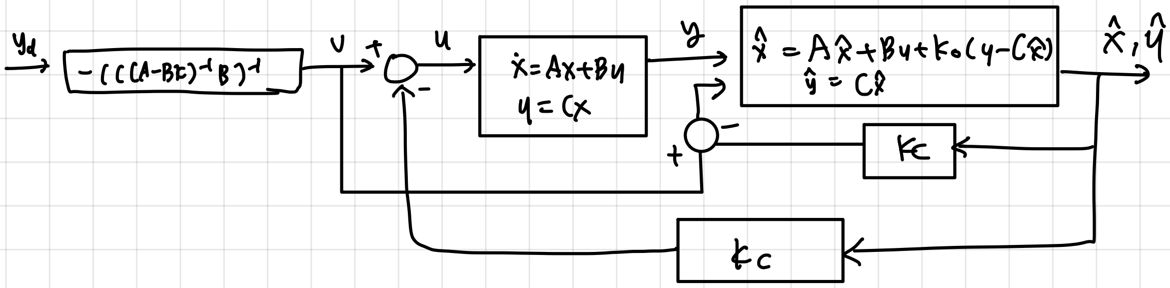
1) rise time 0.5s
2) max % overshoot < 5%
3) steady state error = 0

```
S = stepinfo(sys2);
disp(S.RiseTime);

    0.3646

disp(S.Overshoot);

    3.3563

[y, t] = step(sys2, 10000);
disp(y(end))

    1.0000
```

1) rise time was 0.3646s
2) max % overshoot was 3.3563 %
3) and the step response at t=10000 is 1. so the steady state error is 0.

#3

The new block diagram is as follows



To see if the system is observable, we need to check if $\text{rank}([C \ CA \ CA^2 \cdots CA^{n-1}]^T) = n$

I will use matlab to get the rank

```
Q_prime = [C' A'*C' (A')^2*C' (A')^3*C']

  Q_prime = 4x4
     39.3701         0         0         0
           0         0   39.3701 -118.1102
           0   39.3701 -118.1102  354.3307
           0         0         0   39.3701

rank(Q_prime)

  ans = 4

rank(obsv(A, C))

  ans = 4
```

the rank is 4 for

$\text{rank}(\underbrace{[C^T \ A^TC^T \ (A^T)^2C^T \ (A^T)^3C^T]}_{=Q_o}) = 4$

and using the matlab function

$\text{rank}(\text{obsv}(A, C)) = 4$.

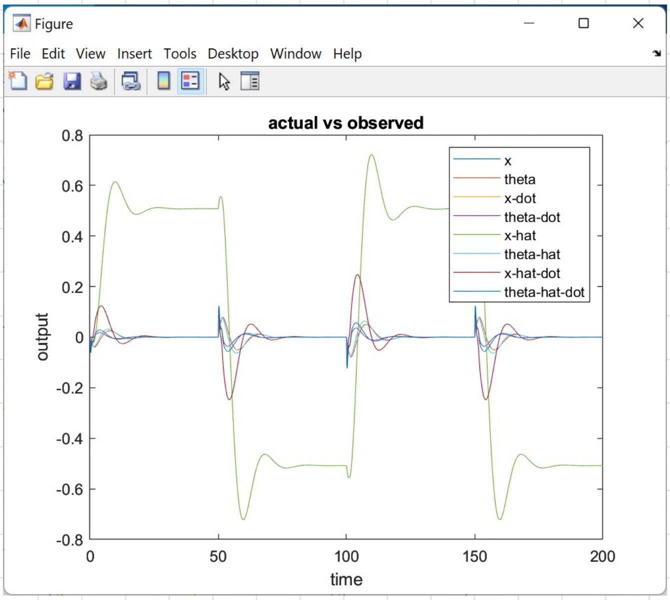
Since $\text{rank}(Q_o) = 4$, the system is observable.

To select K_o, I first looked at the poles of my original system. The poles were

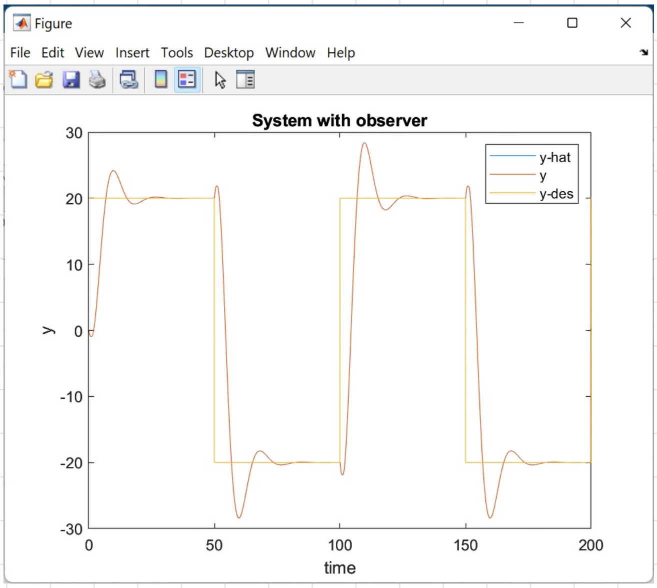
```
eig(A-B*Kc)

ans = 4×1

   -3.5927
   -1.1134
   -0.7726
   -0.3618
```

So, I selected [-3.9 -4 -4.1 -4.2] for my poles. Since it had to be on the further left of the original poles. It is odd since theoretically (in class) the further down I place the pole, it should give better performance but >>ode45 breaks if I give poles that are more negative.

① The plot of the actual state vs observed state is

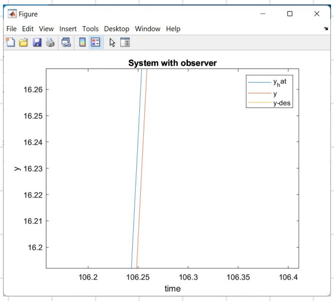


② The plot of y-des vs y-output for state is

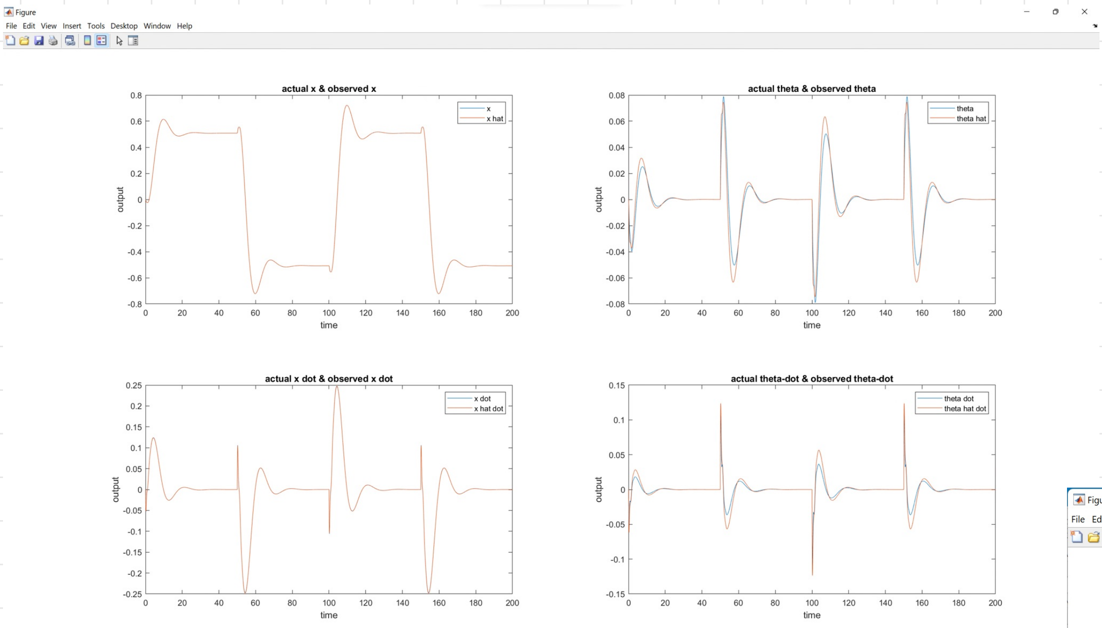


③ I plotted the real state and observed state. The errors btw them were big when y_{des} changed drastically. (t=0, 50, 100, 150, 200)

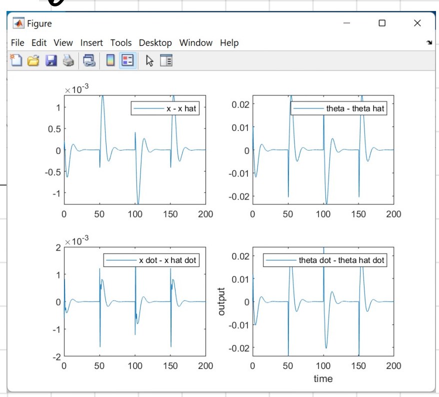
($\hat{y}$ is very similar to y, so it is hard to see but it is there)



plot of actual X and observed X by $x, \theta, \dot{x}, \dot{\theta}$



the diffrence btw actual & observed.



Jiyoon Park MEC Problem2

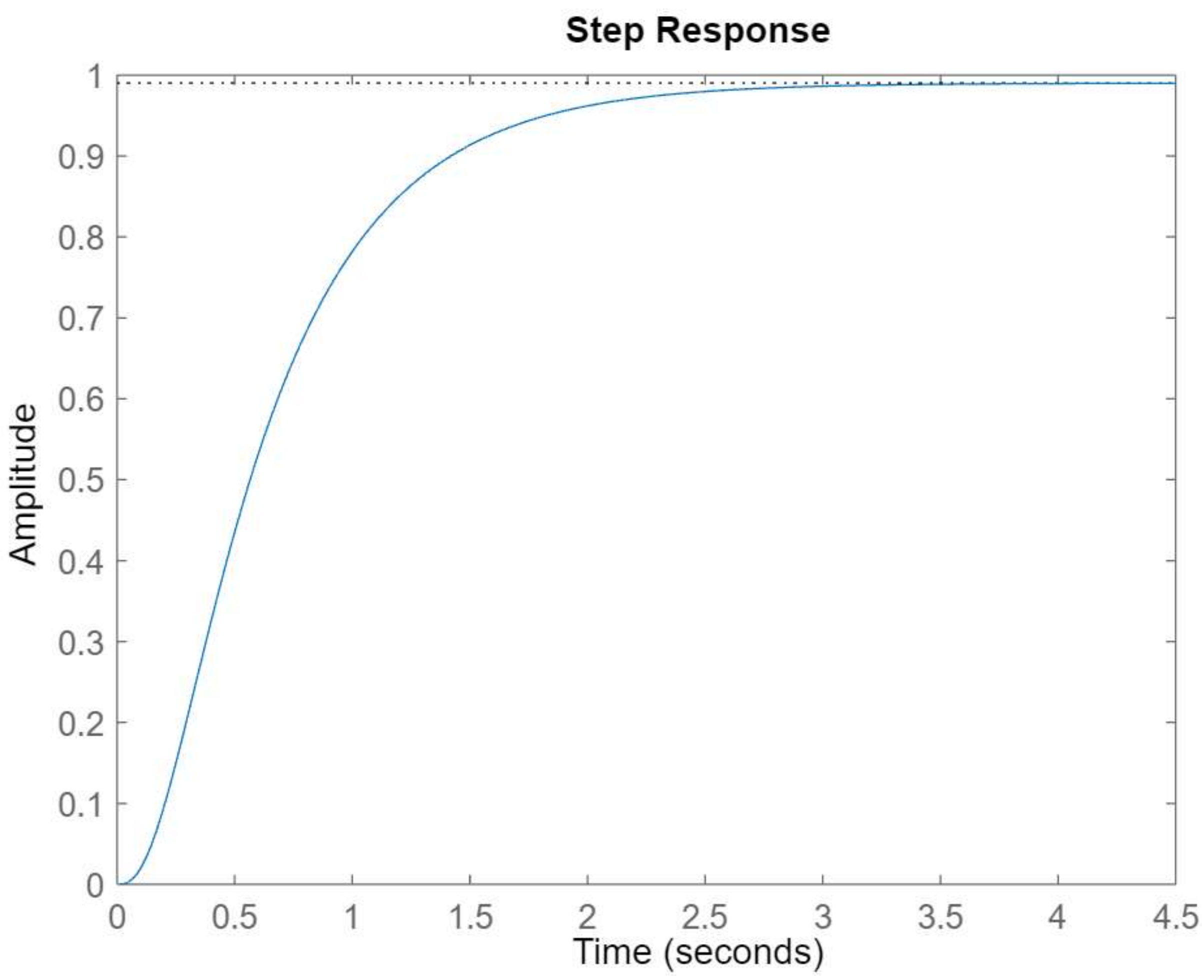
Question1

c)

```
sys = tf(200, [1 22 141 202])
```

```
sys =

            200
  --------------------------
  s^3 + 22 s^2 + 141 s + 202

Continuous-time transfer function.
```

```
step(sys)
```



Question2

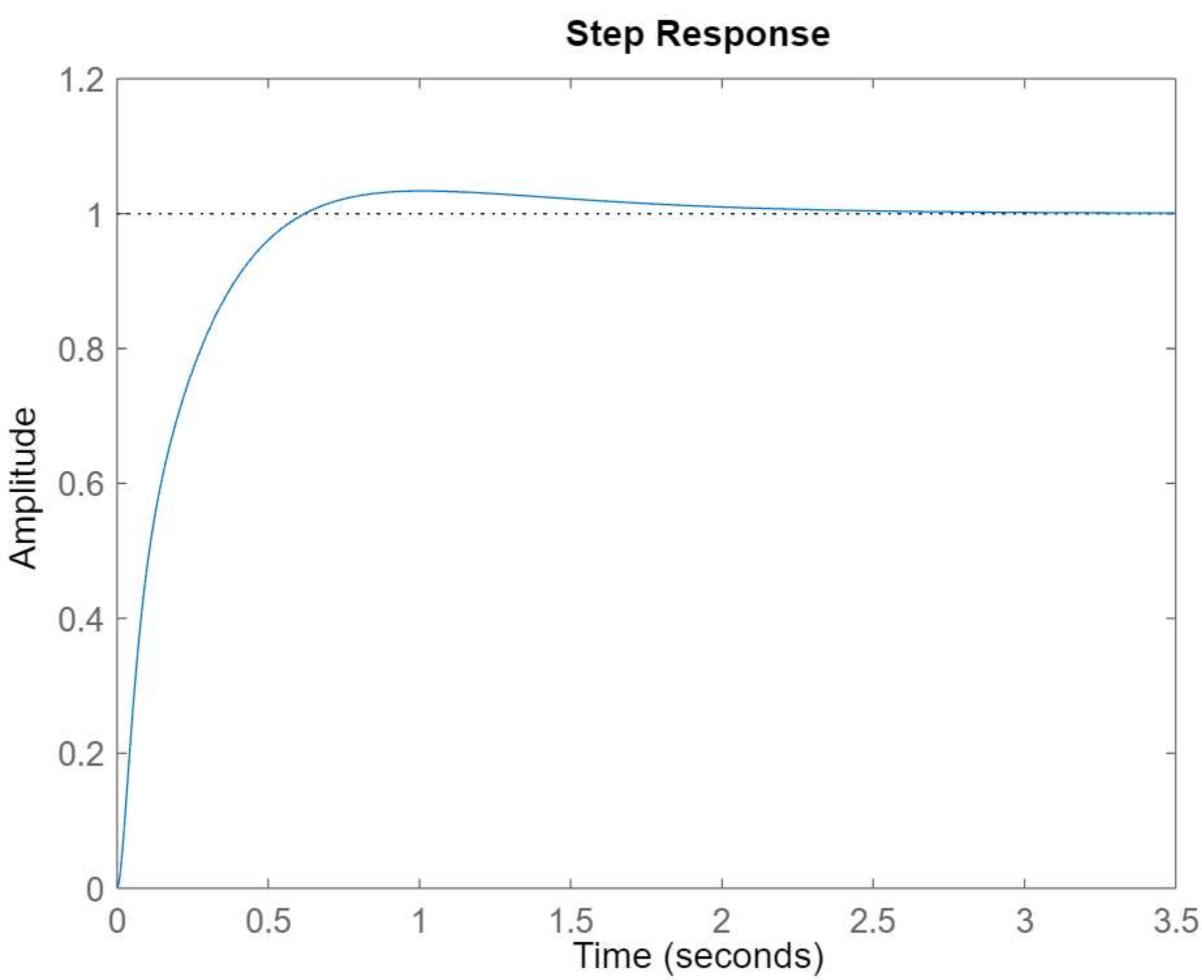
```
kp = 800;
kd=500;
ki = 1;

sys2 = tf([kd kp+10*kd ki+10*kp 10*ki],[1 71 1070+kd kp+10*kd+1000 ki+10*kp 10*ki])
```

```
sys2 =

  500 s^3 + 5800 s^2

              + 8001 s + 10

  --------------------------

  s^5 + 71 s^4 + 1570 s^3

        + 6800 s^2

              + 8001 s + 10
```

```
[y, t] = step(sys2, 10000);
disp(y(end))
```

```
    1.0000
```

```
step(sys2)
```




```
S = stepinfo(sys2);
disp(S.RiseTime);
```

```
    0.3646
```

```
disp(S.Overshoot);
```

```
    3.3563
```

Question3

```
A = [0 0 1 0;
     0 0 0 1;
     0 1 -3 0;
     0 2 -3 0;];
B = [0;0;1;1;];

T = 0.01;
time_all = 0:T:200;
y_des = 20*square(2*pi*T*time_all);
C = [39.3700787 0 0 0];
x0=[0;0;0;0];
```

```
Obs=[C C*A C*A^2 C*A^3]'
```

```
Obs = 16×1
   39.3701
         0
         0
         0
         0
         0
   39.3701
         0
         0
   39.3701
```

```
Q_prime = [C' A'*C' (A')^2*C' (A')^3*C']
```

```
Q_prime = 4×4
   39.3701         0         0         0
         0         0   39.3701 -118.1102
         0   39.3701 -118.1102  354.3307
         0         0         0   39.3701
```

```
rank(Q_prime)
```

```
ans = 4
```

```
rank(obsv(A, C))
```

```
ans = 4
```

```
obsv(A, C)
```

```
ans = 4×4
   39.3701         0         0         0
         0         0   39.3701         0
         0   39.3701 -118.1102         0
         0 -118.1102  354.3307   39.3701
```

```
Q =[10 0 0 0;
     0 5 0 0;
     0 0 10 0;
     0 0 0 5;];
R = 8;
Kc = lqr(A, B, Q, R);
E = -inv(C*inv(A-B*Kc)*B);

eig(A-B*Kc)
```

```
ans = 4×1
   -3.5927
   -1.1134
   -0.7726
   -0.3618
```

```
poles = [-3.9 -4 -4.1 -4.2];
poles1 = [-3.6 -3.65 -3.7 -3.75];
poles2 = [-4 -5 -6 -7];
K0 = place(A',C',poles)';

x_orig_list = zeros(length(time_all), 4);
x_hat_list = zeros(length(time_all), 4);
y_list = zeros(length(time_all), 1);
y_hat_list = zeros(length(time_all), 1);
% x_hat_list(1, :) = [0.01, 0.02, 0, 0];

tic;

for i= 1:length(time_all) - 1

disp(i)
% time = (i-1)*(T):T:i*T;

[t_orig, x_orig] = ode45(@(t, x) func3(t, x, E, Kc, x_hat_list(i, :)'), [(i-1)*T, i*T], x_orig_list(i, :)');
x_orig_list(i+1, :) = x_orig(end, :);
y_list(i+1, :) = C*x_orig(end,:)';


[t_obs, x_hat] = ode45(@(t, x) func4(t, x, E, y_list(i+1, :), Kc, K0, C), [(i-1)*T, i*T], x_hat_list(i, :)');

x_hat_list(i+1, :) = x_hat(end, :);
y_hat_list(i+1, :) = C*x_hat(end, :)';


end
```

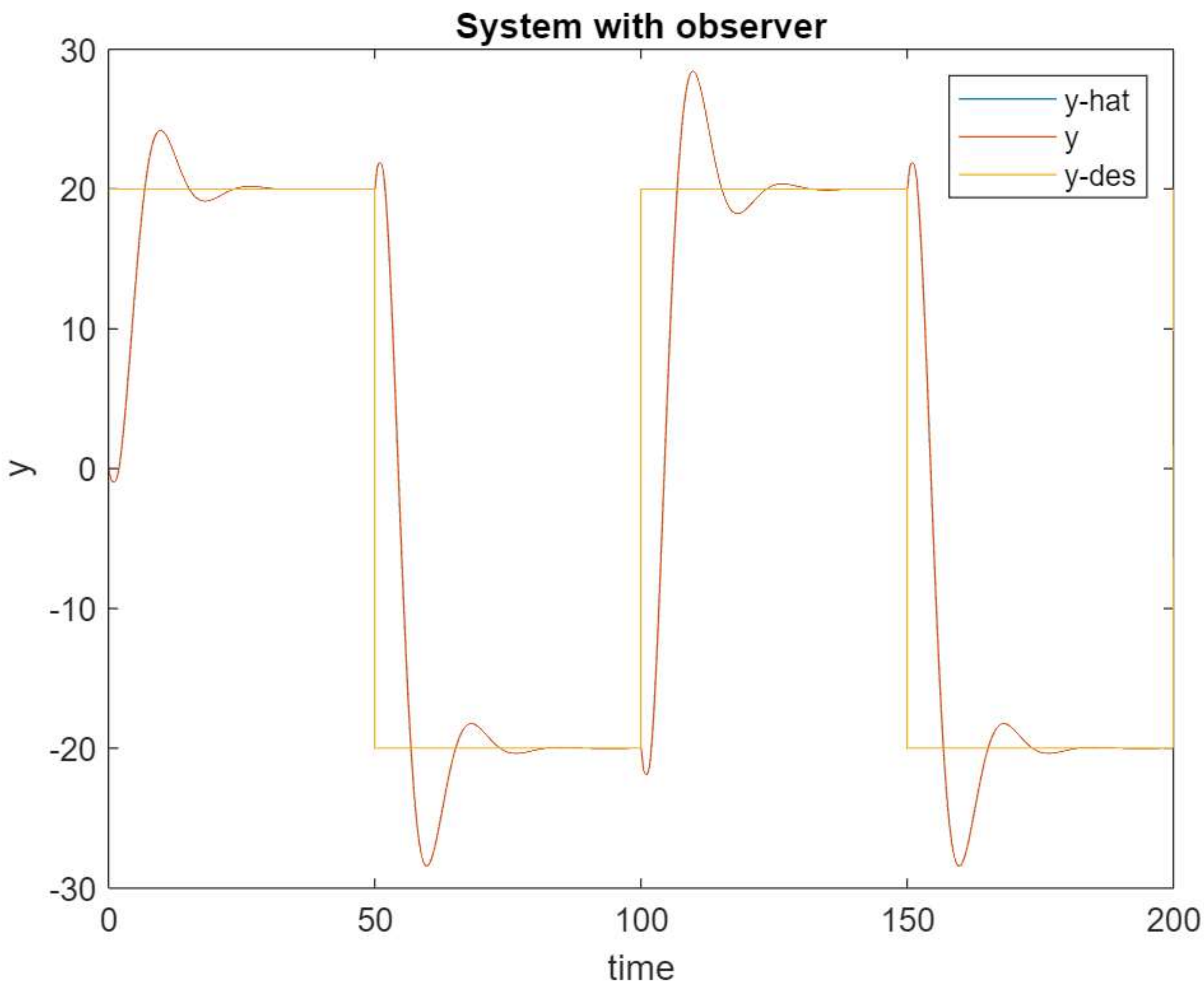
```
1
2
3
4
5
6
7
8
```

```
toc;
```

```
Elapsed time is 38.493544 seconds.
```

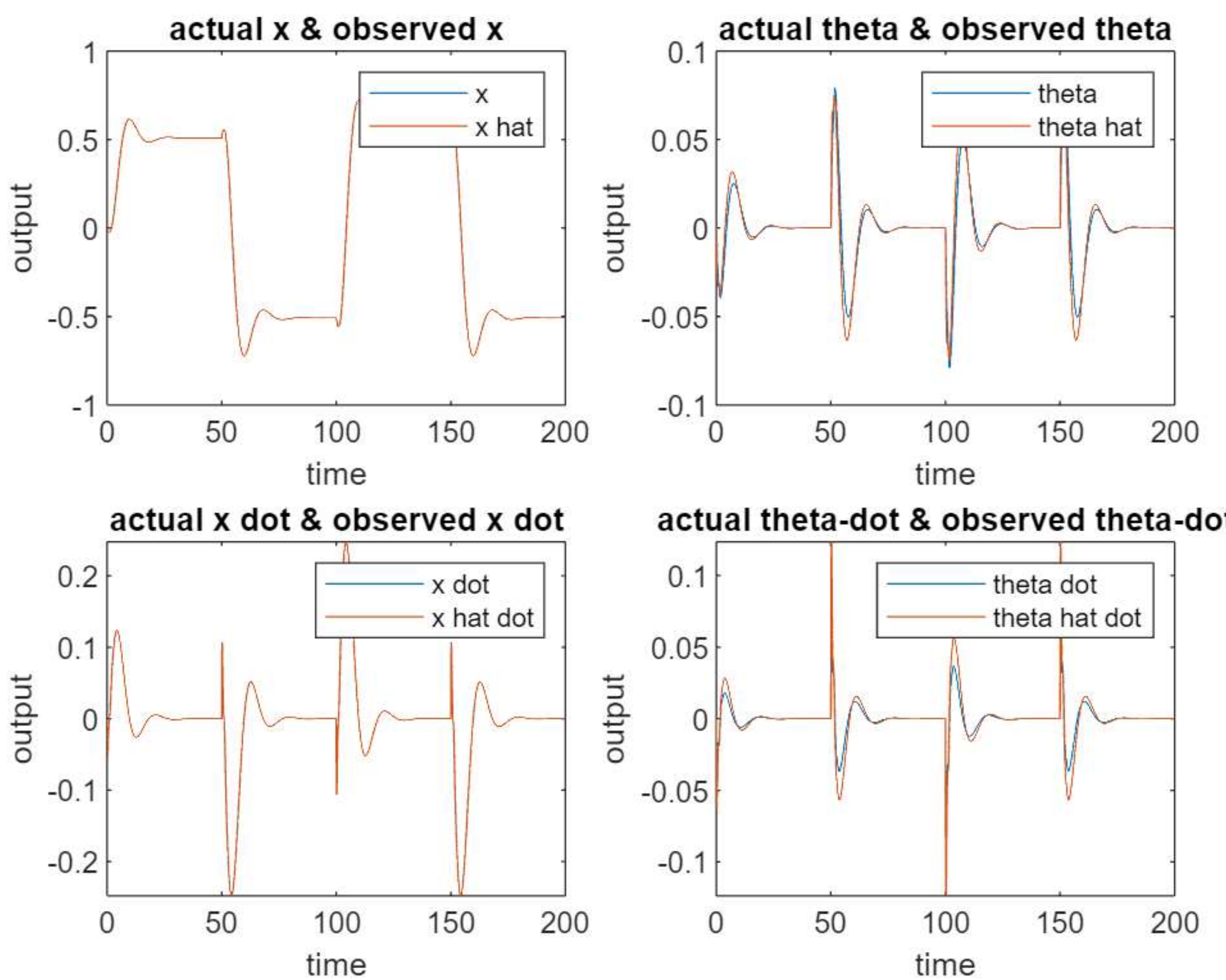
```
figure;
plot(time_all, y_hat_list)
hold on
plot(time_all, y_list)
hold on
plot(time_all, y_des)
hold on
%plot(t3(:,1), C*y3')
title('System with observer')
```

```
legend('y-hat', 'y', 'y-des')
xlabel('time')
ylabel('y')
```

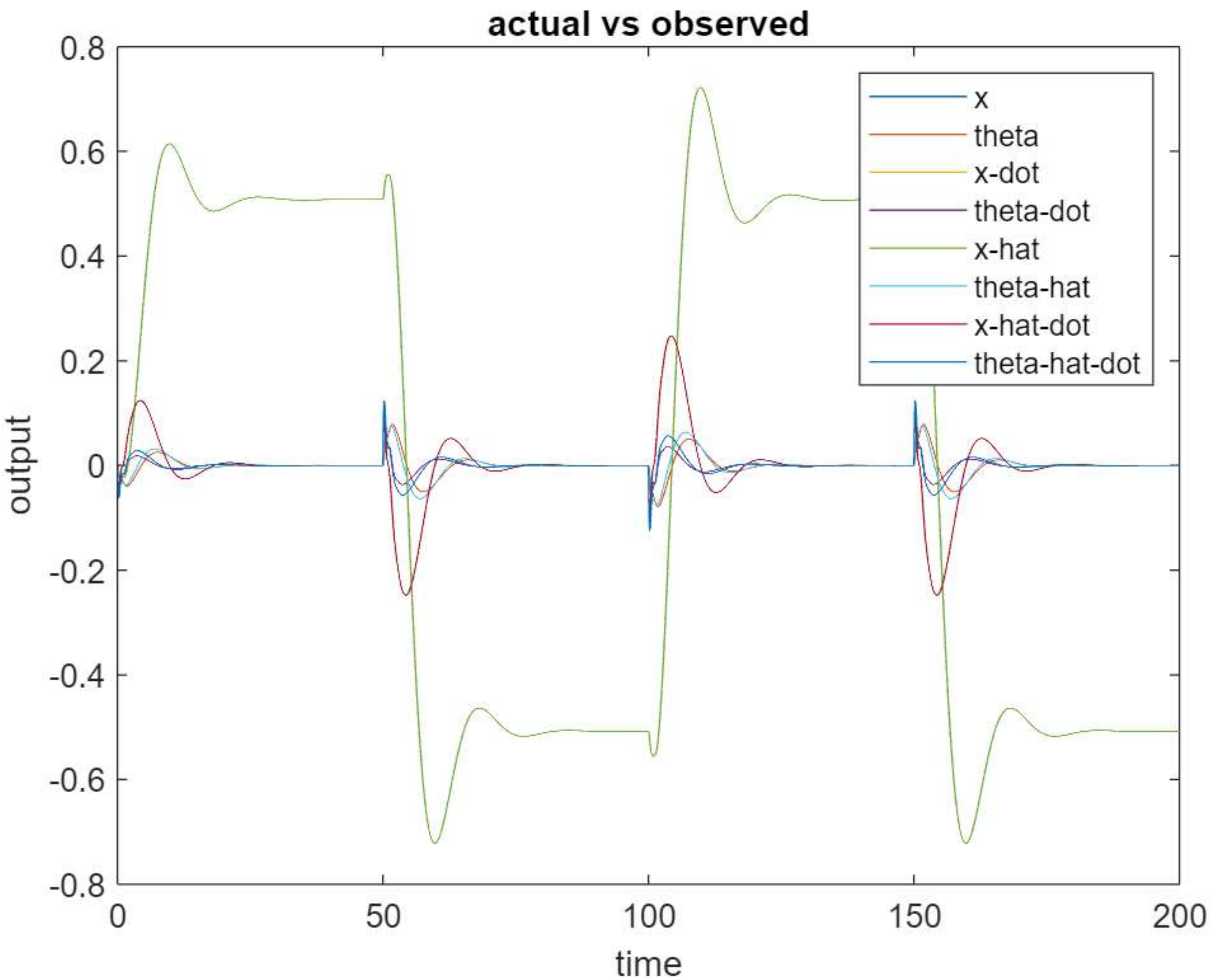



```
figure;

subplot(221);
plot(time_all, x_orig_list(:,1))
hold on
plot(time_all,x_hat_list(:,1))
legend('x','x hat')
hold on
xlabel('time')
ylabel('output')
title('actual x & observed x')


subplot(222);
plot(time_all, x_orig_list(:,2))
hold on
plot(time_all,x_hat_list(:,2))
legend('theta','theta hat')
hold on
xlabel('time')
ylabel('output')
title('actual theta & observed theta')

subplot(223);
plot(time_all, x_orig_list(:,3))
```

```
hold on
plot(time_all,x_hat_list(:,3))
legend('x dot','x hat dot')
hold on
xlabel('time')
ylabel('output')
title('actual x dot & observed x dot')

subplot(224);
plot(time_all, x_orig_list(:,4))
hold on
plot(time_all,x_hat_list(:,4))
legend('theta dot','theta hat dot')
hold on
xlabel('time')
ylabel('output')
title('actual theta-dot & observed theta-dot')
```




```
figure;

plot(time_all, x_orig_list)
hold on
plot(time_all, x_hat_list)
hold on
title('actual vs observed')
legend('x', 'theta','x-dot', 'theta-dot', 'x-hat', 'theta-hat','x-hat-dot', 'theta-hat-dot')
xlabel('time')
ylabel('output')
```




```
function out3 = func3(t, x, E, Kc, x_hat)

    y_des = 20*square(2*pi*0.01*t);
% y_des = 2;
%     disp(y_des)
    V = E*y_des;
    U = V-Kc*x_hat;
    out3 = [x(3);
            x(4);
            (U-x(4)^2*sin(x(2))-3*x(3)+cos(x(2))*sin(x(2)))/(2-cos(x(2))^2);
            (U*cos(x(2))-x(4)^2*cos(x(2))*sin(x(2))-3*x(3)*cos(x(2))+2*sin(x(2)))/(2-cos(x(2))^2)];
end

function out4 = func4(t, x, E, y, Kc, K0, C)
%     disp(t)
    y_des = 20*square(2*pi*0.01*t);
%     y_des = 2;
    V = E*y_des;
    U = V-Kc*x;
    out4 = [x(3);
            x(4);
            (U-x(4)^2*sin(x(2))-3*x(3)+cos(x(2))*sin(x(2)))/(2-cos(x(2))^2);
            (U*cos(x(2))-x(4)^2*cos(x(2))*sin(x(2))-3*x(3)*cos(x(2))+2*sin(x(2)))/(2-cos(x(2))^2)];
    out4 = out4+ K0*(y-C*x);
end
```