AE 353 Homework 6 Solutions

1. State-space form:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{5} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{5} \end{bmatrix} u$$

$$y = [1\ 0]x$$

a) Let us choose the following weights for designing the optimal controller:

$$Q_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_c = \rho = 1$$

On using the LQR command in MATLAB, we get:
>> K = lqr(A,B,Qc,Rc)

$$\boldsymbol{K} = [\mathbf{1} \quad \mathbf{2.4641}]$$

b) Let us choose the following weights for designing the optimal controller:

$$R_o = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad Q_o = \gamma = 1$$

On using the LQR command in MATLAB, we get:
>> L = lqr(A',C',inv(Ro),inv(Qo))'

$$\boldsymbol{L} = \begin{bmatrix} \mathbf{1.5549} \\ \mathbf{0.7088} \end{bmatrix}$$

c) We have seen the closed loop system for this system is of the form:

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} \mathrm{A} & -\mathrm{BK} \\ \mathrm{LC} & \mathrm{A-LC-BK} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} \mathrm{B} \\ \mathrm{B} \end{bmatrix} k_{reference} r$$

$$A_{cl} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.2 & -0.2 & -0.4928 \\ 1.5549 & 0 & -1.5549 & 1 \\ 0.7088 & 0 & -0.9088 & -0.6928 \end{bmatrix}$$

$$k_{ref} = -(C(A-BK)^{-1}B)^{-1} = 1$$

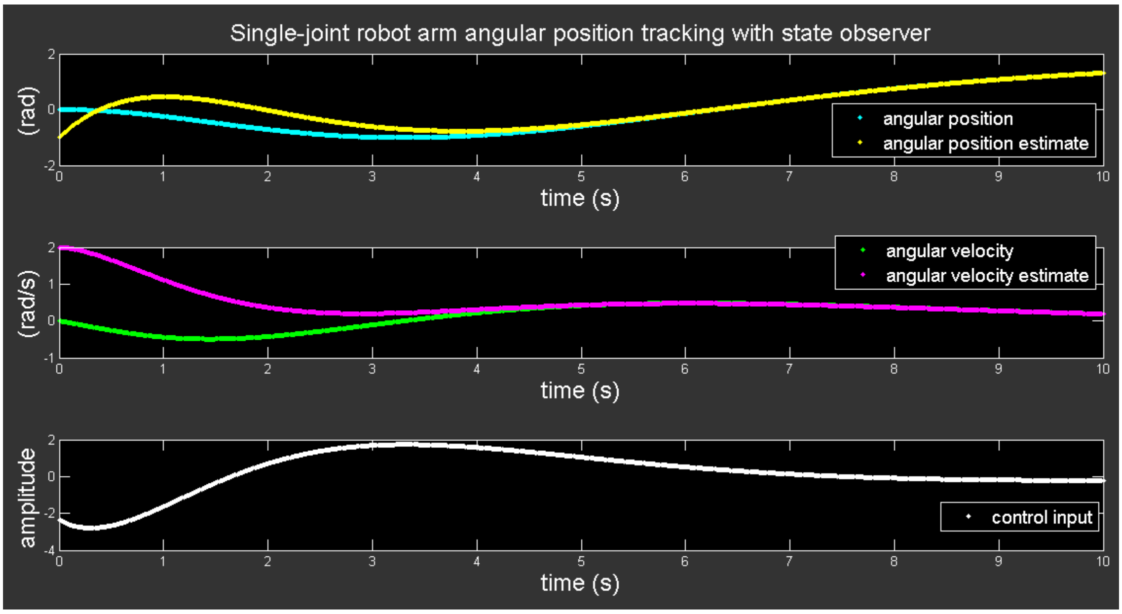
$$B_{cl} = \begin{bmatrix} 0 \\ 0.2 \\ 0 \\ 0.2 \end{bmatrix}$$

The eigenvalues for the above system are:

$$\lambda = -0.8774 \pm 0.5j, -0.3464 \ \pm 0.2828j$$

d) For simulating the closed loop system, we can use the codes provided in the solution for homework 5. The only difference is the way we calculate K and L.

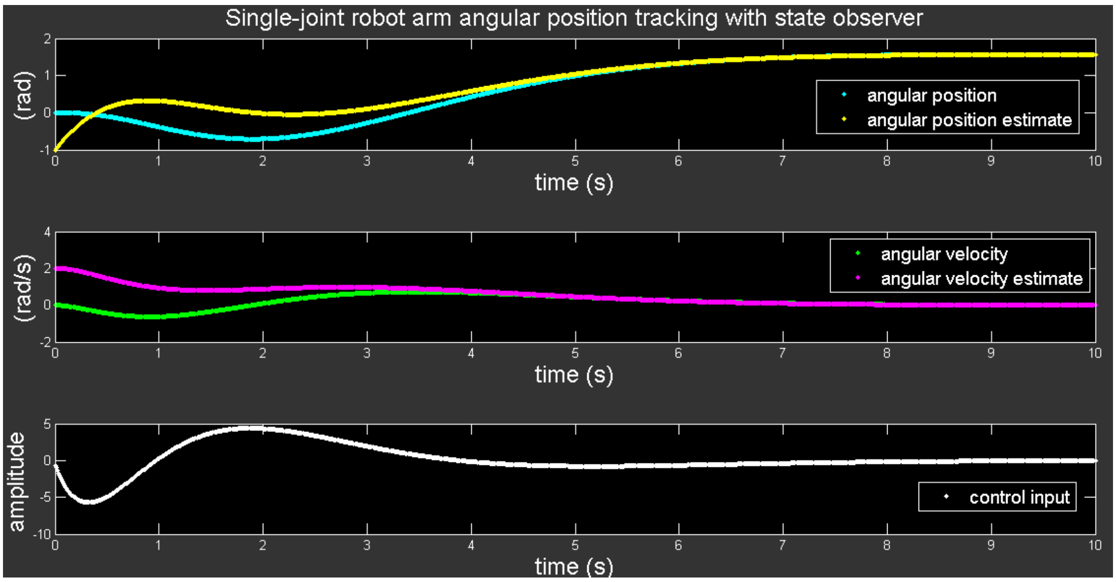
```
A = [0 1;0 -1/5];
B = [0;1/5];
C = [1 0];
D = 0;
Qc = [1 0; 0 1];
Rc = 1;
Qo = 1;
Ro = [1 0; 0 1];

% observer design
L = lqr(A', C', inv(Ro), inv(Qo))';

% controller design
K = lqr(A, B, Qc, Rc);
kref = -inv(C*(inv(A-B*K))*B);

% response
r = pi/2;
X0 = [0;0;-1;2];

tspan = linspace(0,10,1000);

[T,X] = ode45(@(t,X)stateSpaceEOM(t,X,K,kref,r,A,B,C,D,L),tspan,X0);

% recover the control history
uhistory = zeros(length(T),1);

for i = 1:length(T)
uhistory(i) = -K*X(i,3:4)' + kref*r;
end

function [Xdot u] = stateSpaceEOM(t,X,K,kref,r,A,B,C,D,L)

% let's separate the true states and the estimated states
x = X(1:length(X)/2);
xhat = X(length(X)/2+1:end);
y = C*x;

% controller
u = -K*xhat + kref*r;
% state-space representation of the true and estimated dynamics
xdot = A*x + B*u;
xhatdot = A*xhat + B*u - L*(C*xhat - y);

% package the state time derivative vector
Xdot = [xdot;xhatdot];
```



e) We have to choose ρ and γ such that the angular position has a time-to-peak of less than 0.5 seconds and overshoot of less than 10%. In the current response we can see that the angular position has a time-to-peak of about 3 seconds (notice the peak is negative, and this is called *undershoot*) and 0 overshoot.
There are many ways of going about your design process.
One way is to look at the cost function for the infinite-horizon LQR:

$$\int_0^\infty \left(x^T Q x + u^T R u\right) dt$$

We aim to minimize the above cost function. We can see that lower values of R allow us to choose higher magnitudes of the input u. Intuitively, a higher magnitude input would "force" the system to respond quicker, which should result in a lower time-to-peak. We can check this by designing an optimal controller using $R_c = \rho = 0.01$.

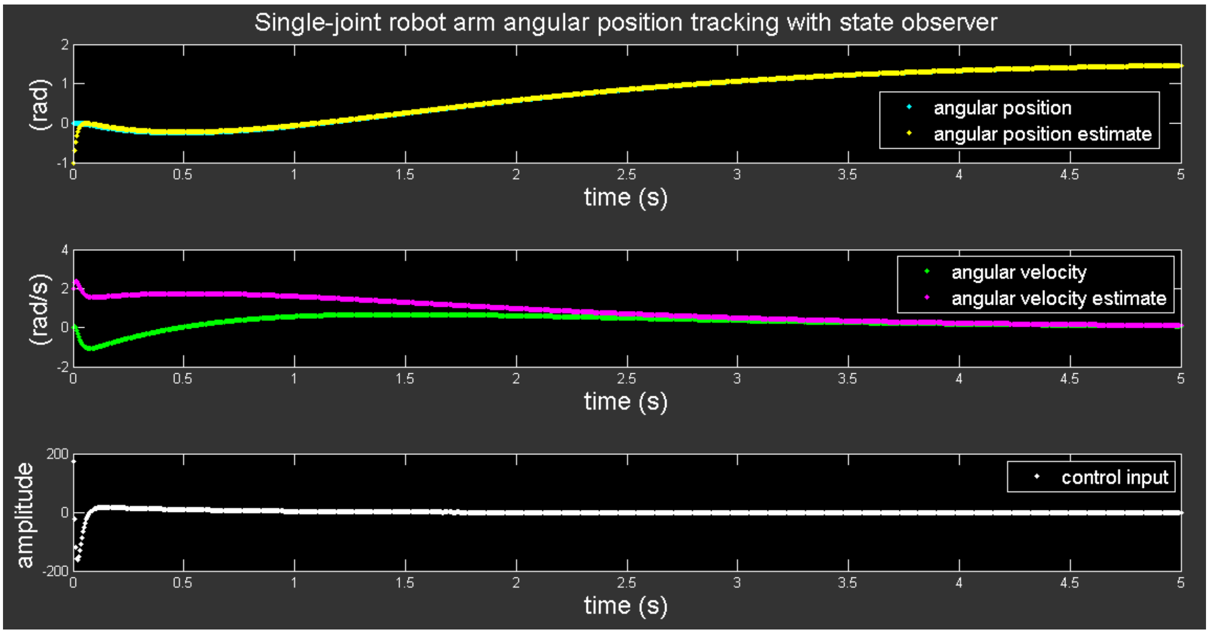


From the above plots we can see that the time-to-peak reduced to about 2 seconds.

Similarly, we can make the observer response faster by reducing the term *inv(Qo)*, that is by choosing a larger γ.

After some trial and error, we can choose the following values:

$$Q_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_c = 10^{-5}, \qquad Q_o = 5000, \qquad R_o = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



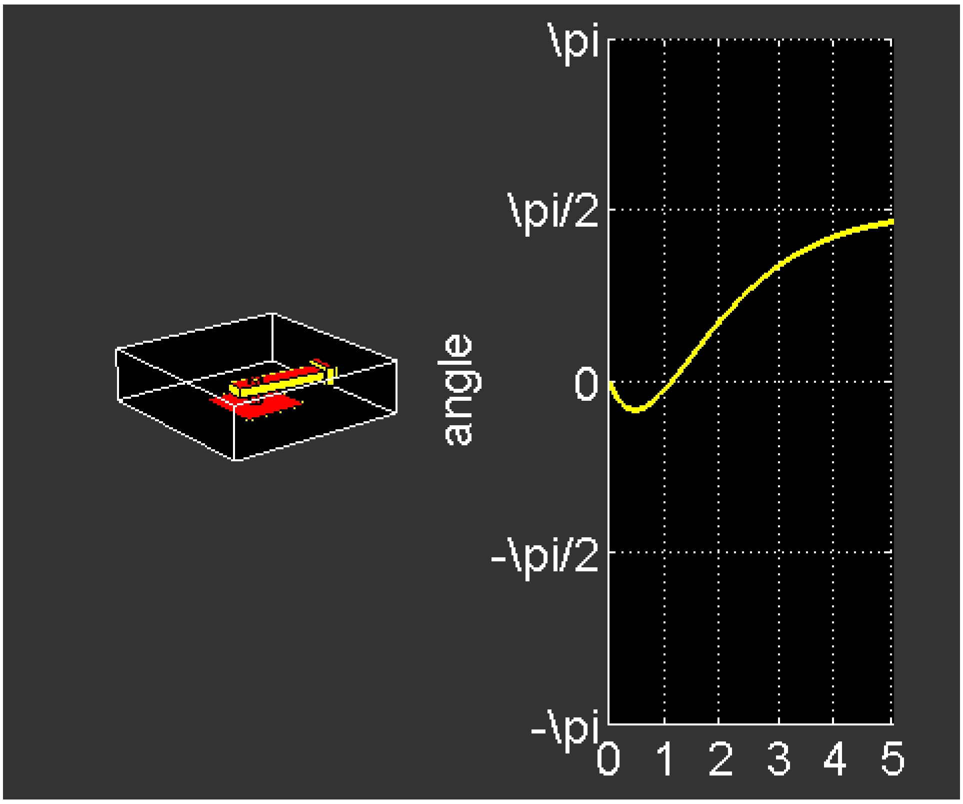
Thus, in the above plots we can see a time-to-peak of just under 0.5 seconds and zero overshoot.

f) Here's how the modified ControlLoop function looks like:

```
function [u,userdata] = ControlLoop(y,r,userdata,params)

persistent isFirstTime
if isempty(isFirstTime)
    isFirstTime = false;
    fprintf(1,'initialize control loop\n');

    userdata.xhat = [-1;2];

    Qc = eye(2);     Rc = 10^-5;
    userdata.K = lqr(params.A, params.B, Qc, Rc);

    Qo = 5000;       Ro = eye(2);
    userdata.L = lqr(params.A', params.C', inv(Ro), inv(Qo))';

    userdata.kref = -inv(params.C*inv(params.A -
params.B*userdata.K)*params.B);

end
u = -userdata.K*userdata.xhat + userdata.kref*r;
```

```
userdata.xhat = userdata.xhat + params.dt*(params.A*userdata.xhat +
params.B*u ...
- userdata.L*(params.C*userdata.xhat - y));
```

The response looks like:



2. State-space form:

$$\dot{x} = \begin{bmatrix} 0 & 9 \\ -9 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

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

a) Let us choose the following weights for designing the optimal controller:

$$Q_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_c = \rho = 1$$

On using the LQR command in MATLAB, we get:
>> K = lqr(A,B,Qc,Rc)

$$\boldsymbol{K} = [\mathbf{1.4131} \quad \mathbf{-0.0554}]$$

b) Let us choose the following weights for designing the optimal controller:

$$R_o = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad Q_o = \gamma = 1$$

On using the LQR command in MATLAB, we get:
>> L = lqr(A',C',inv(Ro),inv(Qo))'

$$\boldsymbol{L} = \begin{bmatrix} -0.0554 \\ 1.4131 \end{bmatrix}$$

c) The closed loop system for this system is of the form:

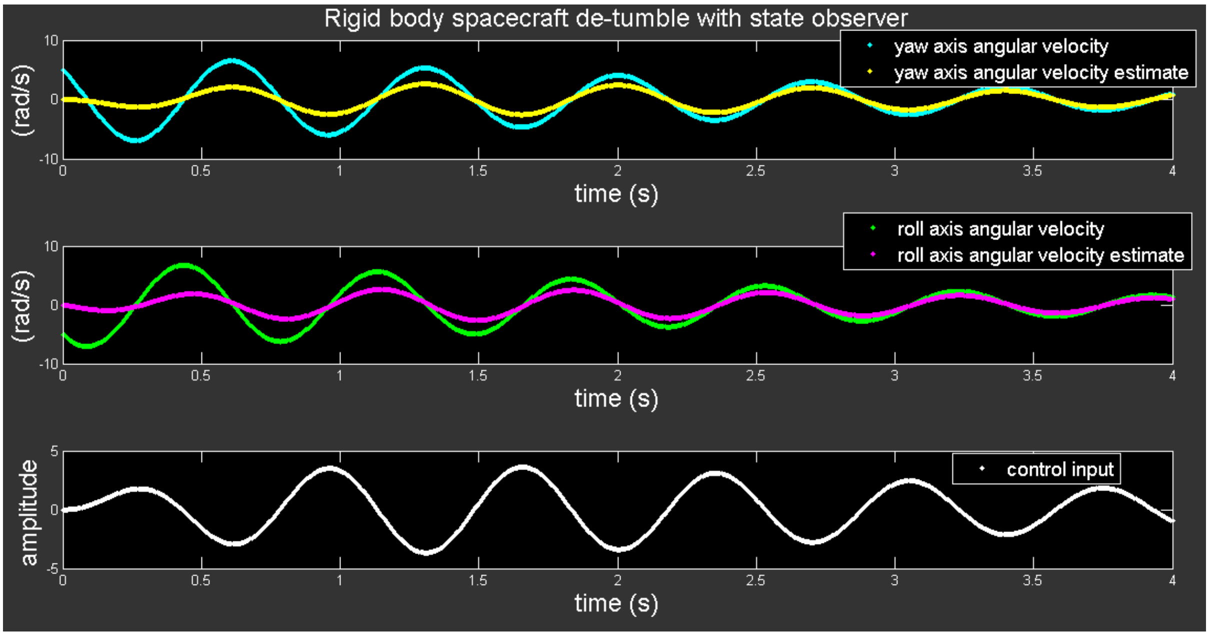
$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & \mathbf{A}-\mathbf{LC}-\mathbf{BK} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

$$A_{cl} = \begin{bmatrix} 0 & 9 & -1.4131 & 0.0554 \\ -9 & 0 & 0 & 0 \\ 0 & -0.0554 & -1.4131 & 9.1108 \\ 0 & 1.4131 & -9 & -1.4131 \end{bmatrix}$$

The eigenvalues for the above system are:

$$\lambda = -0.7066 \pm 9j, -0.7066 \pm 9j$$

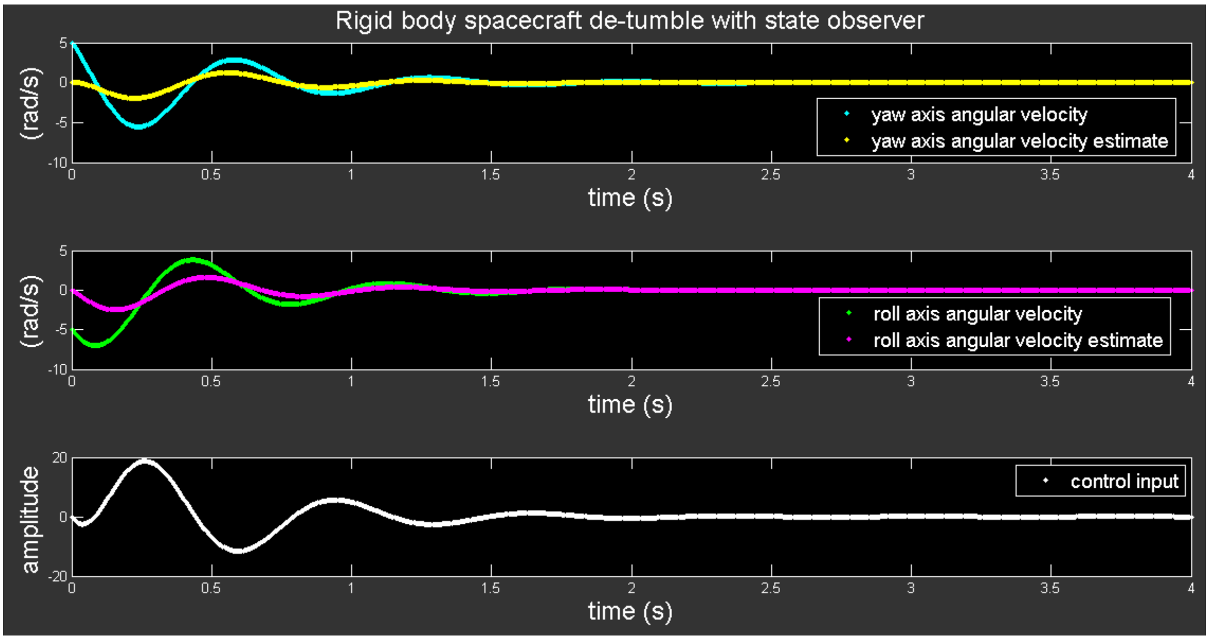
d) Again, using the code from homework 5 solutions, we get the following plots:



e) Using the same method as in Problem 1e, we can select a low ρ and a high γ to make the system faster, but here we also need to check the input magnitude (which cannot be greater than 20). Therefore, after some trial and error the following matrices were chosen:

$$Q_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_c = 0.01, \qquad Q_o = 10, \qquad R_o = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The response looks like:

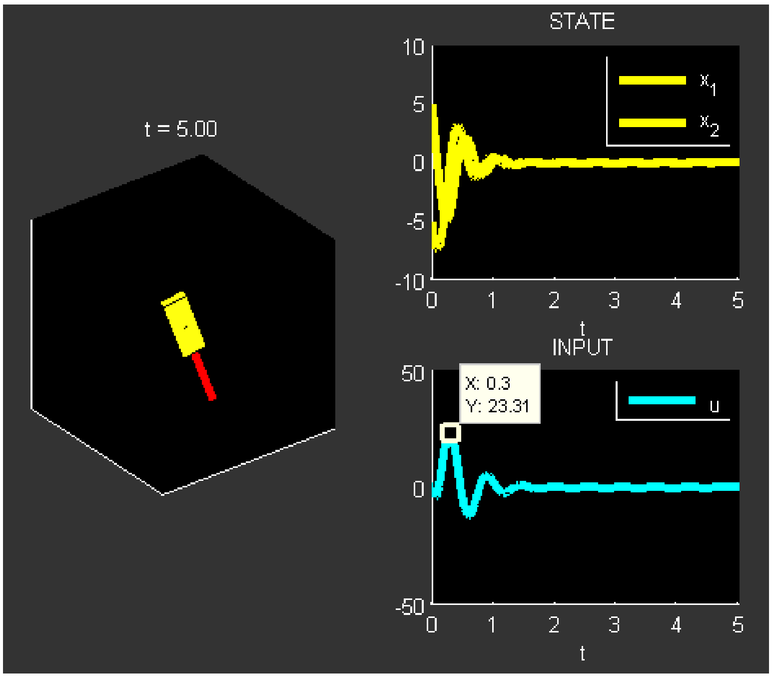


As we can see, the 2% settling time is well within 4 seconds and the input magnitude is always less than 20.

f) The ControlLoop function:

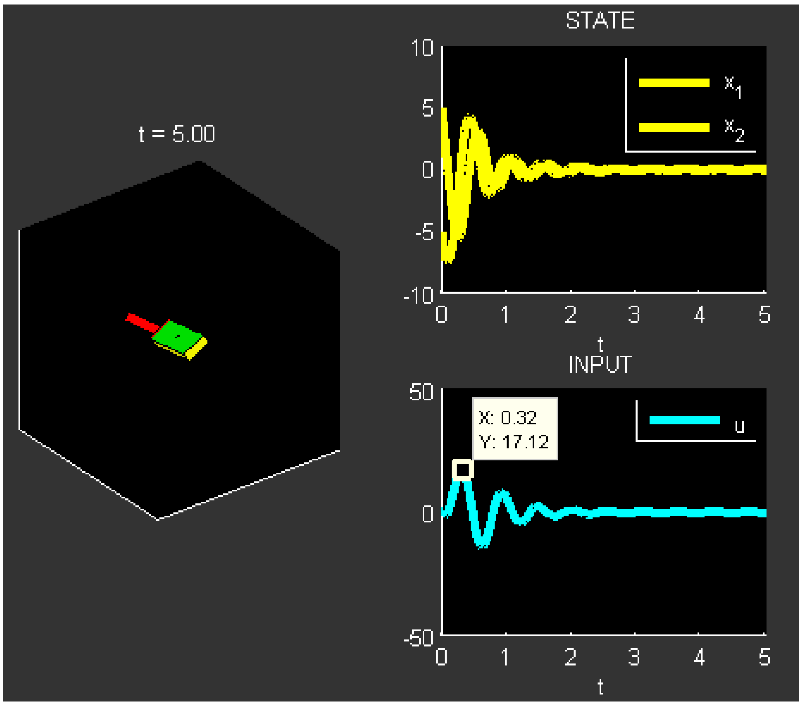
```
function [u,userdata] = ControlLoop(y,userdata,params)

persistent isFirstTime
if isempty(isFirstTime)
    isFirstTime = false;
    fprintf(1,'initialize control loop\n');

    userdata.xhat = [0;0];

    Qc = eye(2);    Rc = 0.01;
    userdata.K = lqr(params.A, params.B, Qc, Rc);

    Qo = 10;        Ro = eye(2);
    userdata.L = lqr(params.A', params.C', inv(Ro), inv(Qo))';

    userdata.kref = -inv(params.C*inv(params.A -
params.B*userdata.K)*params.B);

end
u = -userdata.K*userdata.xhat;
userdata.xhat = userdata.xhat + params.dt*(params.A*userdata.xhat +
params.B*u ...
- userdata.L*(params.C*userdata.xhat - y));
```

Here's how the response looked like:



The settling time is well within 4 seconds as expected. However the input magnitude now goes over 20. Choosing new parameters:

$$\rho = 0.05, \qquad \gamma = 10$$



Here both the conditions on the settling time and the input magnitude are met.

3. State-space system:

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -8 & 4 & -1 & 0.5 \\ 4 & -4 & 0.5 & -0.5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.5 \end{bmatrix} u$$

$$y = [0 \quad 1 \quad 0 \quad 0]$$

a) Let us choose the following weights for designing the optimal controller:

$$Q_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad R_c = \rho = 1$$

On using the LQR command in MATLAB, we get:
>> K = lqr(A,B,Qc,Rc)

$$\boldsymbol{K} = [\mathbf{0.0236} \quad \mathbf{0.1415} \quad \mathbf{0.3880} \quad \mathbf{0.8281}]$$

b) Let us choose the following weights for designing the optimal controller:

$$R_o = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad Q_o = \gamma = 1$$

On using the LQR command in MATLAB, we get:
>> L = lqr(A',C',inv(Ro),inv(Qo))'

$$\boldsymbol{L} = \begin{bmatrix} \mathbf{0.3816} \\ \mathbf{1.0237} \\ \mathbf{0.05} \\ \mathbf{0.024} \end{bmatrix}$$

c) The closed loop system for this system is of the form:

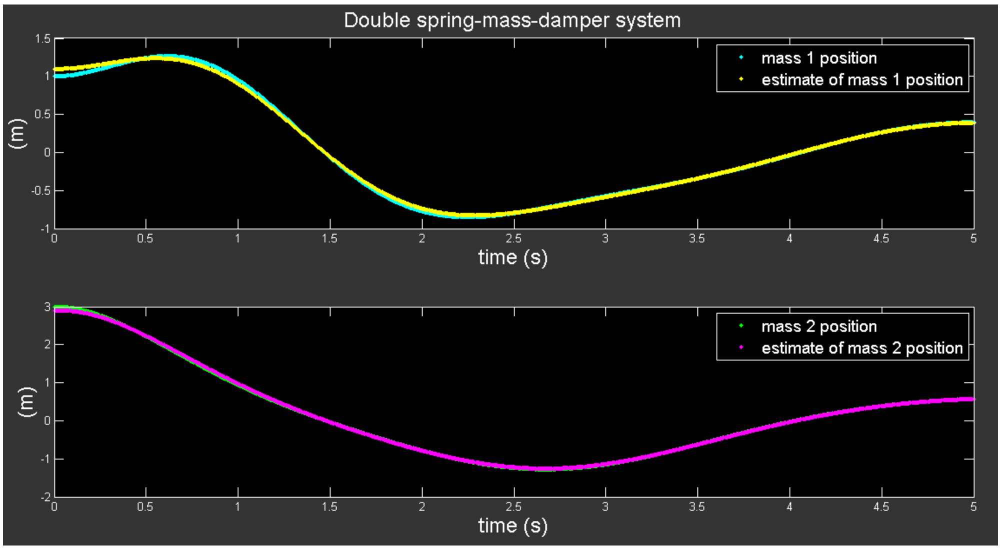
$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & \mathbf{A} - \mathbf{LC} - \mathbf{BK} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

$$A_{cl} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -8 & 4 & -1 & 0.5 & 0 & 0 & 0 & 0 \\ 4 & -4 & 0.5 & -0.5 & -0.0118 & -0.0708 & -0.1940 & -0.4141 \\ 0 & 0.3816 & 0 & 0 & 0 & -0.3816 & 1 & 0 \\ 0 & 1.0237 & 0 & 0 & 0 & -1.0237 & 0 & 1 \\ 0 & 0.05 & 0 & 0 & -8 & 3.95 & -1 & 0.5 \\ 0 & 0.024 & 0 & 0 & 3.9882 & -4.0947 & 0.3060 & -0.9141 \end{bmatrix}$$

The eigenvalues for the above system are:

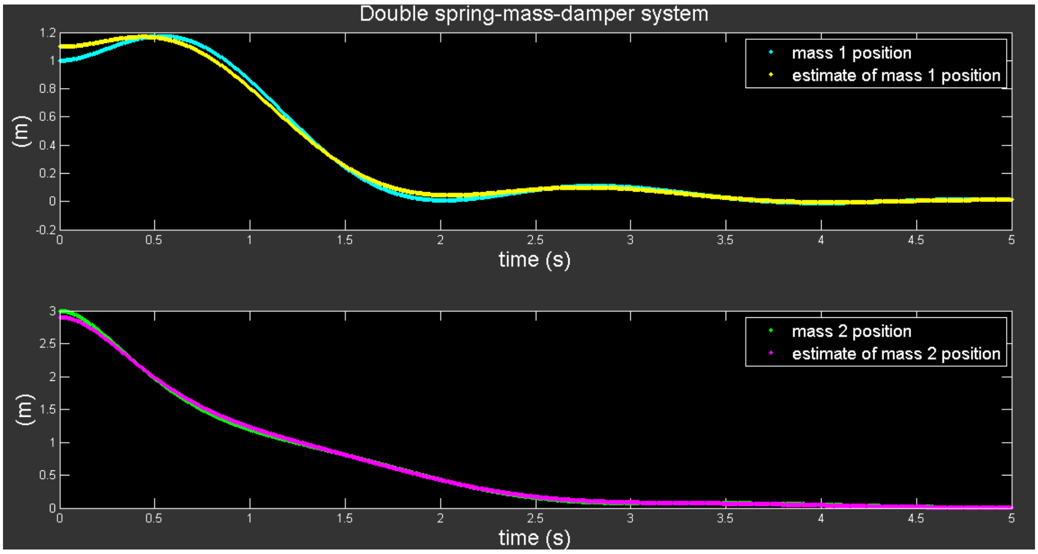
$$\lambda = -0.6685 \pm 3.1666j, -0.7128 \pm 3.1587j, -0.2885 \pm 1.2259j, -0.549 \pm 1.2071j$$

d) Again, using the code from homework 5 solutions, we get the following plots:



e) From the above plots we can see that we already seem to have a pretty good observer design. Using a similar method as in Problem 1e and Problem 2e we can reduce ρ to make the system faster.

Thus choosing: $\rho = 0.01$ and $\gamma = 1$, we get:



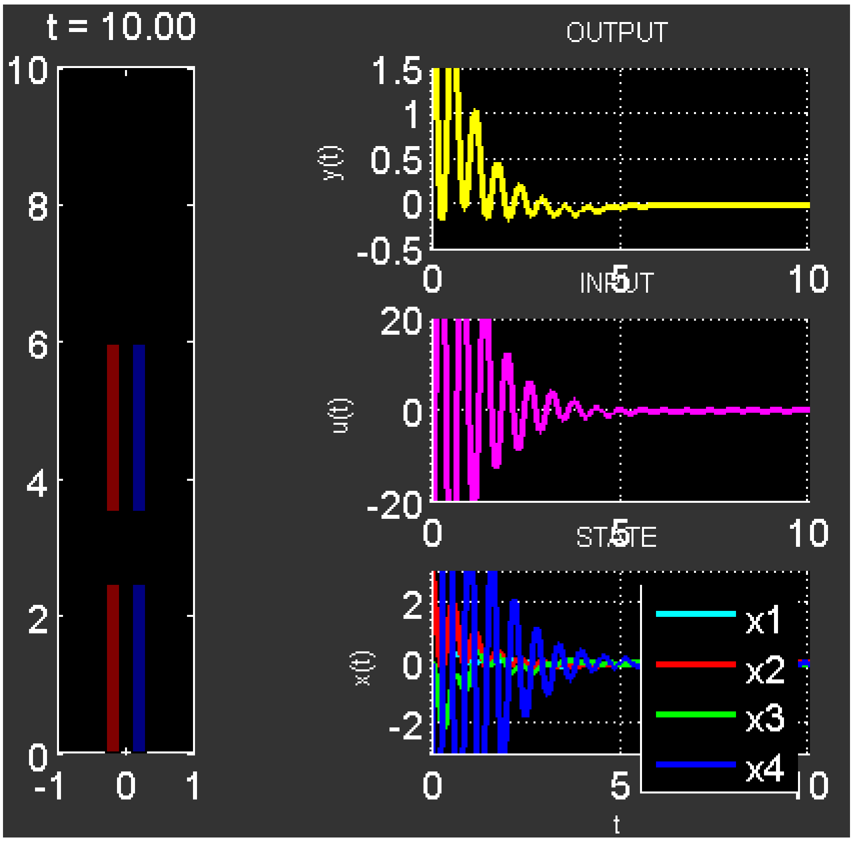
Here the 5% settling time is less than 5 seconds.

f) Here's the ControlLoop function:

Due to the disturbances the settling time is not less than 5 seconds anymore. Thus, we have to change our parameters.

One way to do this would be to look at the output for which we need to satisfy the settling time constraint. For our system $C = [0 \quad 1 \quad 0 \quad 0]$, which means that the position of the second mass is the output. Since we require this value to go to zero quickly, we can assign it a higher weight in the cost function. Therefore after some trial and error, the following matrices were chosen:

$$Q_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad R_c = \rho = 0.001, \qquad R_o = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad Q_o = \gamma = 100$$



The output satisfies the settling time constraints now.

4. Derivation of an optimal observer for a scalar system:

a) On comparing (1) with (2), we get the following:

$$f(x,d) = ax + bu + d$$
$$g(x,d) = q(cx - y)^2 + rd^2$$
$$h\big(x(t_0)\big) = s(cx(t_0) - y(t_0))^2$$

b) Taking partial derivatives of (5), we get:

$$\frac{\partial v}{\partial t} = \frac{\partial(px^2 + 2mx + n)}{\partial t} = \dot{p}x^2 + 2\dot{m}x + \dot{n}$$

$$\frac{\partial v}{\partial x} = \frac{\partial(px^2 + 2mx + n)}{\partial x} = 2px + 2m$$

c) Plugging the above in (3):

$$\frac{\partial v}{\partial t} = \min_{d}\left\{g - \left(\frac{\partial v}{\partial x}\right)f\right\}$$

$$\dot{p}x^2 + 2\dot{m}x + \dot{n} = \min_{d}\ \{q(cx - y)^2 + rd^2 - (2px + 2m)(ax + bu + d)\}$$

d) To minimize the right hand size w.r.t. d, let apply the first and second derivatives:

First derivative: $2rd - (2px + 2m)$ Second derivative: $2r$

Since the second derivative is positive, equating the first derivative to 0 gives us the value of d which minimizes the right hand side:

$$d = \frac{px + m}{r}$$

Plugging this value of d in the right hand side, we get:

$\dot{p}x^2 + 2\dot{m}x + \dot{n} =$

$$x^2\left(-\frac{p^2}{r} + qc^2 - 2pa\right) + x\left(-\frac{2pm}{r} - 2qcy - 2pbu - 2ma\right) + \left(-\frac{m^2}{r} + qy^2 - 2mbu\right)$$

e) Thus on comparing the coefficients of $x^2, x\ and\ 1$ in the result, we get:

$$\dot{p} = \left(-\frac{p^2}{r} + qc^2 - 2pa\right)$$
$$\dot{m} = \left(-\frac{pm}{r} - qcy - pbu - ma\right)$$
$$\dot{n} = \left(-\frac{m^2}{r} + qy^2 - 2mbu\right)$$

On comparing (4) and $h(t_0)$ we get:

$$p(t_0) = sc^2, \qquad m(t_0) = -scy(t_0), \qquad n(t_0) = sy(t_0)^2$$

f) Define: $\hat{x} = -p^{-1}m = -\frac{m}{p}$

Therefore, we have: $\dot{\hat{x}} = (\dot{p}m - \dot{m}p)/p^2$

Substituting the values of $\dot{p}$ and $\dot{m}$ from part e, we get:

$$\dot{\hat{x}} = \frac{\left(-\frac{p^2}{r} + qc^2 - 2pa\right)m}{p^2} - \frac{\left(-\frac{pm}{r} - qcy - pbu - ma\right)}{p}$$

$$\dot{\hat{x}} = a\left(-\frac{m}{p}\right) + bu - \left(\frac{qc}{p}\right)\left(c\left(-\frac{m}{p}\right) - y\right)$$

This gives us:

$$l = \frac{qc}{p}$$

g) We have the following differential equation for *p*:

$$\dot{p} = \left(-\frac{p^2}{r} + qc^2 - 2pa\right)$$

At steady state, $\dot{p} = 0$.
Therefore we get the following quadratic in *p*:

$$\frac{p^2}{r} + 2ap - qc^2 = 0$$

We can solve this for *p* and hence obtain l.

h) Rewriting the above equation in terms of $\bar{p} = p^{-1}$:

$$\frac{1}{\bar{p}^2 r} + \frac{2a}{\bar{p}} - qc^2 = 0$$

Rearranging the terms:

$$\bar{p}^2 - \frac{2a}{qc^2}\bar{p} - \frac{1}{qc^2 r} = 0$$

You can compare this to the continuous algebraic Riccati equation that corresponds to LQR:

$$p^2 - \frac{2ar}{b^2}p - \frac{qr}{b^2} = 0$$

l can be expressed in terms of $\bar{p}$:

$$l = \frac{qc}{p} = qc\bar{p}$$

Solving for $\bar{p}$ we get:

$$\bar{p} = \frac{a}{qc^2} \pm \frac{1}{c^2}\sqrt{\frac{a^2}{q^2} + \frac{c^2}{qr}}$$

$$l = qc\bar{p} = \frac{a}{c} \pm \sqrt{\frac{a^2}{c^2} + \left(\frac{q}{r}\right)}$$

i) As $(q/r) \rightarrow 0$, we get $l = 0,\ 2a/c$. For $l = 0$, the observer would be really slow in catching up with the true states. You can check this by experimenting with the values in problems 1-3. $l = \frac{2a}{c}$ should give an optimal observer.

j) As $(q/r) \rightarrow \infty$, $l = \pm\infty$. Theoretically the observer should drive the estimated states instantly to the true states.

5. The code can be completed in the following steps:

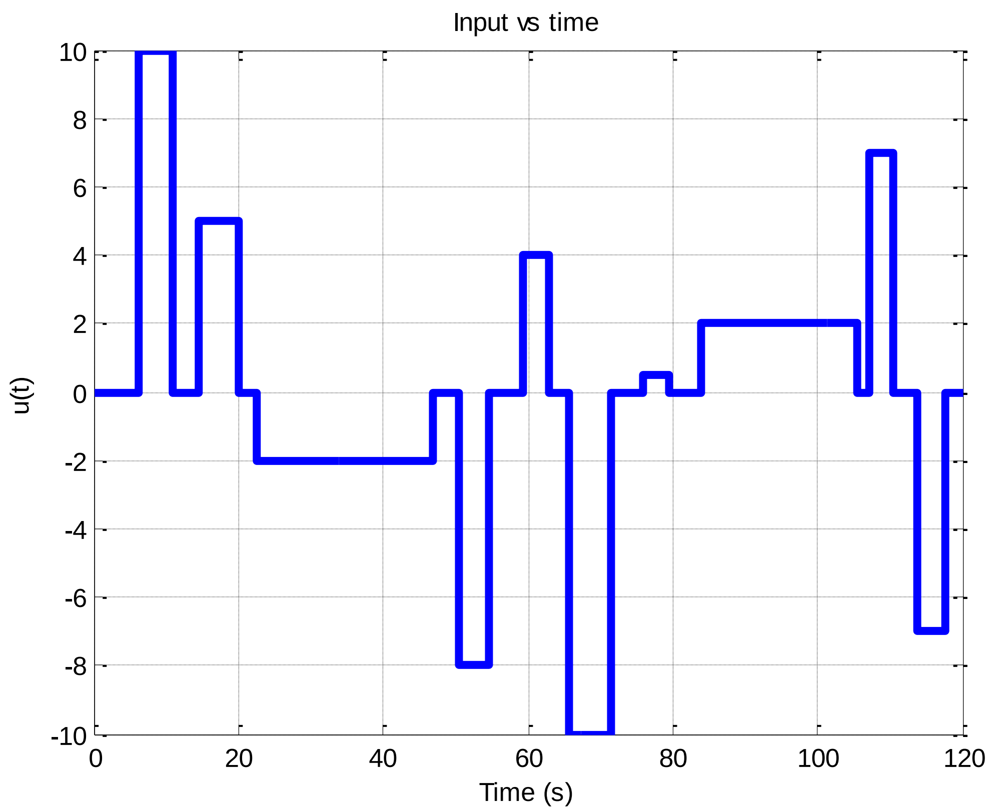
1) Read data file.
2) Initialize the matrices.
3) Find the F matrix.
4) Calculate $\dot{P}$.
5) Update the P matrix.
6) Calculate the observer L.
7) Calculate $\dot{\hat{x}}$.
8) Update $\hat{x}$.
9) Return to step 3.

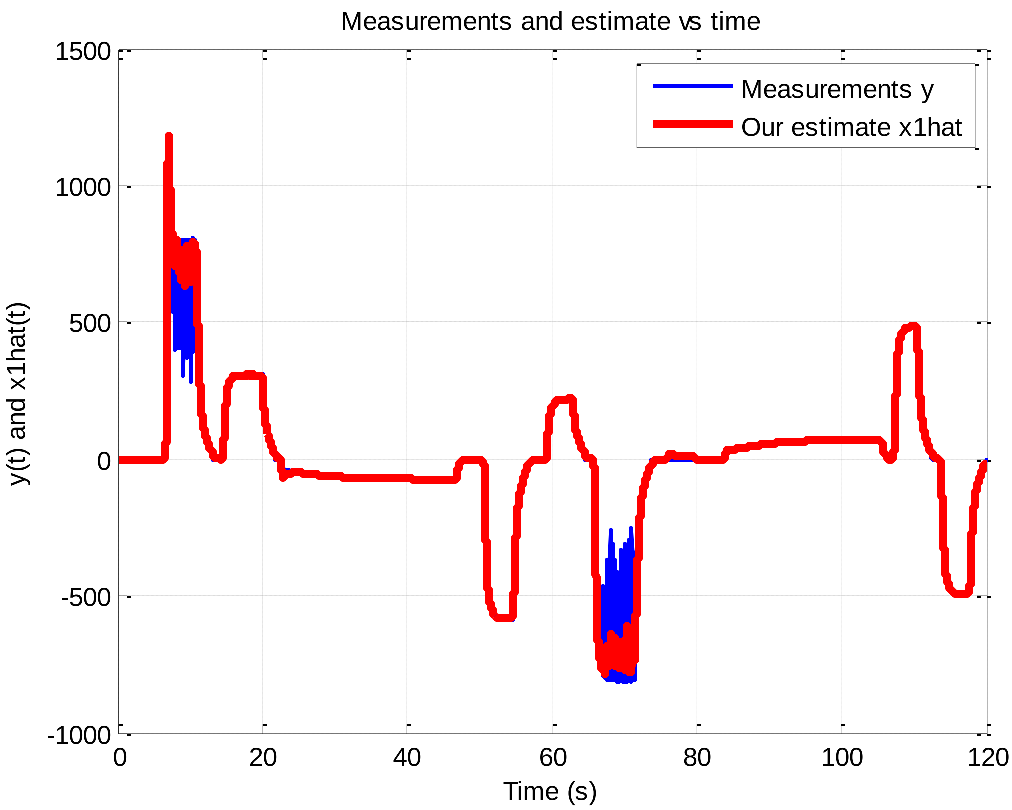
Here's the function hw6prob05:

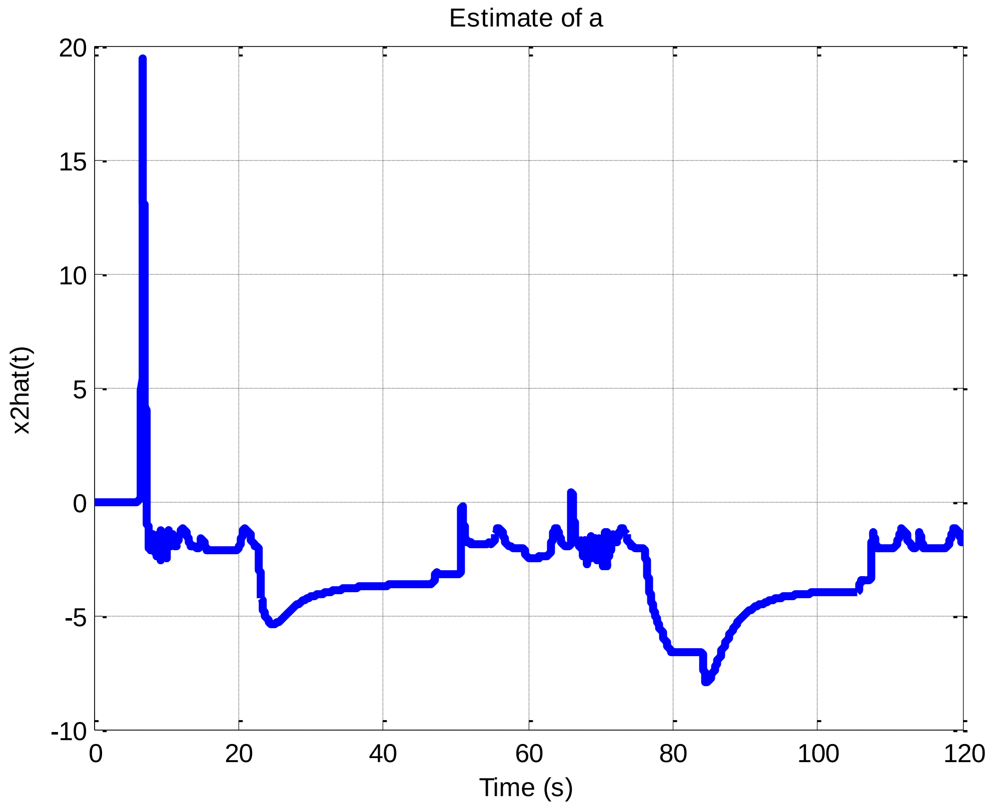
```
function hw6prob05

close all;
clear all;
clc;

% Name of file (change this line)
filename = 'datasec02.txt';

% Get data from file
stuff = importdata(filename);
data = stuff.data(:,[1 2 4]);
t = data(:,1);
y = data(:,2);
u = data(:,3);
dt = t(2)-t(1);

% Implement an observer for parameter identification...
% (your code here)
```

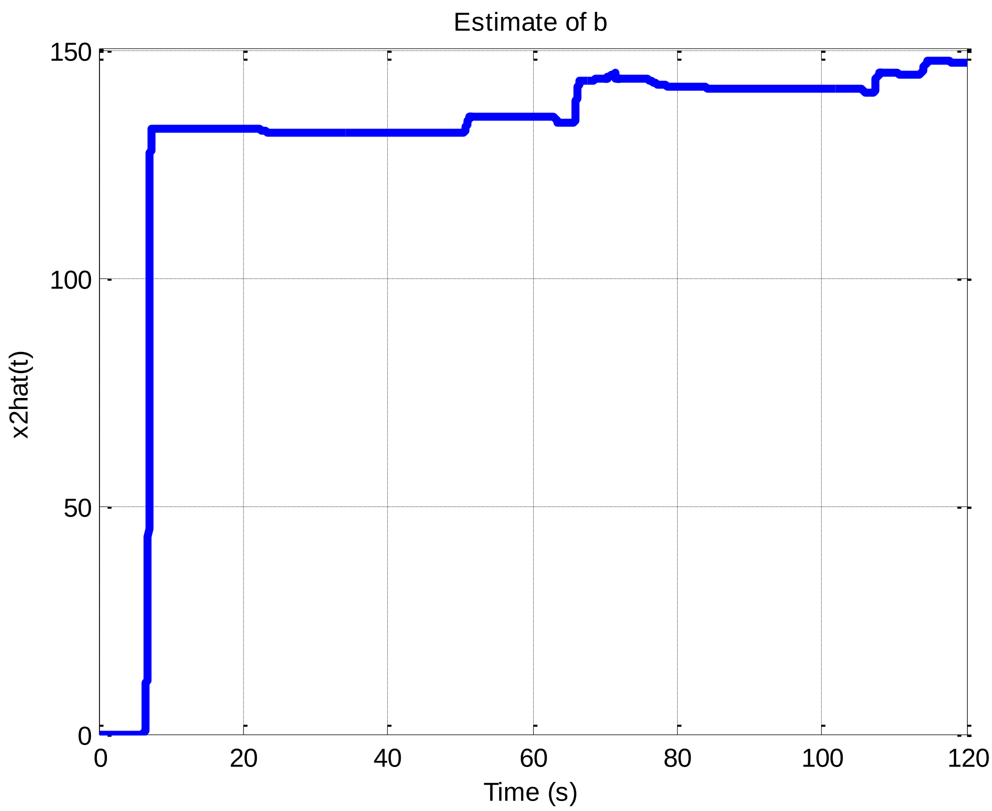
```
Ro = eye(3);    Qo = 0.01;
G = [1 0 0];
dT = t(2) - t(1);

xhat = zeros(3, length(t));

P = eye(3);

for i = 1:length(t)

   F = [xhat(2,i) xhat(1,i) u(i); 0 0 0; 0 0 0];

   Pdot = inv(Ro) + F*P + P*F' - P*G'*Qo*G*P;

   P = P + Pdot*dT;

   L = P*G'*Qo;

   xhatdot = [xhat(2,i)*xhat(1,i) + xhat(3,i)*u(i); 0; 0] - L*(xhat(1,i)-
y(i));

   xhat(:,i+1) = xhat(:,i) + xhatdot*dT;

end

% Plot:
%  (1) u vs t
%  (2) y vs t and x1hat vs t on same plot
%  (3) x2hat vs t -- this is your estimate of "a"
%  (4) x3hat vs t -- this is your estimate of "b"
% (your code here)
figure()
plot(t,u);
grid on;

figure()
plot(t,y);
grid on;
hold on;
plot(t,xhat(1,1:end-1),'r');

figure()
plot(t,xhat(2,1:end-1));
grid on;

figure()
plot(t,xhat(3,1:end-1));
grid on;
```

Here are the plots:






Estimate of a
x2hat(t)
20
15
10
5
0
-5
-10
0
20
40
60
80
100
120
Time (s)


Estimate of b
x2hat(t)
150
100
50
0
0
20
40
60
80
100
120
Time (s)