```
close all
clear all
clc

% Question 1
%Part a)
syms p th pdot thdot
x = [p;th;pdot;thdot];
load pendulum.mat %stores A and B matrices as A and B
n = size(A,2);
B_tau = B(:,2);
CO = ctrb(A,B_tau);
rank(CO)
```

```
ans =

     3
```

```
lambda = eig(A)
```

```
lambda =

         0
    3.2591
   -3.3782
   -0.0908
```

```
for i = 1:n
    if real(lambda(i)) >= 0
        lam = lambda(i)
        ran = rank([A-lambda(i)*eye(n), B_tau])
    end
end
```

```
lam =

     0
```

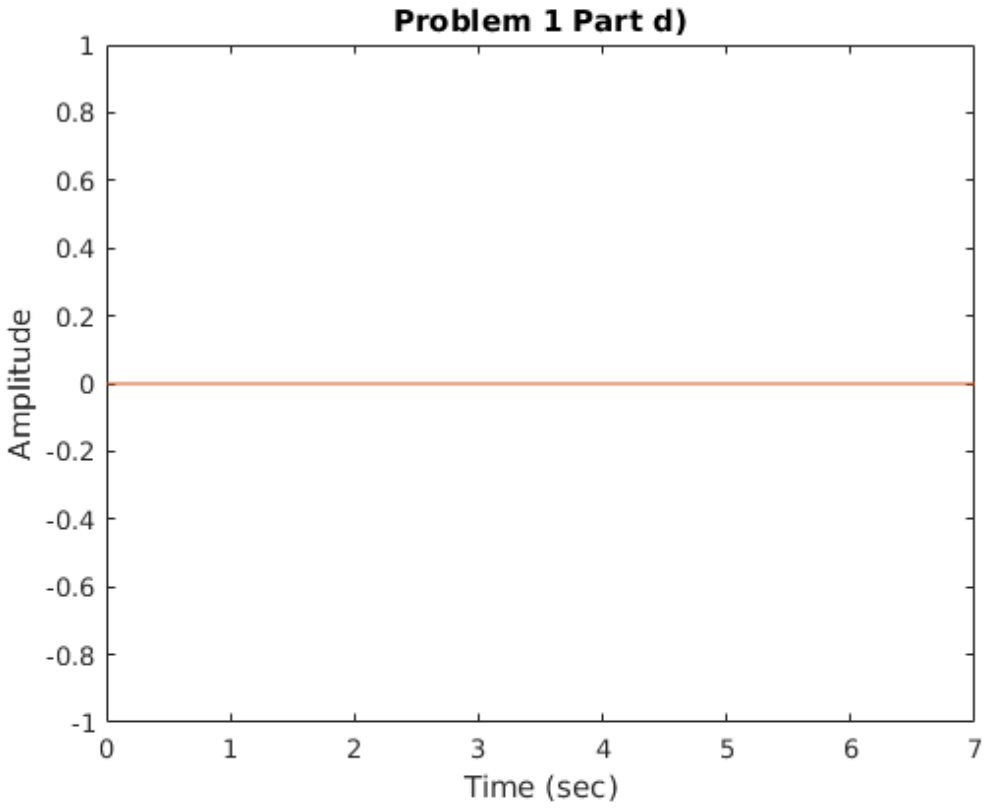
```
ran =

     3
```

```
lam =

    3.2591
```

```
ran =

     4


u1 = orth(CO);
u2 = null(CO.');
T = [u1, u2];
Abar = inv(T)*A*T;
Bbar = inv(T)*B_tau;
xbar = simplify(inv(T)*x);
Au = Abar(4,4);
xu = vpa(xbar(4),2)


xu =

0.09*p + 0.99*pdot - 0.089*th + 0.09*thdot


%-------------------------------------------------------------------------
%
%Part b)
clc
C_p = [1 0 0 0]; %I can measure p, but not theta
O = obsv(A,C_p); %compute observability matrix
rank(O)


ans =

     4


C_th = [0 1 0 0]; %I can measure theta, but not p
O = obsv(A,C_th); %compute observability matrix
rank(O)


ans =

     3


lambda = eig(A);
for i = 1:n
    lam = lambda(i)
    ran_th = rank([A-lambda(i)*eye(n); C_th])
    ran_p = rank([A-lambda(i)*eye(n); C_p])
end
```

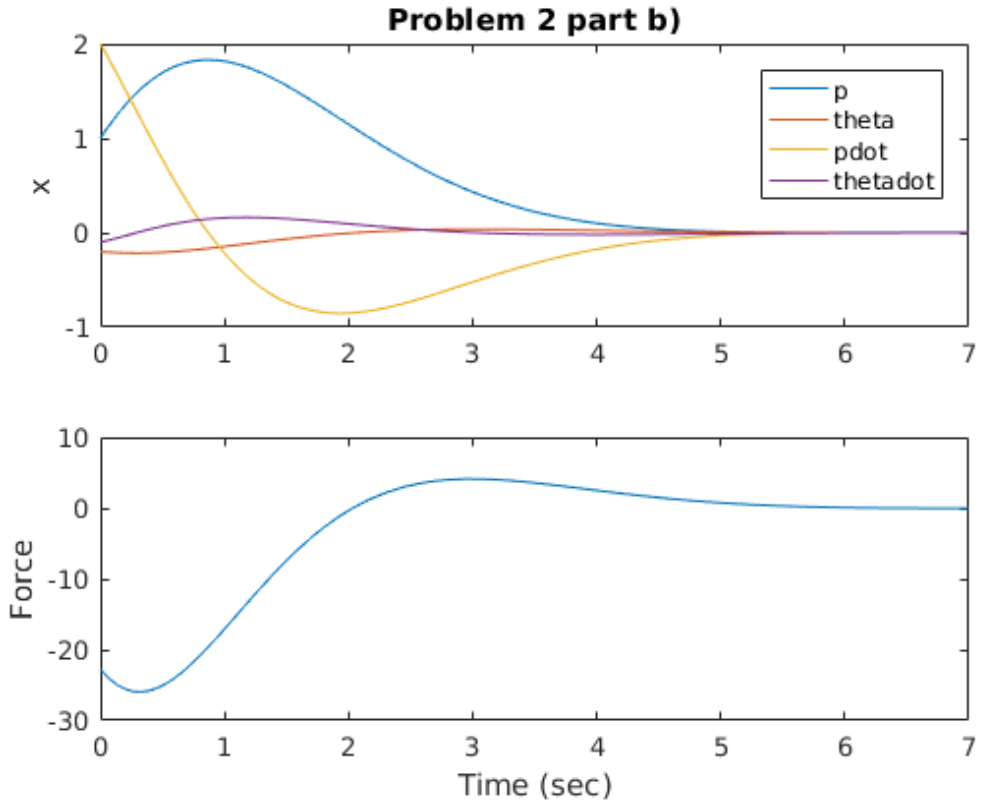
```
lam =

     0

ran_th =

     3

ran_p =

     4

lam =

    3.2591

ran_th =

     4

ran_p =

     4

lam =

   -3.3782

ran_th =

     4

ran_p =

     4

lam =

   -0.0908

ran_th =

     4
```

```
ran_p =

     4


%-------------------------------------------------------------------------
%
%Part c)
clc
C = [0 0 1 0; 0 1 0 0];
D = [0;0];
O = obsv(A,C);
rank(O)


ans =

     3


syms s
G_hat = simplify(C*inv(s*eye(4) - A)*B);
simplify(det(G_hat))


ans =

-100/(- 1000*s^3 - 210*s^2 + 10999*s + 1000)


%-------------------------------------------------------------------------
%
%Part d)
x0 = [0 0 0 0]';
tspan = [0 7];
[t, x] = ode45(@dynamics_free,tspan,x0, [], A);
figure()
plot(t,x(:,1))
hold on
plot(t,x(:,2))
title('Problem 1 Part d)')
xlabel('Time (sec)')
ylabel('Amplitude')
```




```
%-------------------------------------------------------------------------
%

% Question 2
%Part a)
B_F = B(:,1); %B matrix with only F as an input
P = [-1;-2;complex(-1,1);complex(-1,-1)]; %desired closed loop poles
K = place(A,B_F, P) %u = -K*x


K =

   -4.0000 -214.1000  -11.0040  -58.9040


eig(A-B_F*K) %check that closed loop poles are where they're suppose
 to be


ans =

  -2.0000 + 0.0000i
  -1.0000 + 1.0000i
  -1.0000 - 1.0000i
  -1.0000 + 0.0000i
```

```
%-------------------------------------------------------------------------
%
%Part b)
x0 = [1;-0.2;2;-0.1];
tspan = [0 7];
[t, x] = ode45(@dynamics,tspan,x0, [], A, B_F, K);
for i = 1:length(t)
    F(i) = -K*x(i,:)';
end

figure()
subplot(2,1,1)
plot(t,x(:,1))
hold on
plot(t,x(:,2))
plot(t,x(:,3))
plot(t,x(:,4))
ylabel('x')
legend('p','theta','pdot','thetadot')
title('Problem 2 part b)')
subplot(2,1,2)
plot(t,F)
xlabel('Time (sec)')
ylabel('Force')
```



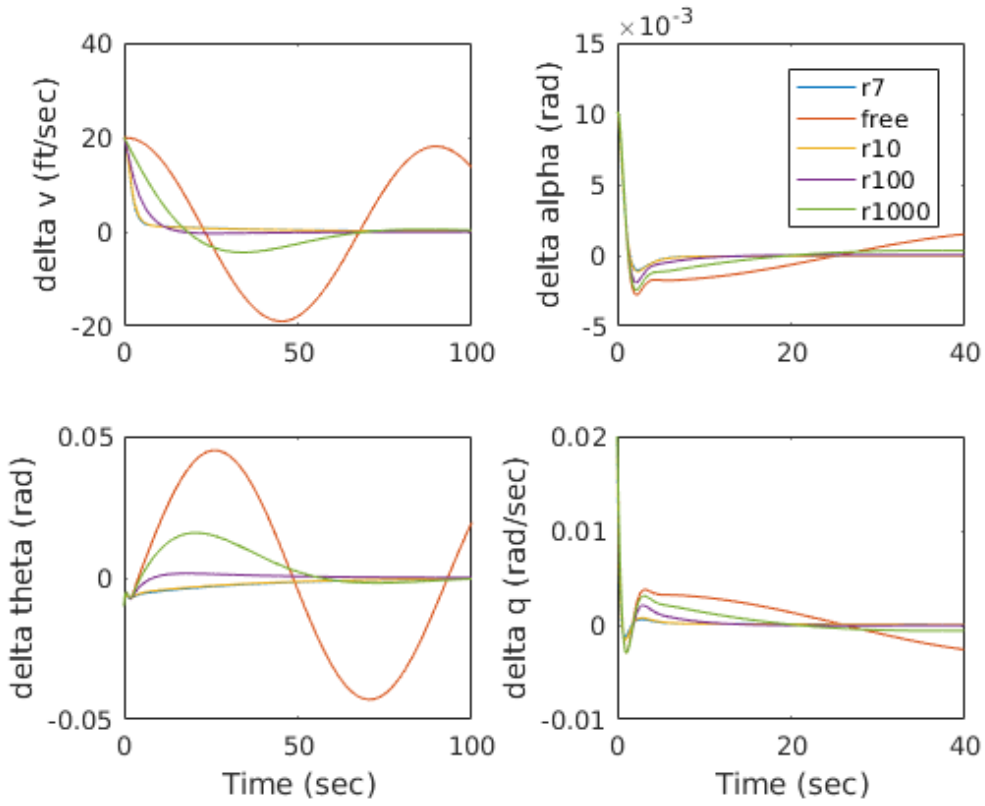

```
%-------------------------------------------------------------------------
%
```

```
%Part c)
sys = ss(A-B_F*K,B_F,eye(4),0)
figure()
sigma(sys)
%-------------------------------------------------------------------------
%

% Problem 4
clc
clear all
close all
load f16_long.mat %stores Along and Blong matrices
%Part a)
r = 7;
R1 = [1/(5^2), 0; 0, 1/((25*pi/180)^2)];
R = r*R1;
q = 1;
Q1 = [1/500^2, 0, 0, 0;...
    0, 1/(2.3*pi/180)^2, 0, 0;...
    0, 0, 1/(17.2*pi/180)^2, 0;...
    0, 0, 0, 1/(0.5*pi/180)^2];
Q = q*Q1;

K = lqr(Along,Blong,Q,R);

x0 = [20, 0.01, -0.01, 0.02]';
tspan = 0:0.1:100;
[t, x7] = ode45(@dynamics_pr4,tspan,x0, [], Along, Blong, K);
[t,xf] = ode45(@dynamics_pr4_free_resp,tspan,x0, [], Along, Blong, K);
r = 10;
R = r*R1;
K = lqr(Along,Blong,Q,R);
[t, x10] = ode45(@dynamics_pr4,tspan,x0, [], Along, Blong, K);
r = 100;
R = r*R1;
K = lqr(Along,Blong,Q,R);
[t, x100] = ode45(@dynamics_pr4,tspan,x0, [], Along, Blong, K);
r = 1000;
R = r*R1;
K = lqr(Along,Blong,Q,R);
[t, x1000] = ode45(@dynamics_pr4,tspan,x0, [], Along, Blong, K);

figure()
subplot(2,2,1)
plot(t,x7(:,1))
hold on
plot(t,xf(:,1))
plot(t,x10(:,1))
plot(t,x100(:,1))
plot(t,x1000(:,1))
ylabel('delta v (ft/sec)')

subplot(2,2,2)
plot(t(1:400),x7(1:400,2))
```

```
hold on
plot(t(1:400),xf(1:400,2))
plot(t(1:400),x10(1:400,2))
plot(t(1:400),x100(1:400,2))
plot(t(1:400),x1000(1:400,2))
legend('r7','free','r10','r100','r1000')
ylabel('delta alpha (rad)')

subplot(2,2,3)
plot(t,x7(:,3))
hold on
plot(t,xf(:,3))
plot(t,x10(:,3))
plot(t,x100(:,3))
plot(t,x1000(:,3))
ylabel('delta theta (rad)')
xlabel('Time (sec)')

subplot(2,2,4)
plot(t(1:400),x7(1:400,4))
hold on
plot(t(1:400),xf(1:400,4))
plot(t(1:400),x10(1:400,4))
plot(t(1:400),x100(1:400,4))
plot(t(1:400),x1000(1:400,4))
ylabel('delta q (rad/sec)')
xlabel('Time (sec)')
```

```
sys =

  A =
          x1      x2      x3      x4
   x1      0       0       1       0
   x2      0       0       0       1
   x3    0.4   20.41       1    5.99
   x4   -0.4  -10.41      -1      -6

  B =
        u1
   x1    0
   x2    0
   x3  0.1
   x4 -0.1

  C =
       x1  x2  x3  x4
   y1   1   0   0   0
   y2   0   1   0   0
   y3   0   0   1   0
   y4   0   0   0   1

  D =
       u1
```

```
y1   0
y2   0
y3   0
y4   0

Continuous-time state-space model.
```



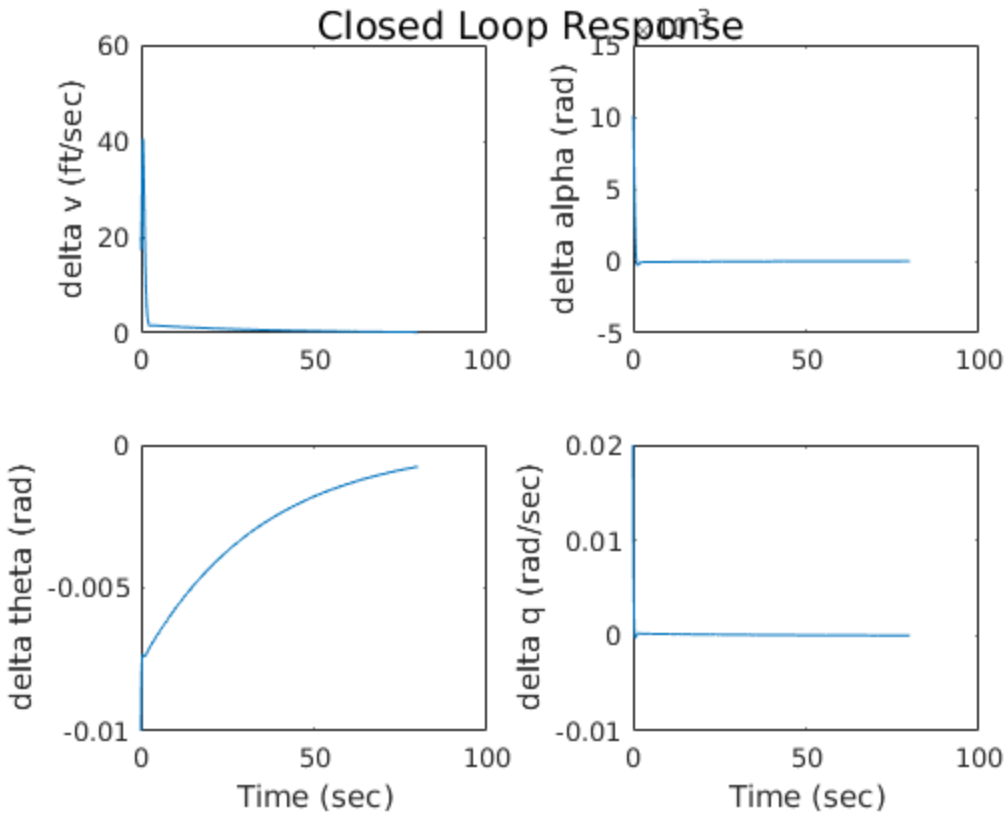

```
%-------------------------------------------------------------------------
%
%Part b)
x0 = [20, 0.01, -0.01, 0.02]';
tspan = 0:0.1:80;
tspanf = 0:0.5:4000;
R = [0.001325, 0; 0, 11.6];
K = lqr(Along,Blong,Q,R);
[t, x] = ode45(@dynamics_pr4,tspan,x0, [], Along, Blong, K);
[tfr,xf] = ode45(@dynamics_pr4_free_resp,tspanf,x0,[],Along,Blong,K);
for i = 1:length(t)
   u(:,i) = -K*x(i,:)';
end
u(2,:) = u(2,:)*180/pi; %convert to degrees

max(abs(min(u(1,:))),max(u(1,:)))


ans =
```

```
    4.9969
```

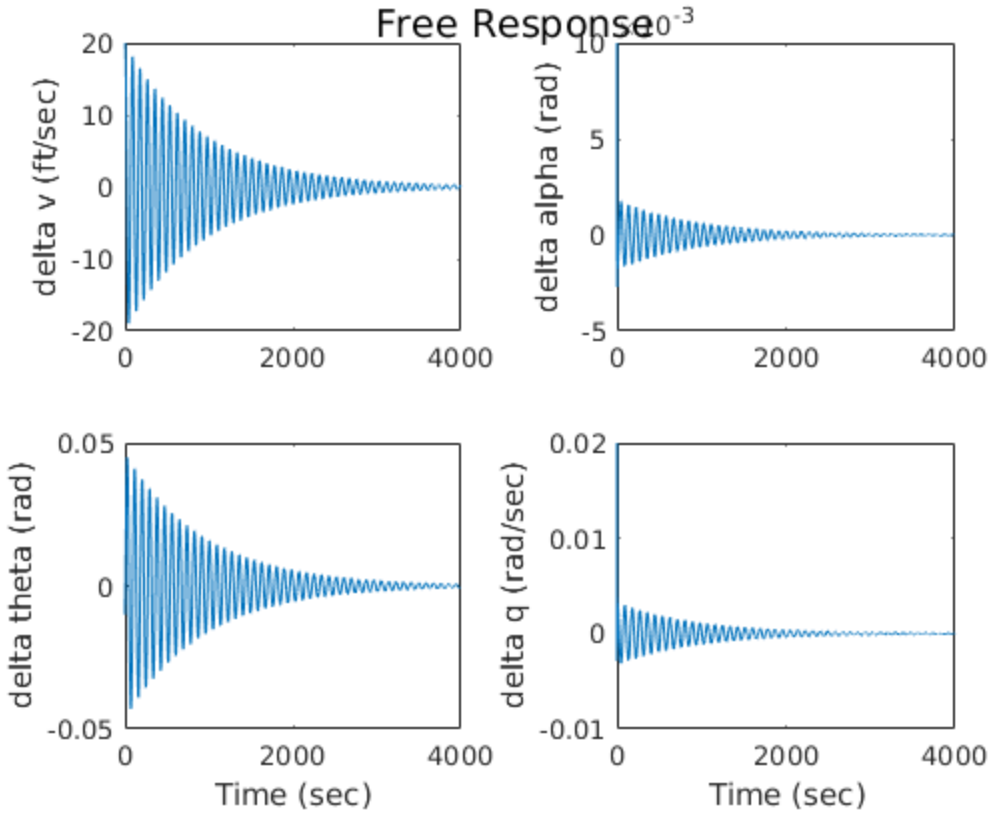
```
max(abs(min(u(2,:))),max(u(2,:)))
```

```
ans =

   24.9939
```

```
figure()
suptitle('Closed Loop Response')
subplot(2,2,1)
plot(t,x(:,1))
ylabel('delta v (ft/sec)')

subplot(2,2,2)
plot(t,x(:,2))
ylabel('delta alpha (rad)')

subplot(2,2,3)
plot(t,x(:,3))
ylabel('delta theta (rad)')
xlabel('Time (sec)')

subplot(2,2,4)
plot(t,x(:,4))
ylabel('delta q (rad/sec)')
xlabel('Time (sec)')
```

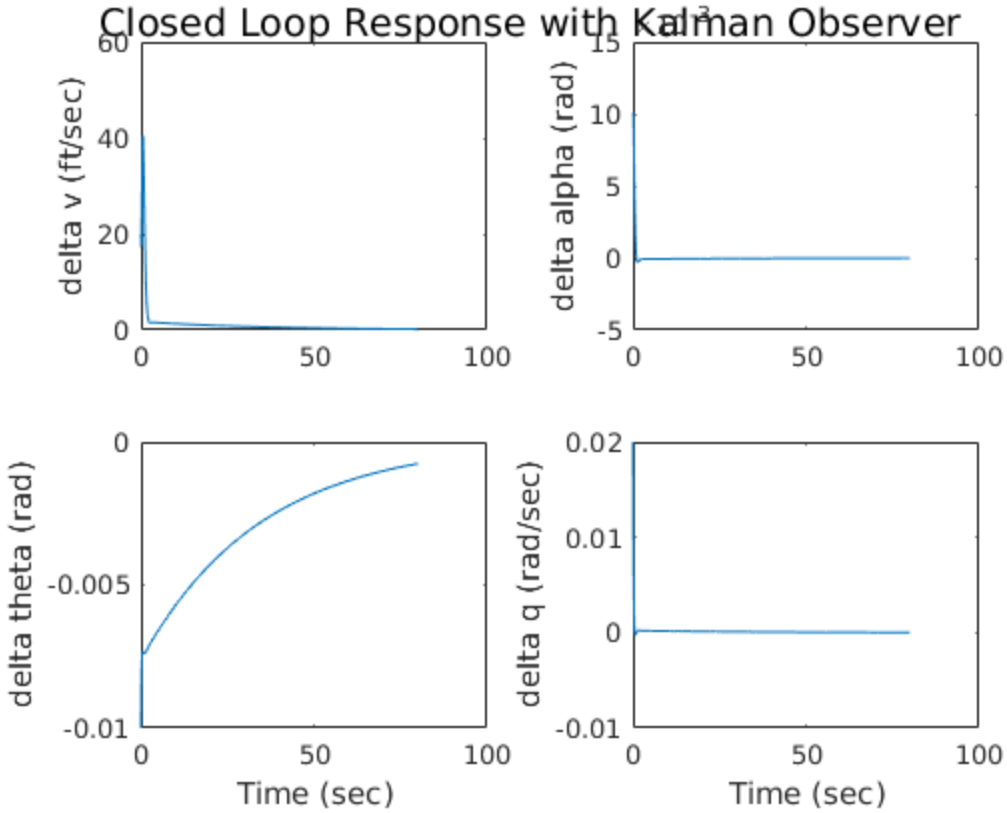



```
figure()
suptitle('Free Response')
subplot(2,2,1)
plot(tfr,xf(:,1))
ylabel('delta v (ft/sec)')

subplot(2,2,2)
plot(tfr,xf(:,2))
ylabel('delta alpha (rad)')

subplot(2,2,3)
plot(tfr,xf(:,3))
ylabel('delta theta (rad)')
xlabel('Time (sec)')

subplot(2,2,4)
plot(tfr,xf(:,4))
ylabel('delta q (rad/sec)')
xlabel('Time (sec)')
```




```
%-------------------------------------------------------------------------
%
%Part c)
C = [1 0 0 0; 0 -1 1 0];
R = diag([1,10^-5]');
D = (10^-4)*eye(2);
Q = Blong*D*Blong';
L = lqe(Along,eye(4),C,Q,R)


L =

    0.2450   -2.9982
   -0.0000    0.0110
   -0.0001    0.1702
    0.0000   -0.0026


xh0 = [0;0;0;0];
states0 = [x0',xh0']';
[t, states]=ode45(@dynamics_pr4_kalman,tspan,states0,
[],Along,Blong,C,K,L);
x = states(:,1:4);
xh = states(:,5:8);

figure()
suptitle('Closed Loop Response with Kalman Observer')
```

```
subplot(2,2,1)
plot(t,x(:,1))
ylabel('delta v (ft/sec)')

subplot(2,2,2)
plot(t,x(:,2))
ylabel('delta alpha (rad)')

subplot(2,2,3)
plot(t,x(:,3))
ylabel('delta theta (rad)')
xlabel('Time (sec)')

subplot(2,2,4)
plot(t,x(:,4))
ylabel('delta q (rad/sec)')
xlabel('Time (sec)')
```




```
%-------------------------------------------------------------------------
%
%Part d)
Aa = [Along Blong; C zeros(2)];
temp= inv(Aa)*[0;0;0;0;1;1];
F=temp(1);
N=temp(2);
```

```
s = tf('s');
L_hat = K*inv(s-Along)*Blong;
Gp_hat = C*inv(s*eye(4)-Along)*Blong+zeros(2);

CLTF = Gp_hat*inv(1+L_hat)*(N+K*F);

r=[5;5];

%t = 0:0.01:50;
%figure()
%step(r*CLTF,t)

% Dynamics Functions
function xdot = dynamics(t, x, A, B, K)
    xdot = (A-B*K)*x;
end

function xdot = dynamics_free(t,x,A)
    xdot = A*x;
end

function xdot = dynamics_pr4(t,x,A,B,K)
    xdot = (A-B*K)*x;
end

function xdot = dynamics_pr4_free_resp(t,x,A,B,K)
    xdot = A*x;
end

function out = dynamics_pr4_kalman(t,states,A,B,C,K,L)
    x = states(1:4);
    xh = states(5:8);
    xhdot = (A-L*C-B*K)*xh + L*C*x;
    xdot = (A-B*K)*x;
    out = [xdot;xhdot];
end
```

Published with MATLAB® R2016b