ECE 463/663 - Homework #9

Calculus of Variations. LQG Control. Due Wednesday, April 4th, 2022

Soap Film

1) Calculate the shape of a soap film connecting two rings around the X axis:

- Y(0) = 10
- Y(4) = 9

From the lecture notes, a soap film minimizes the surface area. The corresponding funtional is

$$J = \int \left(y\sqrt{1+y'^2} \right) dx$$

which has the solution

$$y = a \cdot \cosh\left(\frac{x-b}{a}\right)$$

Plugging in the two endpoints to solve for a and b

$$10 = a \cdot \cosh\left(\frac{-b}{a}\right)$$

$$9 = a \cdot \cosh\left(\frac{4-b}{a}\right)$$

Solving in Matlab, first create a cost function

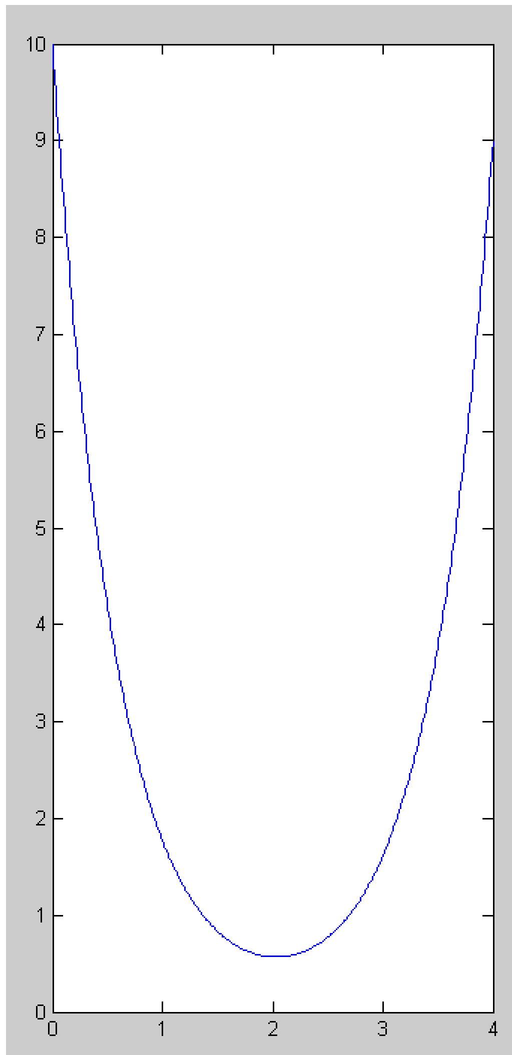
```
function [J] = soap(z)
   a = z(1);
   b = z(2);
   e1 = a*cosh(-b/a) - 10;
   e2 = a*cosh((4-b)/a) - 9;
   J = e1^2 + e2^2;
   end
```

Solve using fminsearch:

```
>> [z,e] = fminsearch('soap',[1,2])
              a          b
z =      0.5710    2.0301
e =   1.7898e-006
```

meaning

$$y = 0.5710 \cdot \cosh\left(\frac{x-2.0301}{0.5710}\right)$$



2) Calculate the shape of a soap film connecting two rings around the X axis:

- Y(0) = 10
- Y(2) = free

From the lecture notes,

$$y = a \cdot \cosh\left(\frac{x-b}{a}\right)$$

The endpoint constraint is

$$10 = a \cdot \cosh\left(\frac{-b}{a}\right)$$

The right endpoint constraint is

$$y' = -\sinh\left(\frac{2-b}{a}\right) = 0$$

Setting up a cost funiton in Matlab

```
function [J] = soap(z)
   a = z(1);
   b = z(2);
   e1 = a*cosh(-b/a) - 10;
   e2 = sinh((2-b)/a);
   J = e1^2 + e2^2;
   end
```

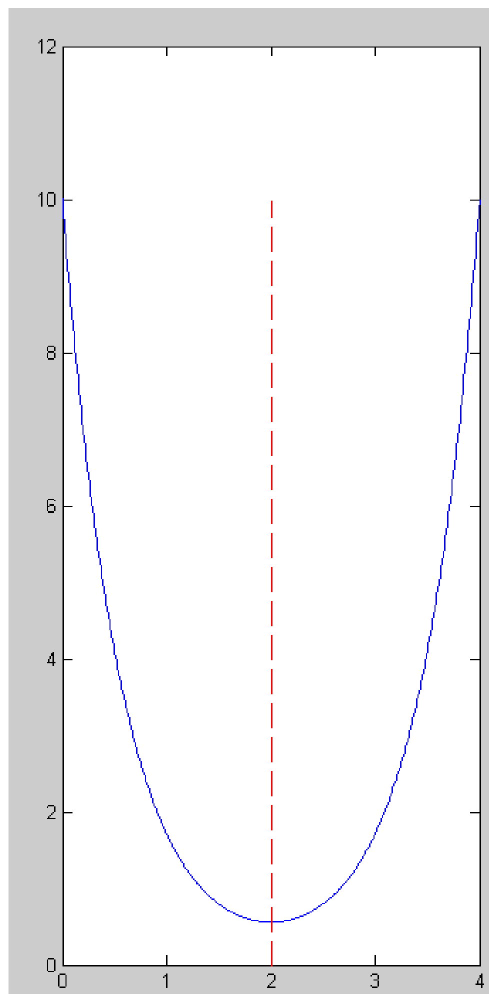
Solving

```
>> [z,e] = fminsearch('soap',[1,2])

            a            b
z =     0.5593     2.0000
e =   5.6278e-009
```

so

$$y = 0.5593 \cdot \cosh\left(\frac{x-2}{0.5593}\right)$$

```
>> a = z(1);
>> b = z(2);
>> y = a * cosh( (x-b)/a );
>> plot(x,y);
>> plot(x,y,[2,2],[0,10],'r--');
```



Hanging Chain

3) Calculate the shape of a hanging chain subject to the following constraints

- Length of chain = 12 meters
- Left Endpoint: (0,0)
- Right Endpoint: (10,1)

From the lecture notes, a hanging chain

- Minimes the potential energy,
- With the constraint that the total lenngth is 12 meters

The corresponding funcitonal is

$$F = x\sqrt{1+y'^2} + M\sqrt{1+y'^2}$$

which results in the solution

$$y = a \cdot \cosh\left(\frac{x-b}{a}\right) - M$$

$$\left(a \cdot \sinh\left(\frac{x-b}{a}\right)\right)_0^{10} = 12$$

Set up a cost function

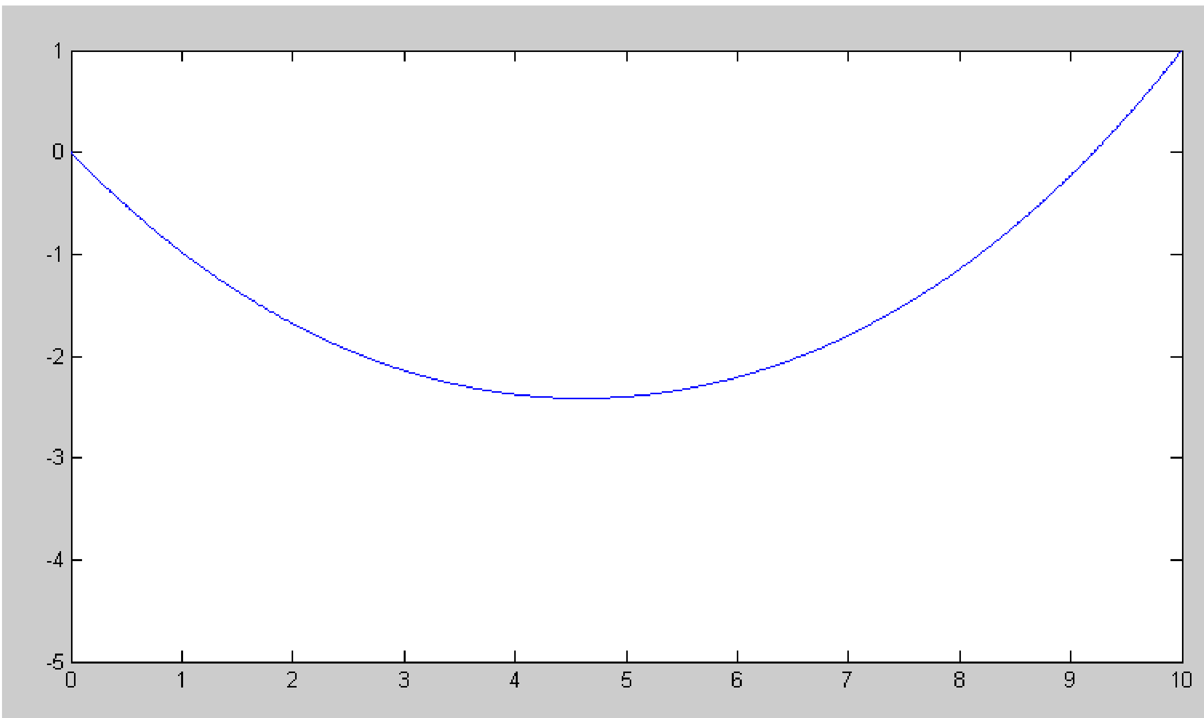
```
function J = chain(z)
a = z(1);
b = z(2);
M = z(3);

Length = 12;
x1 = 0;
y1 = 0;

x2 = 10;
y2 = 1;

e1 = a*cosh((x1-b)/a) - M - y1;
e2 = a*cosh((x2-b)/a) - M - y2;
e3 = a*sinh((x2-b)/a) - a*sinh((x1-b)/a) - Length;

J = e1^2 + e2^2 + e3^2;

end
```

Solve

```
>> [z,e] = fminsearch('chain',[1,2,3])

          a          b          M
z =     4.7425    4.6039    7.1583

e =   2.9425e-009
```

plotting the shape

```
>> a = z(1);
>> b = z(2);
>> M = z(3);
>> x = [0:0.01:10]';
>> y = a*cosh( (x-b)/a ) - M;
>> plot(x,y);
>> ylim([-5,1])
```



Ricatti Equation

4) Find the function, x(t), which minimizes the following funcional

$$J = \int_0^{10} (x^2 + 4\dot{x}^2)dt$$

x(0) = 6

x(10) = 4

Any funciton that minimizes this functional must minimize the Euler LaGrange equation

$$F_x - \frac{d}{dt}(F_{x'}) = 0$$

Solving

$$2x - \frac{d}{dt}(8\dot{x}) = 0$$

$$4\ddot{x} + x = 0$$

$$(4s^2 + 1)x = 0$$

Either

- x = 0, or
- s = {+0.5, -0.5}

going with the latter solution

$$x(t) = ae^{0.5t} + be^{-0.5t}$$

Plugging in the endpoint constraints

$$x(0) = 6 = a + b$$

$$x(10) = 4 = ae^5 + be^{-5}$$

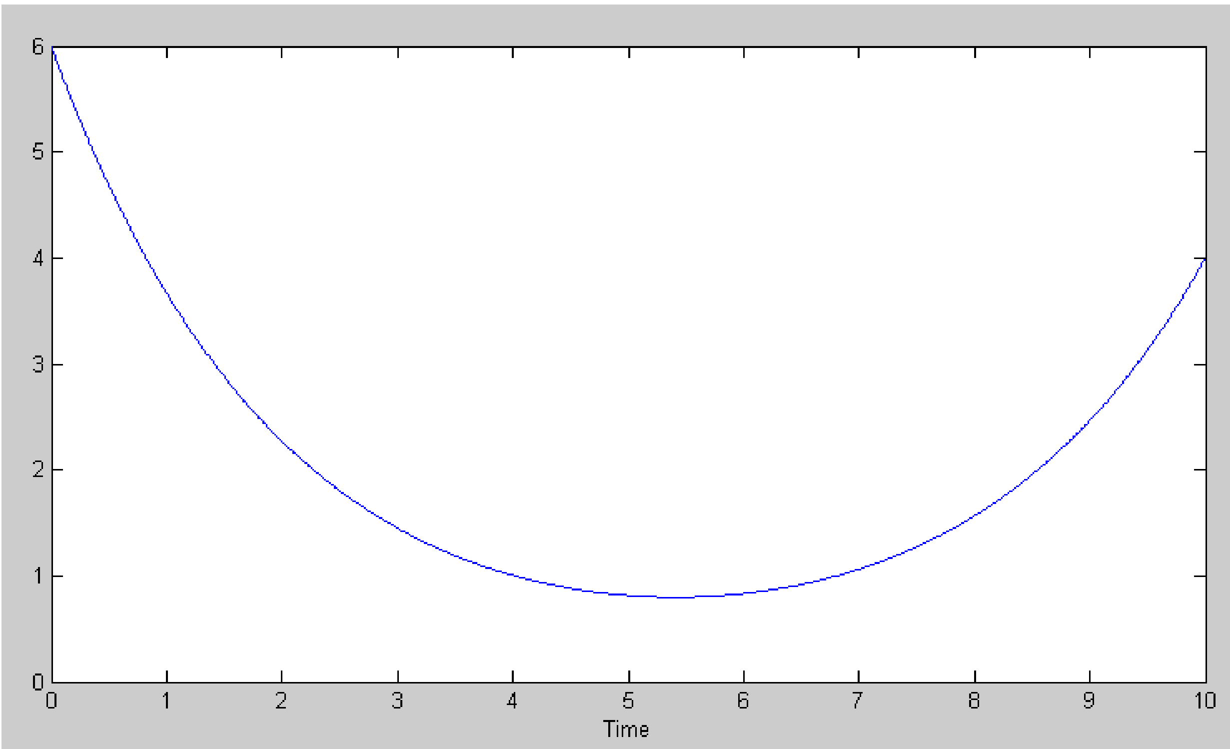
Solving 2 equations for 2 unknowns

```
>> A = [1,1 ; exp(5),exp(-5)]

    1.0000     1.0000
  148.4132     0.0067

>> B = [6;4]

     6
     4

>> ab = inv(A)*B

a    0.0267
b    5.9733
```

$$x(t) = 0.0267e^{0.5t} + 5.9733e^{-0.5t}$$

Plotting x(t)

```
>> a = ab(1);
>> b = ab(2);
>> t = [0:0.01:10]';
>> x = a*exp(t/2) + b*exp(-t/2);
>> plot(t,x);
>> xlabel('Time');
```



5) Find the function, x(t), which minimizes the following funcional

$$J = \int_0^{10} (x^2 + 9u^2)dt$$

$$\dot{x} = -0.1x + u$$

x(0) = 6

x(10) = 4

The functional for this problem (including a LaGrange multiplier) is

$$F = (x^2 + 9u^2) + m(\dot{x} + 0.1x - u)$$

Solving three Euler LaGrange equations

$$F_x - \frac{d}{dt}(F_{x'}) = 0$$

$$(2x + 0.1m) - \frac{d}{dt}(m) = 0$$

(1) $$2x + 0.1m - \dot{m} = 0$$

$$F_u - \frac{d}{dt}(F_{u'}) = 0$$

(2) $$18u - m = 0$$

$$F_m - \frac{d}{dt}(F_{m'}) = 0$$

(3) $$\dot{x} + 0.1x - u = 0$$

Substituting

$$m = 18u$$

$$2x + 1.8u - 18\dot{u} = 0$$

$$u = \dot{x} + 0.1x$$

$$\dot{u} = \ddot{x} + 0.1\dot{x}$$

$$2x + 1.8(\dot{x} + 0.1x) - 18(\ddot{x} + 0.1\dot{x}) = 0$$

Simplifying

$$-18\ddot{x} + 2.18x = 0$$

$$(-18s^2 + 2.18)x = 0$$

meaning

s = {0.3480, -0.3480}

and

$$x(t) = a \cdot e^{0.3480t} + b \cdot e^{-0.3480t}$$

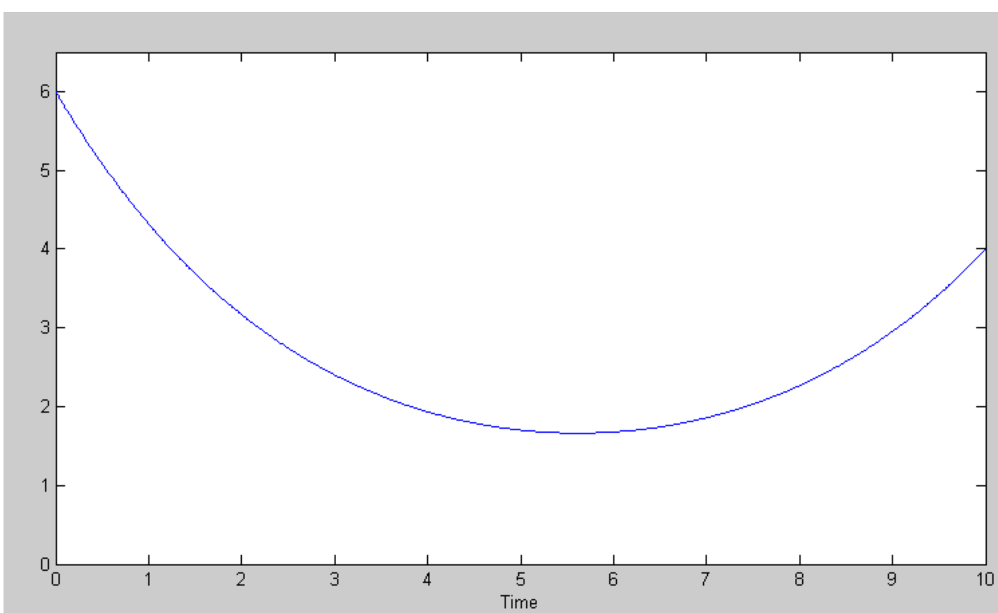
Plugging in the endpoints

$$x(0) = 6 = a + b$$

$$x(10) = 4 = a \cdot e^{3.480} + b \cdot e^{-3.480}$$

Solving

```
>> s = roots([-18,0,2.18])

    0.3480
   -0.3480

>> A = [1,1 ; exp(s(1)*10),exp(s(2)*10)]

    1.0000    1.0000
   32.4630    0.0308

>> B = [6;4];
>> ab = inv(A)*B

    0.1176
    5.8824

>> t = [0:0.01:10]';
>> a = ab(1);
>> b = ab(2);
>> x = a*exp(s1*t) + b*exp(s2*t);
>> plot(t,x);
>> xlabel('Time');
```



LQG Control

6) Cart & Pendulum (HW #4 & HW#6):

$$s\begin{bmatrix} x \\ \theta \\ sx \\ s\theta \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -14.7 & 0 & 0 \\ 0 & 24.5 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \theta \\ sx \\ s\theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ -0.5 \end{bmatrix}F$$

Design a full-state feedback control law of the form

$$F = U = K_r R - K_x X$$

for the cart and pendulum system from homework #4 using LQG control so that

- The DC gain is 1.00
- The 2% settling time is 8 seconds, and
- There is less than 10% overshoot for a step input.

```
>> A = [0,0,1,0 ; 0,0,0,1 ; 0,-14.7,0,0 ; 0,24.5,0,0 ];
>> B = [0;0;0.5;-0.5];
>> C = [1,0,0,0];
>> D = 0;
```

Guess #1: $J = x^2 + u^2$

```
>> Kx = lqr(A, B, diag([1,0,0,0]), 1);
>> eig(A - B*Kx)

  -4.9496 + 0.0303i
  -4.9496 - 0.0303i
  -0.3182 + 0.3143i
  -0.3182 - 0.3143i
```

Too slow: Increase the weighting on x:

```
>> Kx = lqr(A, B, diag([10,0,0,0]), 1);
>> eig(A - B*Kx)

  -4.9482 + 0.0959i
  -4.9482 - 0.0959i
  -0.5732 + 0.5514i
  -0.5732 - 0.5514i
```

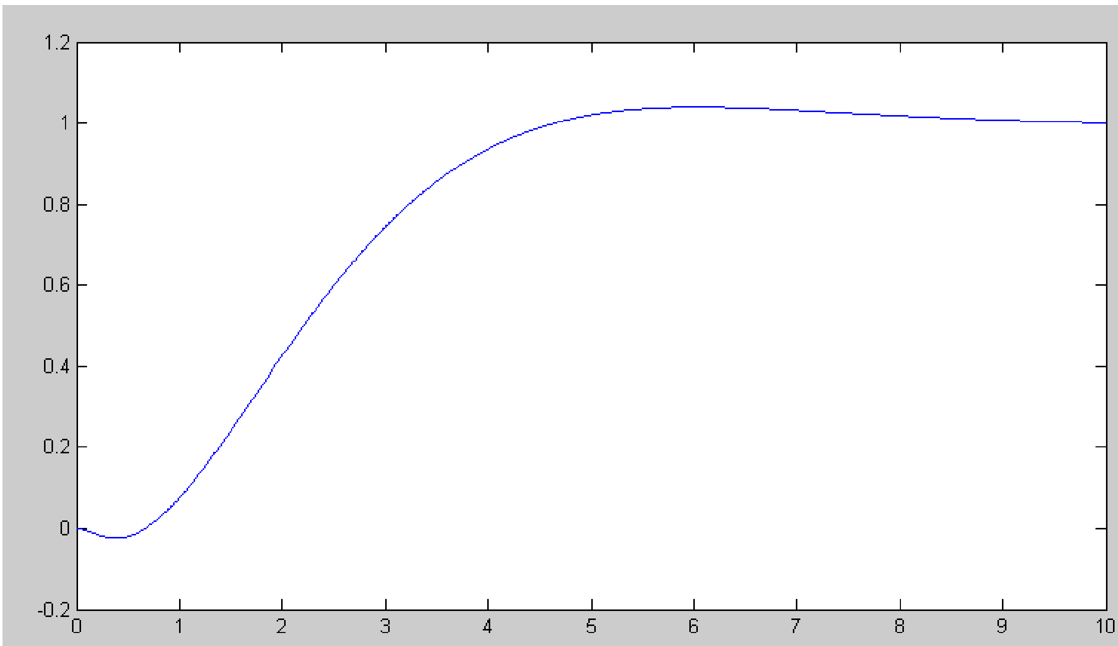
Good enough.

```
>> DC = -C*inv(A - B*Kx)*B

DC =   -0.3162

>> Kr = 1/DC
```

```
Kr =   -3.1623
```

Plotting the step response

```
>> G = ss(A-B*Kx, B*Kr, C, D);
>> t = [0:0.01:10]';
>> y = step(G,t);
>> plot(t,y);
```



Compare your results with homework #6

- Where are the closed-loop poles with pole placement and with LQG control?
- Are the feedback gains larger or smaller with LQG control?
- Which one works better?

```
Kx6 = ppl(A, B, [-0.5+j*0.6, -0.5-j*0.6, -2, -3])

ppl:     Kx6 =   -0.7469  -72.9669   -1.8469  -13.8469
lqr:     Kx  =   -3.1623 -125.1059   -7.0083  -29.0939
```

Closed-Loop Poles

```
>> [eig(A-B*Kx), eig(A-B*Kx6)]

  -4.9482 + 0.0959i  -3.0000
  -4.9482 - 0.0959i  -2.0000
  -0.5732 + 0.5514i  -0.5000 + 0.6000i
  -0.5732 - 0.5514i  -0.5000 - 0.6000i
```

7) Ball and Beam (HW #4 & HW#6):

$$s\begin{bmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -7 & 0 & 0 \\ -4.2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.143 \end{bmatrix} T$$

Design a full-state feedback control law of the form

$$T = U = K_r R - K_x X$$

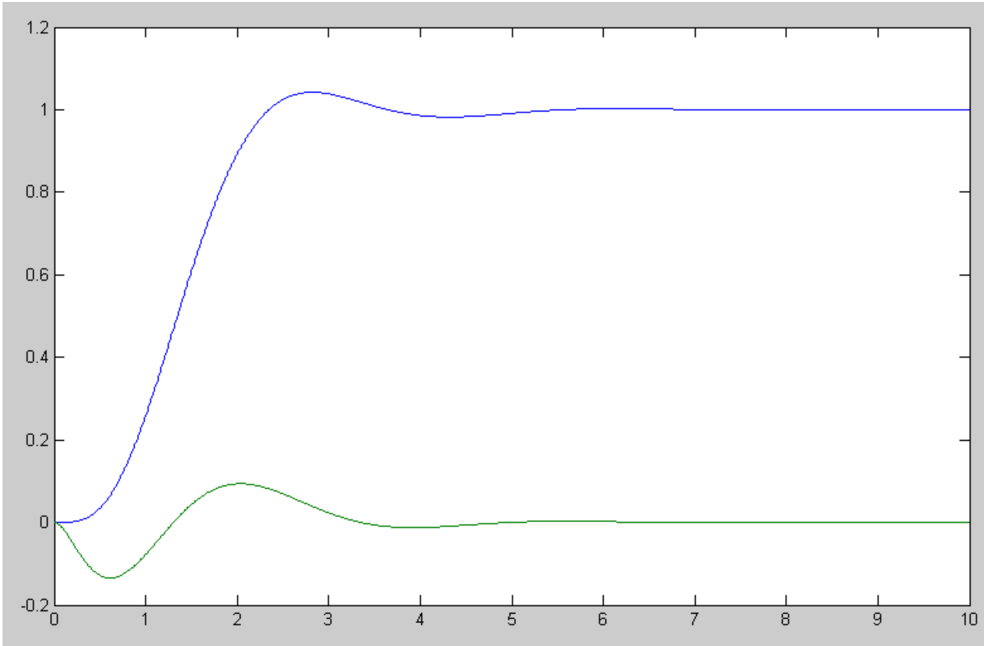
for the ball and beam system from homework #4 using LQG control so that

- The DC gain is 1.00
- The 2% settling time is 8 seconds, and
- There is less than 10% overshoot for a step input.

Use trial and error - adjusting the gains of Q until

- Start with Q = diag([1 0 0 0])
- Reduce the oscillations: Q = diag([1 0 200 0])
- Reduce the osicllations some more; Q = diag([1 0 200 1000])

```
Q = diag([1,0,200,1000]);
R = 1;

A = [0,0,1,0 ; 0,0,0,1 ; 0,-7,0,0 ; -4.2,0,0,0];
B = [0;0;0;0.143];
C = [1,0,0,0];
D = [0];

Kx = lqr(A, B, Q, R);
DC = -C*inv(A-B*Kx)*B;
Kr = 1/DC;

C = [1,0,0,0 ; 0,1,0,0];
D = [0;0];
G = ss(A-B*Kx, B*Kr, C, D);
t = [0:0.01:10]';
y = step(G,t);
plot(t,y);
```



Compare your results with homework #6

- Where are the closed-loop poles with pole placement and with LQG control?
- Are the feedback gains larger or smaller with LQG control?
- Which one works better?

```
>> [eig(A-B*Kx), eig(A-B*Kx6)]

     LQR                  PPL
  -4.8246              -4.0000
  -0.9248 + 1.8249i   -3.0000
  -0.9248 - 1.8249i   -0.5000 + 0.6000i
  -1.4568              -0.5000 - 0.6000i

>> [Kx6 ; Kx]

PPL   -36.6833  137.1329  -16.2537   55.9441
LQR   -58.7583  159.6637  -39.2506   56.8600
```

>>

The results are about the same.

- Pole placement is nice: you have complete control over where the poles go
- LQR is nice: you have knobs you can adjust to get the response you want