Cart and Pendulum System

Pole Placement Example

NDSU ECE 463/663

Lecture #15

Inst: Jake Glower

Please visit Bison Academy for corresponding
lecture notes, homework sets, and solutions

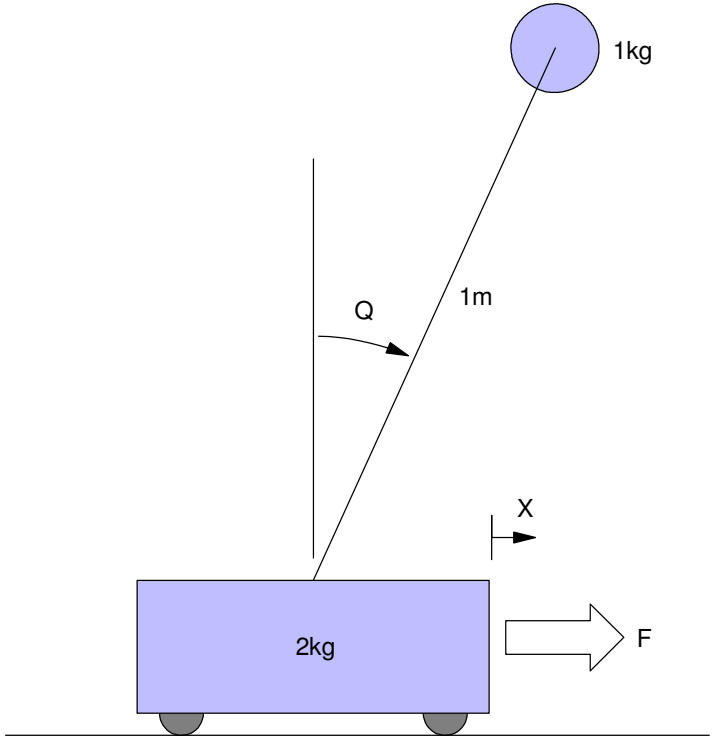
Cart and Pendulum System:

Nonlinear Model:

$$\begin{bmatrix} 3 & \cos\theta \\ \cos\theta & 1 \end{bmatrix}\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta}^2 \sin\theta \\ g\sin\theta \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} F$$

Linear Model:

$$s\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -4.9 & 0 & 0 \\ 0 & 14.7 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ -0.5 \end{bmatrix} F$$



Eigenvalues:

- {0, 0, -3.83, +3.83}

Pole Placement:

Requirement:

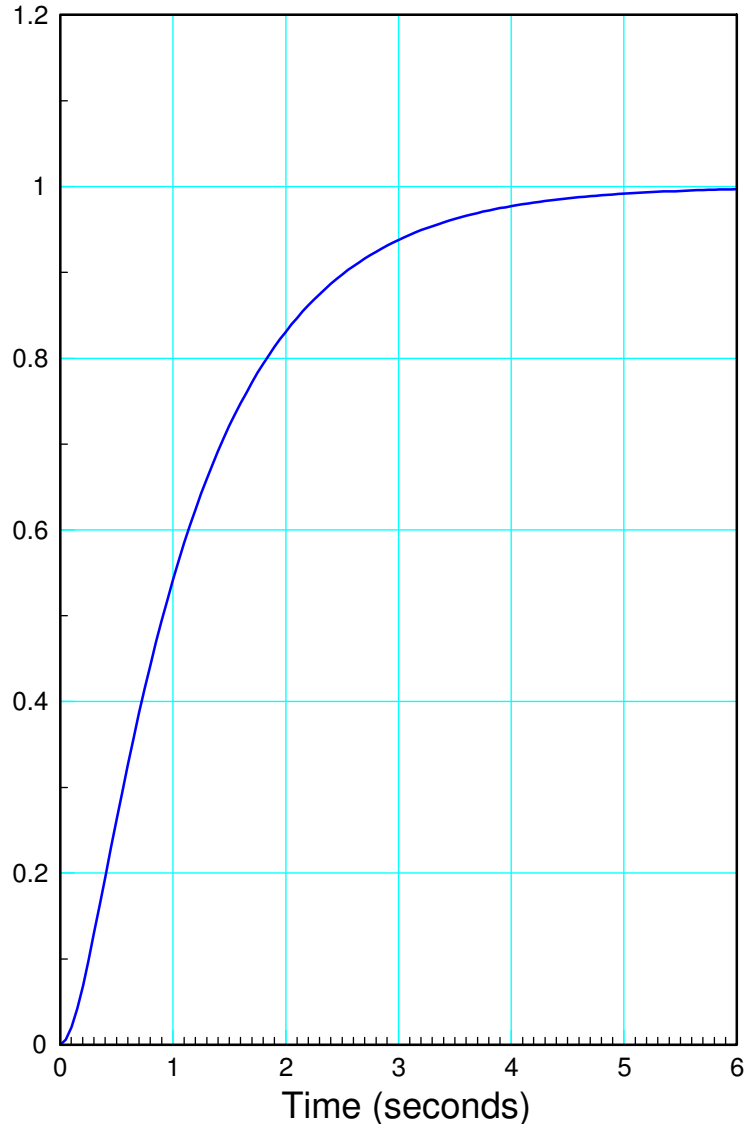
- 2% settling time of 4 seconds
- No overshoot
- DC gain = 1.000

Translation

- Place the dominant pole at s = -1

Pole Placement:

- Place the poles at {-1, -2, -3, -4}
- Somewhat arbitrary



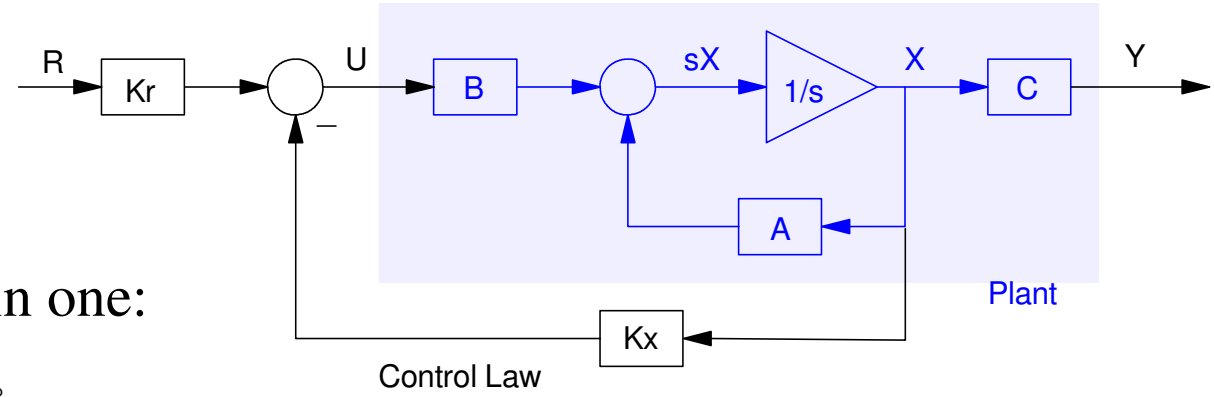
Matlab Code:

First, input the system

```
A = [0,0,1,0;  0,0,0,1;  0,-4.9,0,0;  0,14.7,0,0]

          0          0    1.0000          0
          0          0          0    1.0000
          0   -4.9000          0          0
          0   14.7000          0          0

B = [0;0;0.5;-0.5]

      0
      0
      0.5
     -0.5

eig(A)

          0
          0
     3.8341
    -3.8341
```

Compute the feedback gains

```
>> Kx = ppl(A, B, [-1, -2, -3, -4])

   -4.8980 -104.2980  -10.2041  -30.2041

>> eig(A - B*Kx)

   -4.0000
   -3.0000
   -1.0000
   -2.0000
```



Find Kr to make the DC gain one:

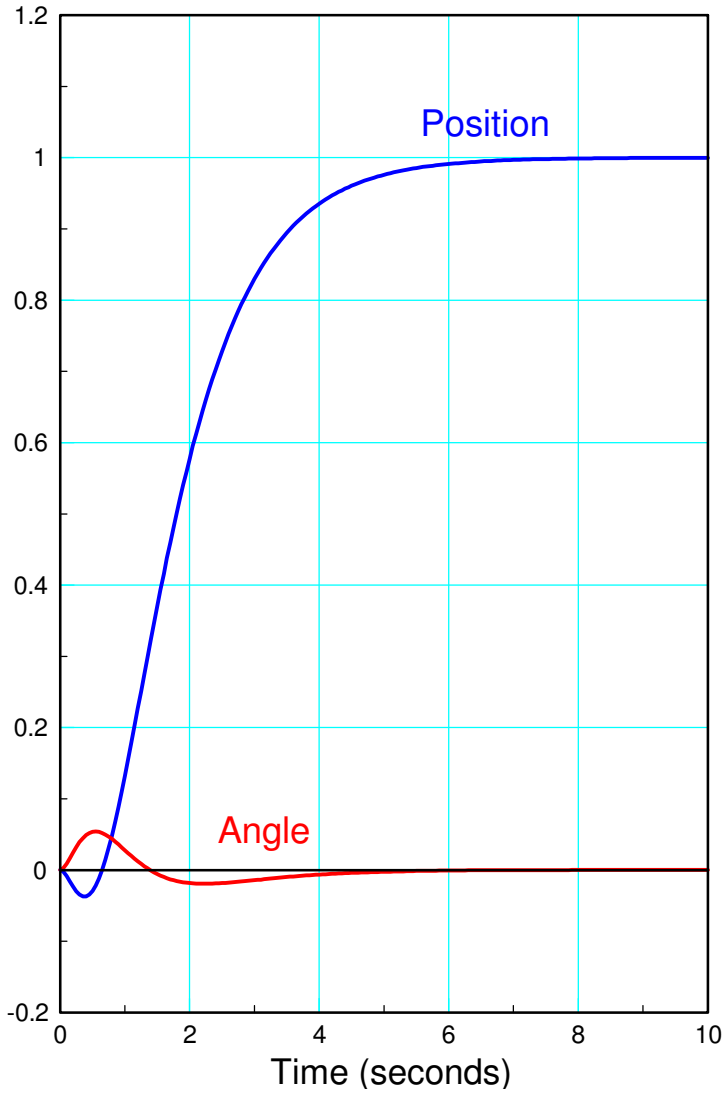
```
>> DC = -C*inv(A - B*Kx)*B

   -0.2042

>> Kr = 1/DC

   -4.8980
```

Does this meet the requirements?

Check the step response

- Position (x), and
- Angle (θ)

```
>> C2 = [1,0,0,0; 0,1,0,0]

     1     0     0     0          position
     0     1     0     0          angle

>> D2 = [0;0]

     0
     0

>> G = ss(A-B*Kx, B*Kr, C2, D2);
>> t = [0:0.05:10]';
>> y = step(G,t);
>> plot(t,y)
>> xlabel('Time (seconds)');
```



Is the response reasonable?

Check the input

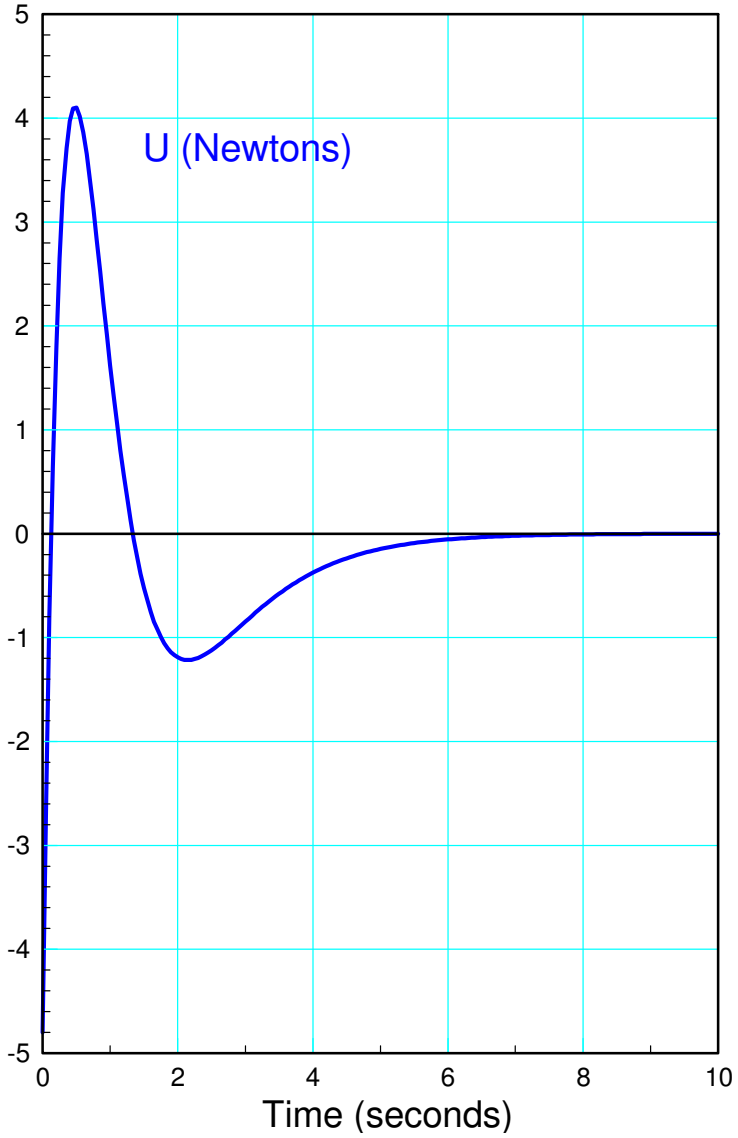
- Max force = 5N
- Lets you size the motor

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

$$C = -K_x$$

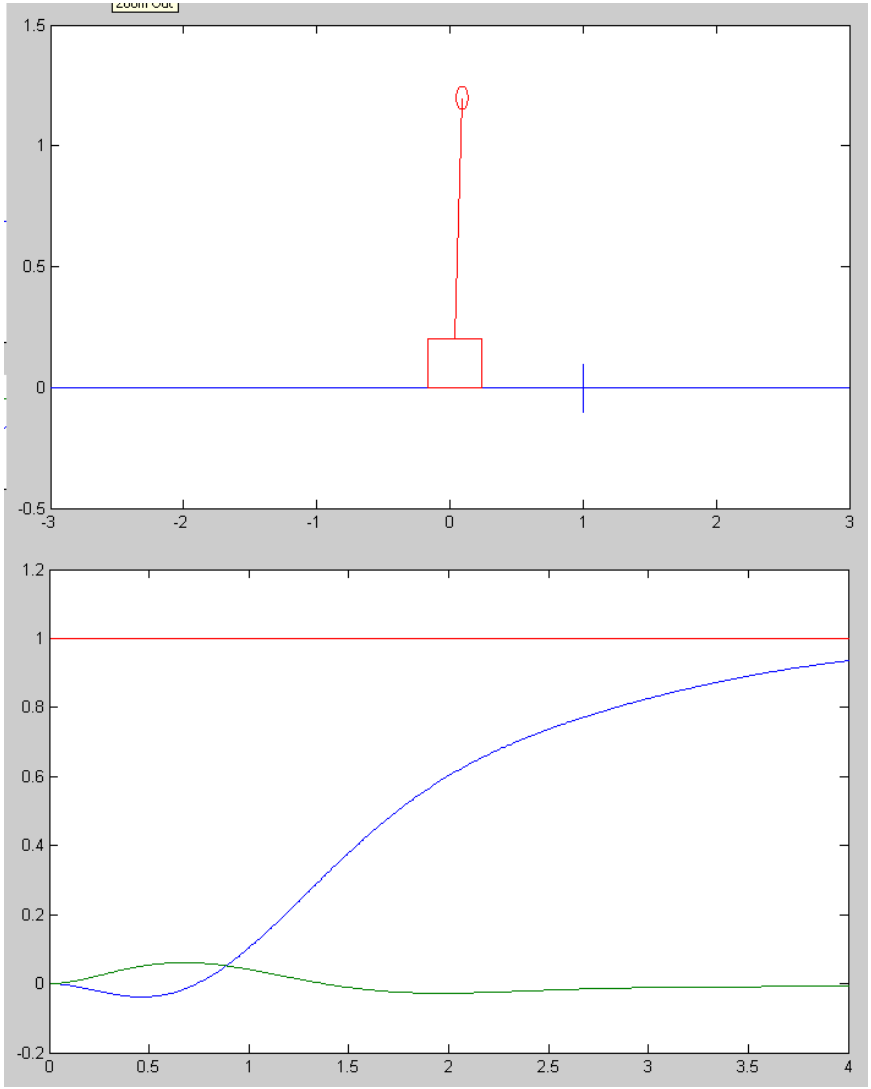
$$D = K_r$$

```
Gu = ss(A-B*Kx, B*Kr, -Kx, Kr);

U = step(Gu,t);
plot(t,U);
xlabel('Time (seconds)');
ylabel('Input (U)');
```



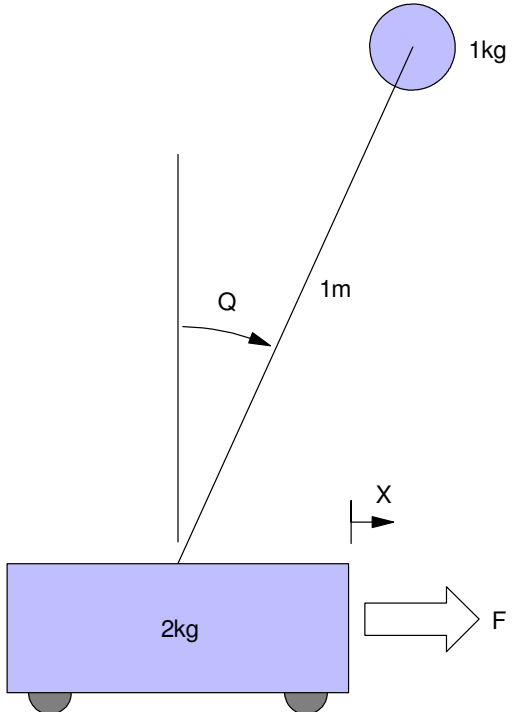
Nonlinear Simulation

```
% Cart and Pendulum
X = zeros(4,1);
dX = zeros(4,1);
Ref = 1;
dt = 0.01;
t = 0;
Kx = [-4.89 -104.29 -10.20 -30.20];
Kr = -4.8980;

U = 0;
y = [];
while(t < 4)
  U = Kr*Ref - Kx*X;
  dX = CartDynamics(X, U);
  X = X + dX * dt;
  t = t + dt;
  CartDisplay(X, Ref);
  y = [y; X(1), X(2), Ref];
end
```



Summary

- Balancing a yardstick on your hand isn't that hard (for a computer)
- Pole placement (Bass Gura) allows you to find stabilizing feedback gains
- It requires measurements of all four states
  - Position
  - Velocity
  - Angle
  - Angular Velocity

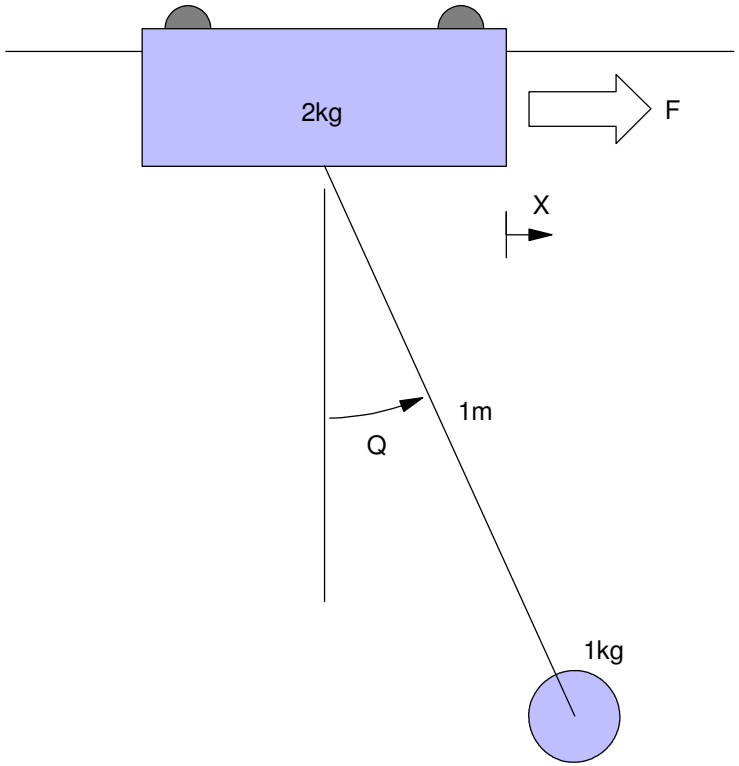


Gantry System

- Reverse gravity:

Nonlinear Model

$$\begin{bmatrix} 3 & 1\cos\theta \\ \cos\theta & 1 \end{bmatrix}\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta}^2\sin\theta \\ -g\sin\theta \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} F$$



Linear Model:

$$s\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 4.9 & 0 & 0 \\ 0 & -14.7 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ -0.5 \end{bmatrix} F$$

Eigenvalues:

- {0, 0, -j3.83, +j3.83}

Pole Placement:

Requirement:

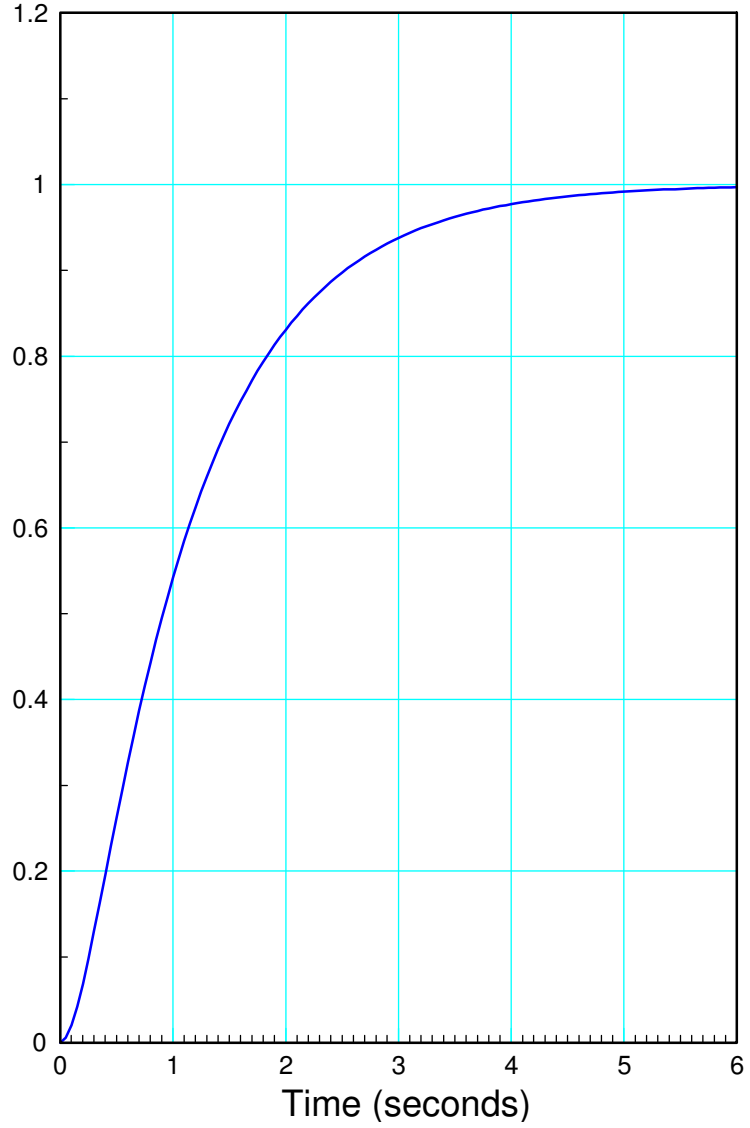
- 2% settling time of 4 seconds
- No overshoot
- DC gain = 1.000

Translation

- Place the dominant pole at s = -1

Pole Placement:

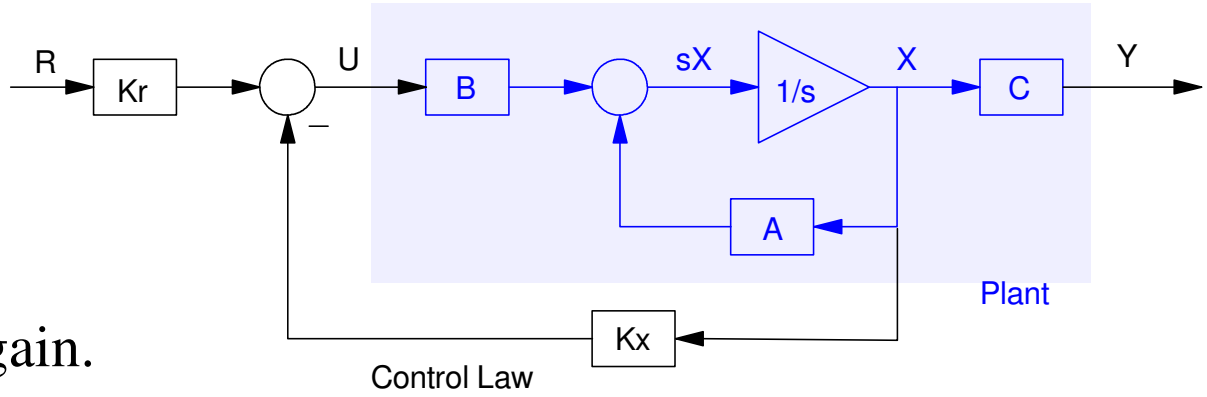
- Place the poles at {-1, -2, -3, -4}
- Somewhat arbitrary



Finding Kx and Kr

Use Bass Gura:

```
>> Kx = ppl(A, B, [-1, -2, -3, -4])

    4.8980  -35.7020   10.2041   -9.7959

>> eig(A - B*Kx)

   -4.0000
   -3.0000
   -1.0000
   -2.0000
```



Now find Kr to set the DC gain.

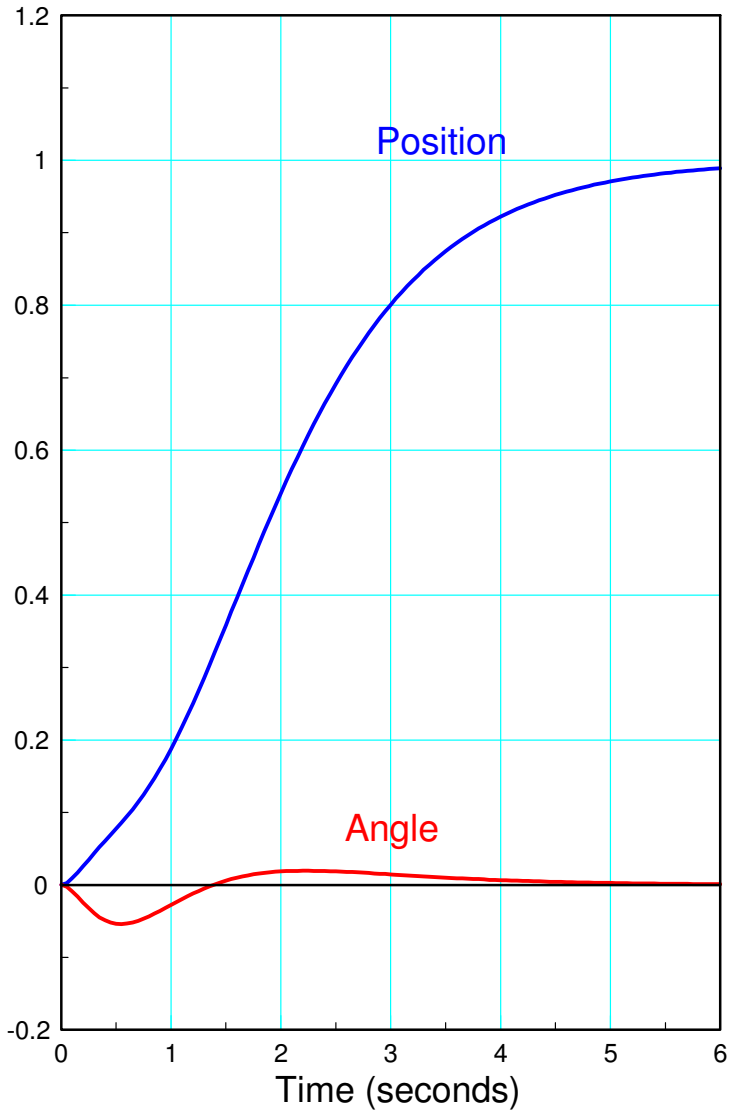
```
>> DC = -C*inv(A - B*Kx)*B

    0.2042

>> Kr = 1/DC

    4.8980
```

Does this meet the requirements?

- DC gain = 1
- No overshoot
- Ts = 4 seconds

Take the Step Response:

```
>> C2 = [1,0,0,0; 0,1,0,0]

     1     0     0     0        position
     0     1     0     0        angle

>> D2 = [0;0]

     0
     0

>> G = ss(A-B*Kx, B*Kr, C2, D2);
>> t = [0:0.05:6]';
>> y = step(G,t);
>> plot(t,y)
>> xlabel('Time (seconds)');
>>
```



Is the control law reasonable?

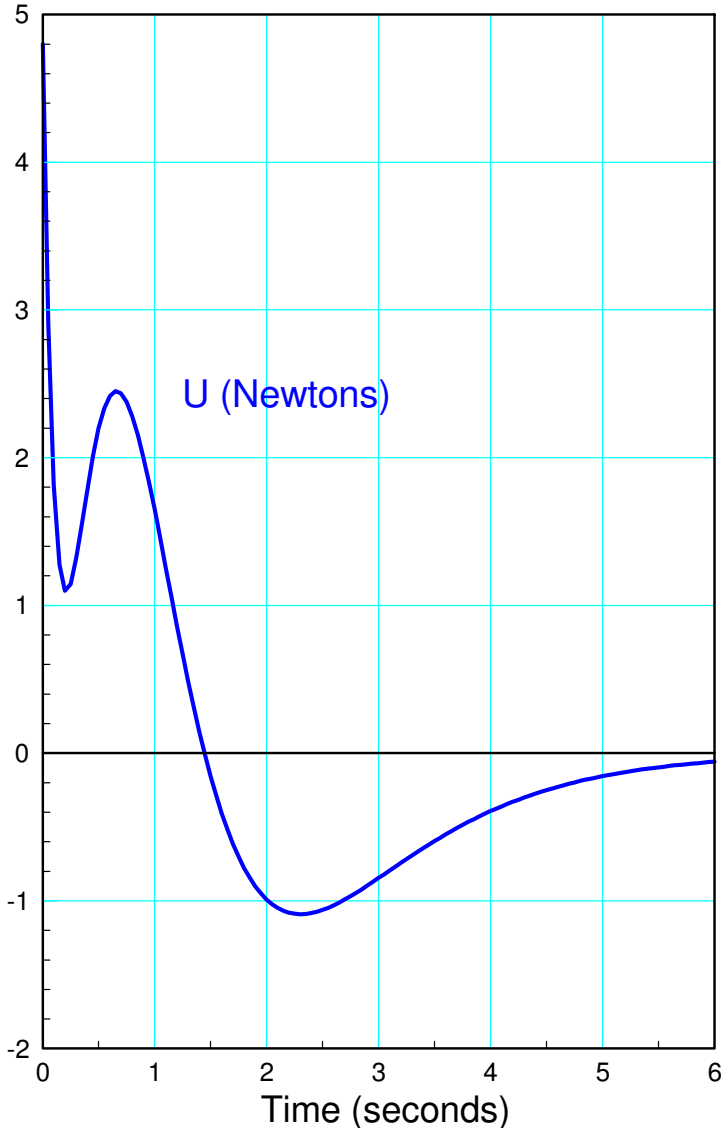
- Is the input excessive?

Plot the resulting input (U)

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

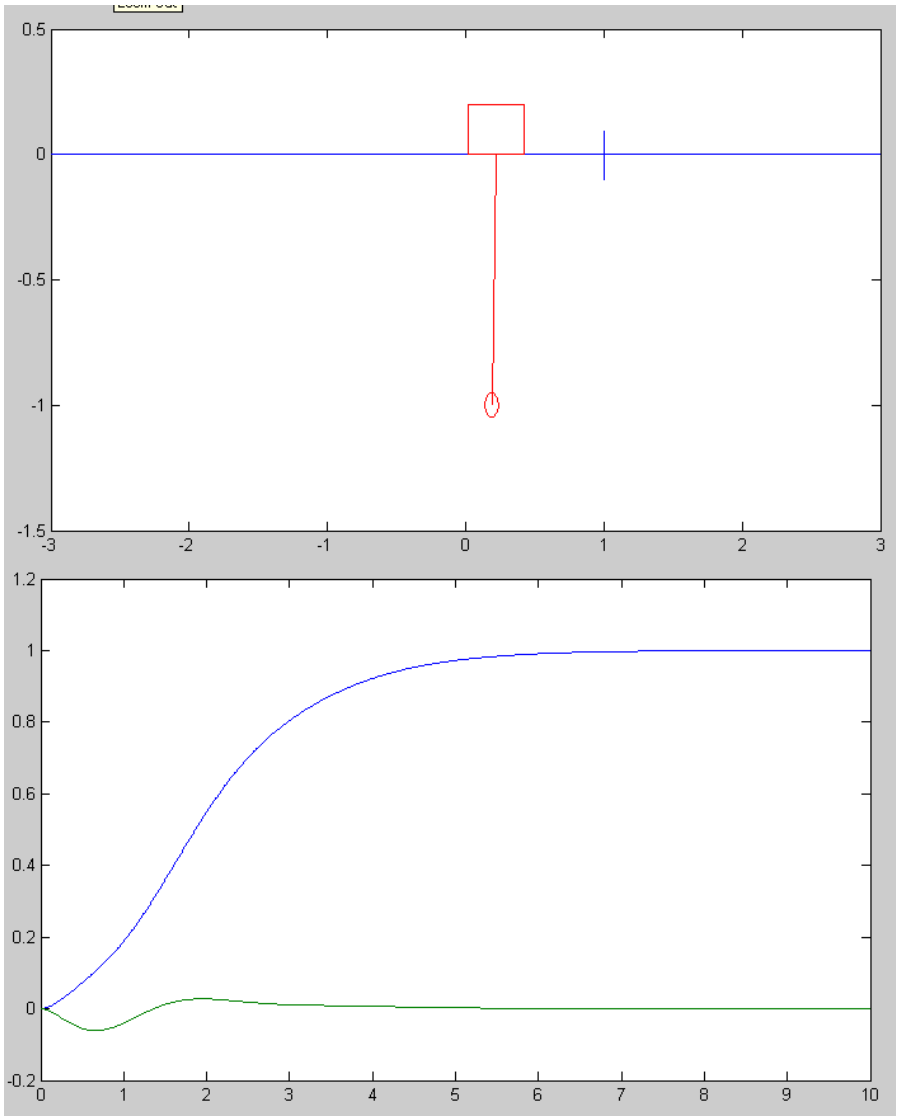
```
>> Gu = ss(A-B*Kx, B*Kr, -Kx, Kr);
>> U = step(Gu,t);
>> plot(t,U);
>> xlabel('Time (seconds)');
>> ylabel('Input (U)');
```

The motor needs to be able to apply 5N of force



Matlab Simulation (Gantry)

```
% Gantry System
X = [0;0;0;0];
dX = zeros(4,1);
Ref = 1;
dt = 0.01;
t = 0;
y = [];
Kx = [4.898 -35.70 10.20 -9.79];
Kr = 4.8980;


while(t < 10)
   U = Kr*Ref - Kx*X;
   dX = GantryDynamics(X, U);
   X = X + dX * dt;
   t = t + dt;
   GantryDisplay(X, Ref);
   y = [y ; X(1), X(2) ];
   end


hold off
t = [0:length(y)-1]' * dt;
plot(t,y);
```



Summary

- Controlling a gantry system isn't that hard (for a computer)
- Pole placement (Bass Gura) allows you to find stabilizing feedback gains
- It requires measurements of all four states
  - Position
  - Velocity
  - Angle
  - Angular Velocity