Servo-Compensator Design

NDSU ECE 463/663

Lecture #16

Inst: Jake Glower

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

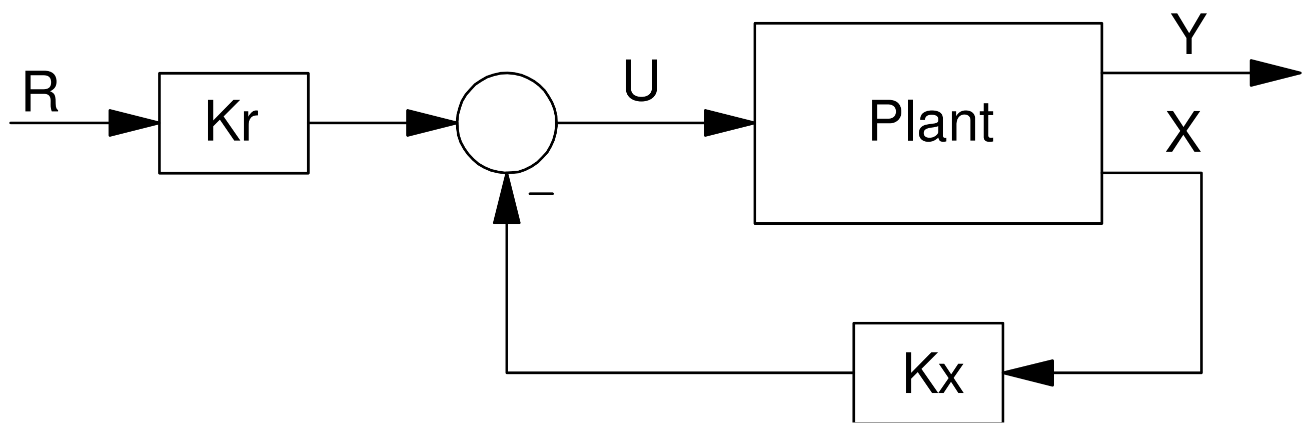
Tracking a Set Point

How do you force a system to track a set point?

- How do you force the DC gain to be 1.0000?

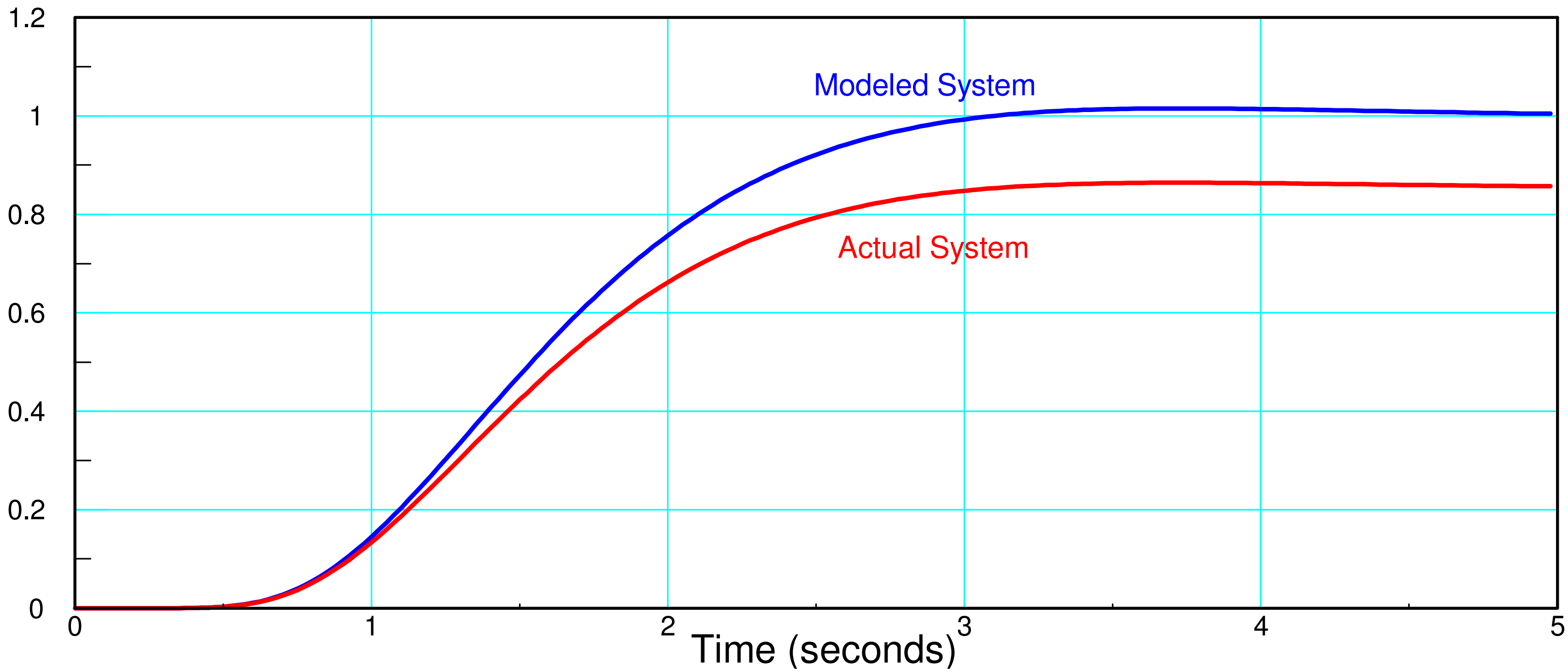
Previous Solution:

- Add a gain, Kr, to make the DC gain equal to 1.0000



What if the plant model is uncertain?

- If the dynamics change, the DC gain will change
- If the model is incorrect the DC gain will be incorrect
- Kr needs to change accordingly.



Servo Compensator

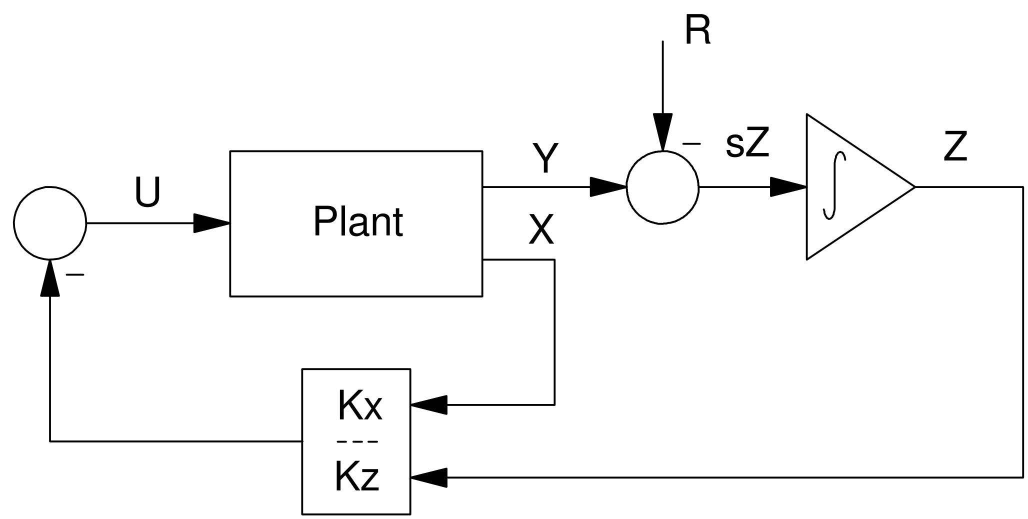
Add an integrator (termed a *servo compensator*)

- $Z = \int (Ref - Y)dt$

At steady state

- $\frac{dZ}{dt} = 0$
- Y = Ref

Add feedback gains to stabilize the closed-loop system



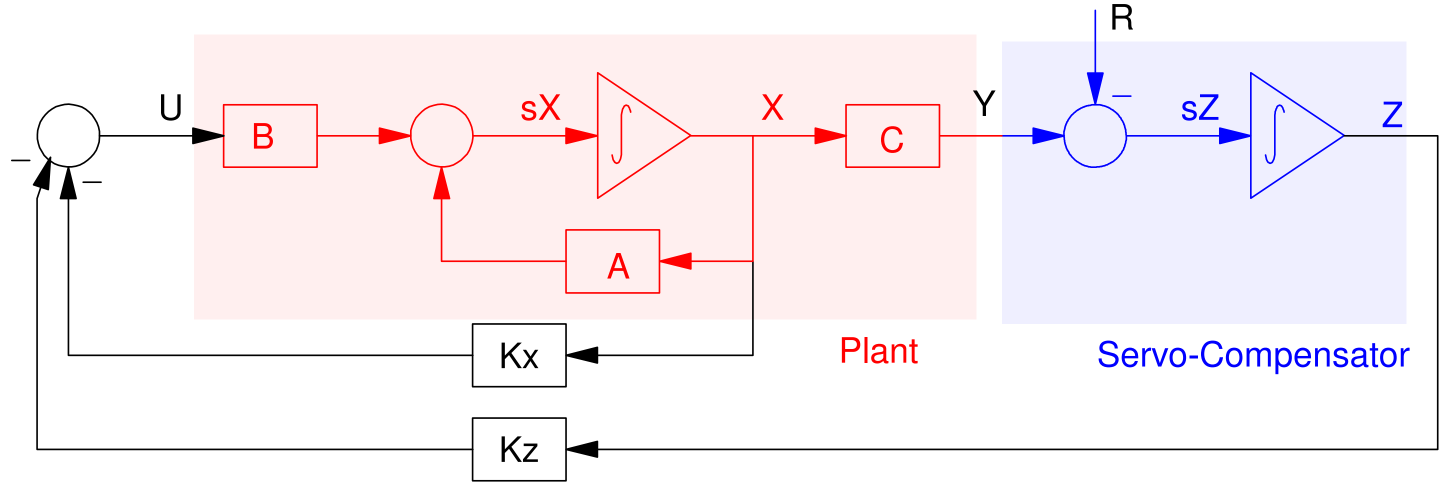
Open-Loop System

$$s\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}\begin{bmatrix} X \\ Z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} U + \begin{bmatrix} 0 \\ -1 \end{bmatrix} R$$

$$U = -\begin{bmatrix} K_x & K_z \end{bmatrix}\begin{bmatrix} X \\ Z \end{bmatrix}$$

Closed-Loop System

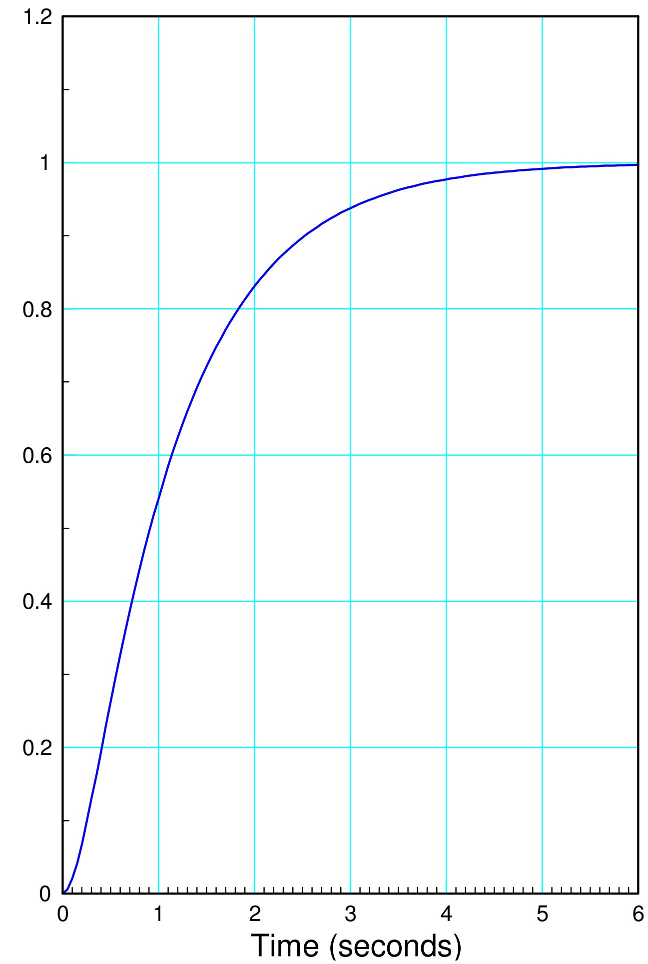
$$s\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} A - BK_x & -BK_z \\ C & 0 \end{bmatrix}\begin{bmatrix} X \\ Z \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} R$$



Example: 4th-Order Heat Equation

Design a feedback control law for a 4-stage RC filter so that

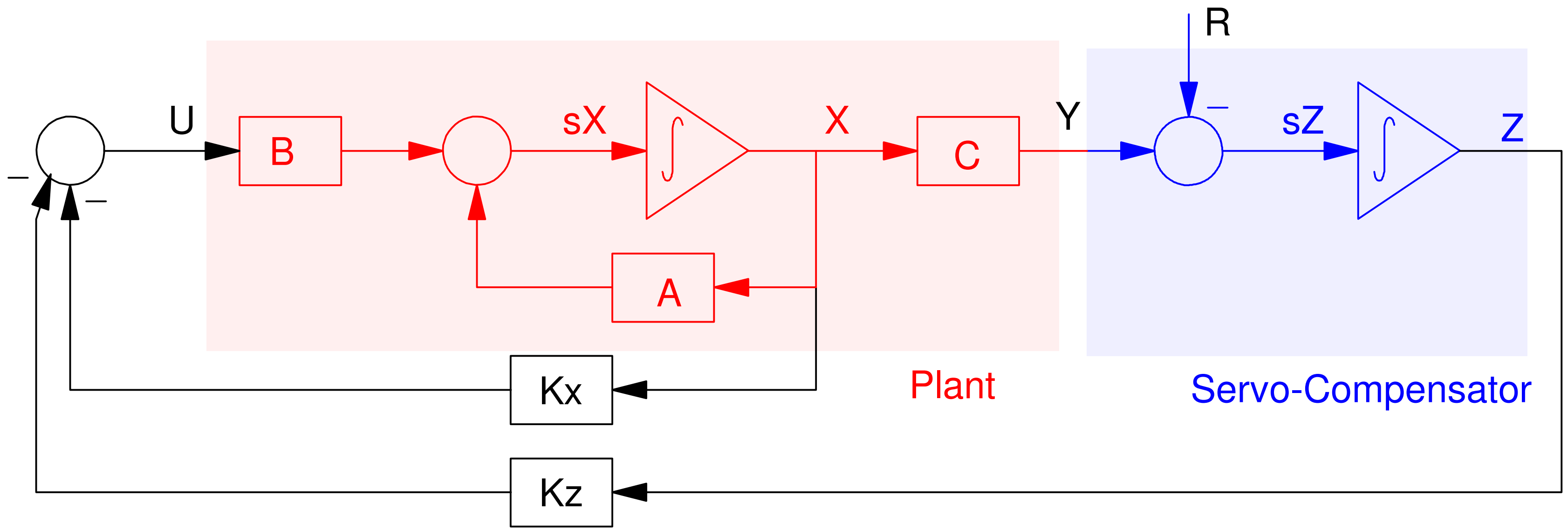
- The 2% settling time is 4 seconds,
- There is no overshoot for a step input, and
- The DC gain from R to Y is one.



Solution: Add a servo compensator:

Open-Loop Plant & Servo:

$$\begin{bmatrix} sX \\ \cdots \\ sZ \end{bmatrix} = \begin{bmatrix} A & \vdots & 0 \\ \cdots & & \cdots \\ C & \vdots & 0 \end{bmatrix} \begin{bmatrix} X \\ \cdots \\ Z \end{bmatrix} + \begin{bmatrix} B \\ \cdots \\ 0 \end{bmatrix} U \qquad U = \begin{bmatrix} -K_x & \vdots & -K_z \end{bmatrix} \begin{bmatrix} X \\ \cdots \\ Z \end{bmatrix}$$



In Matlab, input the system:

```
A = [-2,1,0,0;1,-2,1,0;0,1,-2,1;0,0,1,-1]
B = [1;0;0;0]
C = [0,0,0,1];
```

Create the augmented system (plant plus servo-compensator)

```
A5 = [A, B*0; C, 0]

    -2     1     0     0     0
     1    -2     1     0     0
     0     1    -2     1     0
     0     0     1    -1     0
     0     0     0     1     0

B5 = [B; 0]

     1
     0
     0
     0
     0
```

Use Bass-Gura to place the poles at {-1, -2, -2, -2, -2}.

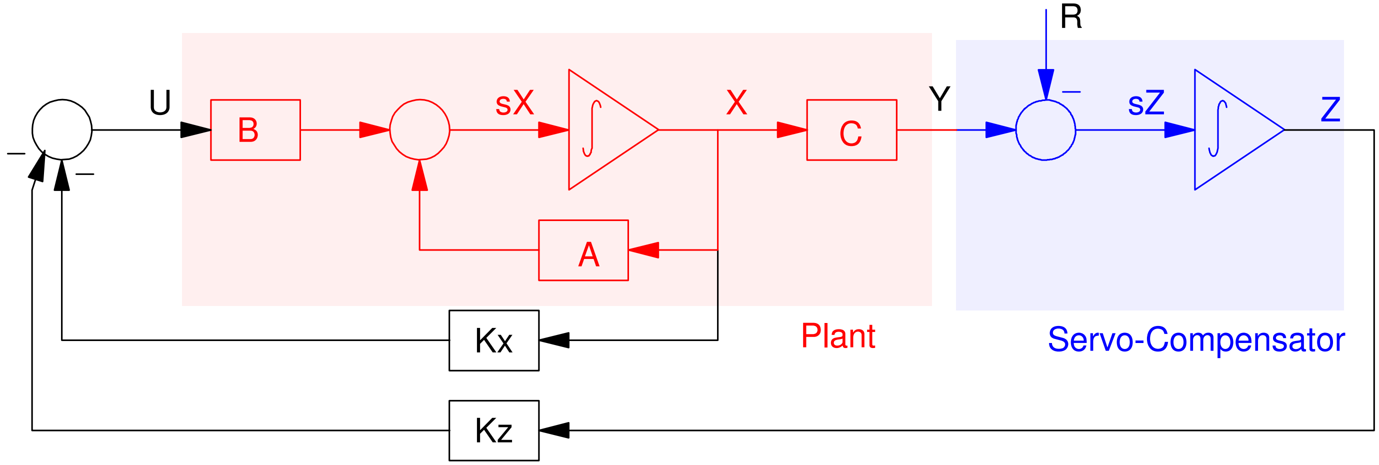
```
>> K5 = ppl(A5, B5, [-1,-2,-2,-2,-2])

     2.0000    7.0000   13.0000   25.0000   16.0000

>> eig(A5-B5*K5)

   -2.0005
   -2.0000 + 0.0005i
   -2.0000 - 0.0005i
   -1.9995
   -1.0000
```

The feedback gains, Kx and Kz, are then

```
>> Kx = K5(1:4)

     2.0000    7.0000   13.0000   25.0000

>> Kz = K5(5)

    16.0000
```

The closed-loop system is

$$s\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} A - BK_x & -BK_z \\ C & 0 \end{bmatrix}\begin{bmatrix} X \\ Z \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} R$$

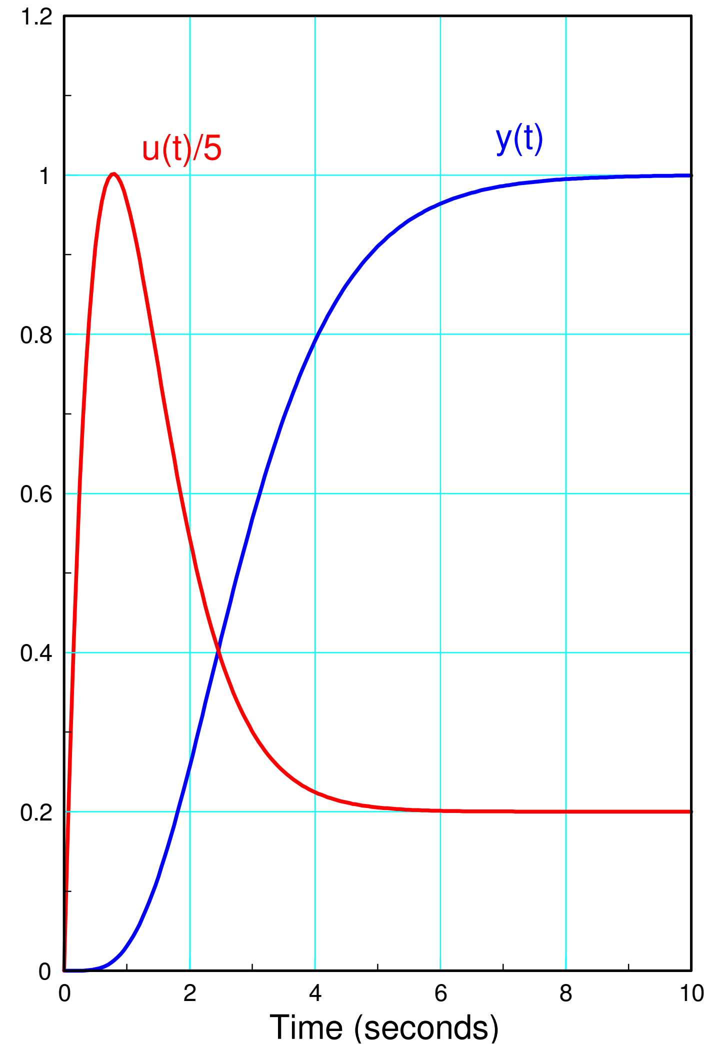


To plot the step response from R to both Y and U, define the output to be

$$\begin{bmatrix} Y \\ U \end{bmatrix} = \begin{bmatrix} C & 0 \\ -Kx & -Kz \end{bmatrix} \begin{bmatrix} X \\ Z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} R$$

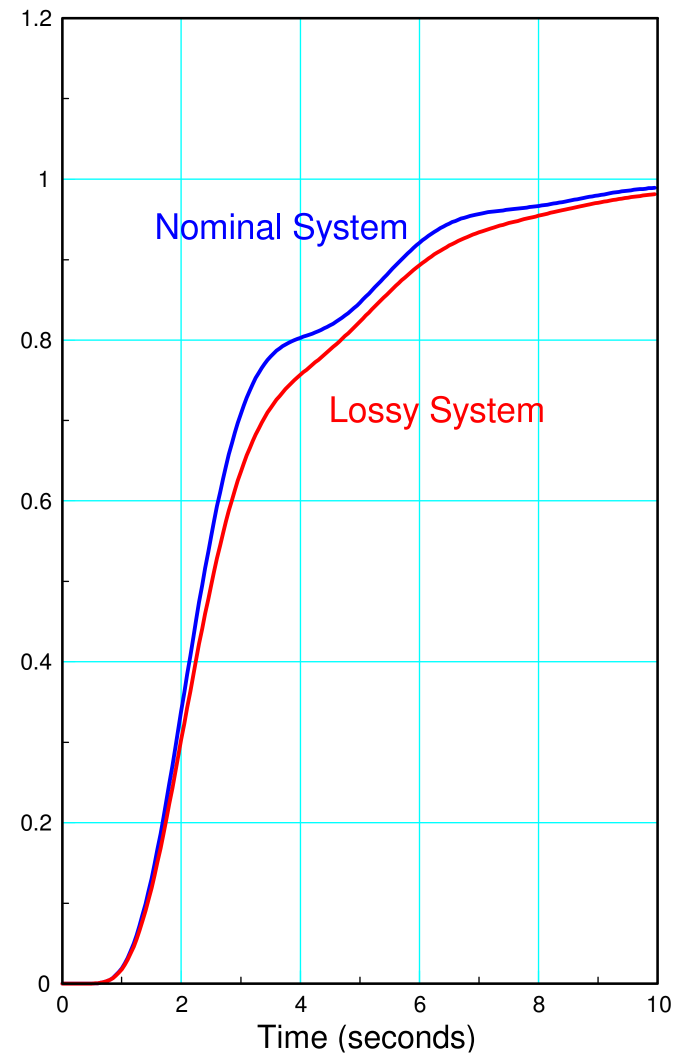
In Matlab:

```
>> C5 = [C, 0; -K5]

     0     0     0     1      0
    -2    -7   -13  -250    -16

>> D5 = [0; 0]

     0
     0

>> Br = [0*B ; -1];
>> G5 = ss(A5 - B5*K5, Br, C5, D5);
>> y = step(G5,t);
>> plot(t,y)
>>
```



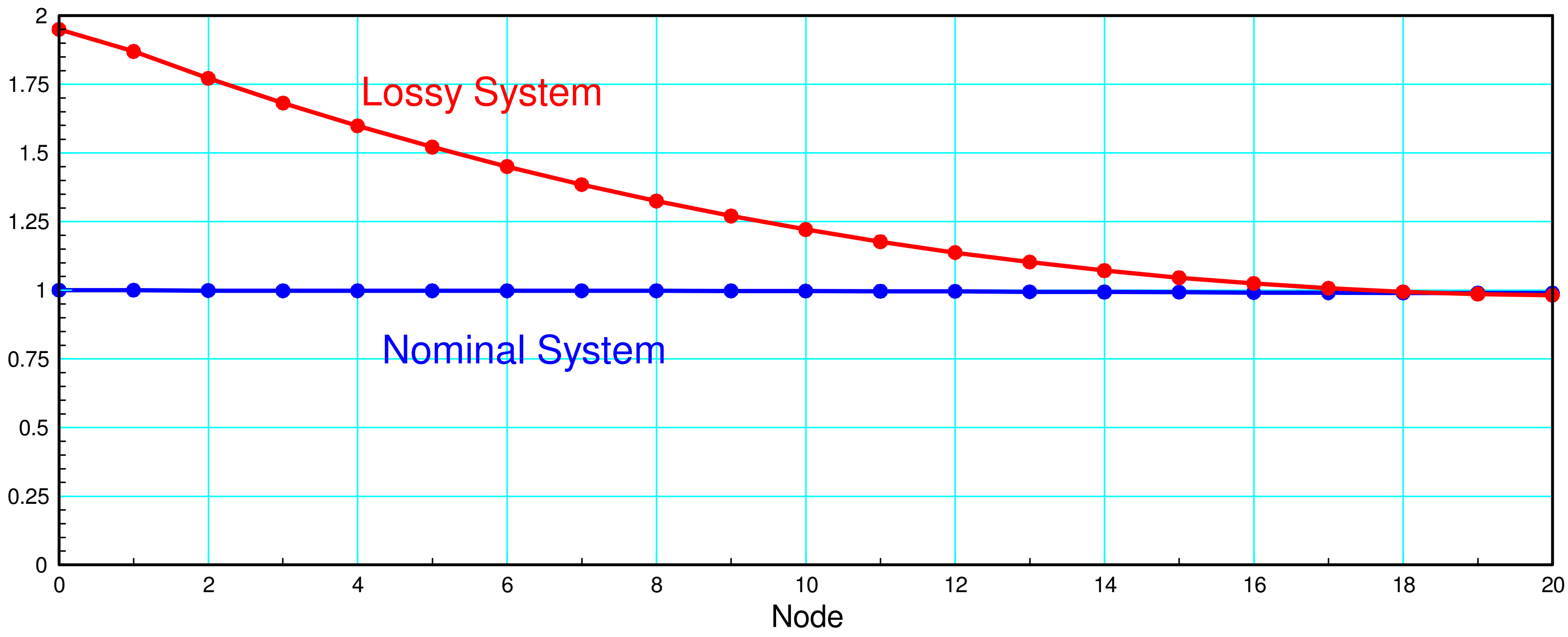
Matlab Simlation (modified Heat20.m)

```
V = zeros(20,1);
Z = 0;
Ref = 1;
Kx = [2 7 13 25];
Kz = 16;
dt = 0.001;
t = 0;
while(t < 10)
   for i2=1:10
      X = [V(5) ; V(10) ; V(15) ; V(20)];
      V0 = -Kx*X - Kz*Z;
      dZ = V(20) - Ref;
      dV(1) = 25*V0 - 50.1*V(1) + 25*V(2);
      for i=2:19
         dV(i) = 25*V(i-1) - 50.1*V(i)
               + 25*V(i+1);
         end
      dV(20) = 25*V(19) - 25.1*V(20);
      V = V +  dV*dt;
      Z = Z +  dZ*dt;
      t = t + dt;
      end
   plot([0:20], [V0;V], '.-');
   ylim([0,2]);
   pause(0.01);
   end
```



Comments:

- The servo compensator forces the output to track a constant set point
- It does so by adding a constant to the input (integration constant)
- It searches until it finds the constant that forces $Y \rightarrow Ref$



-

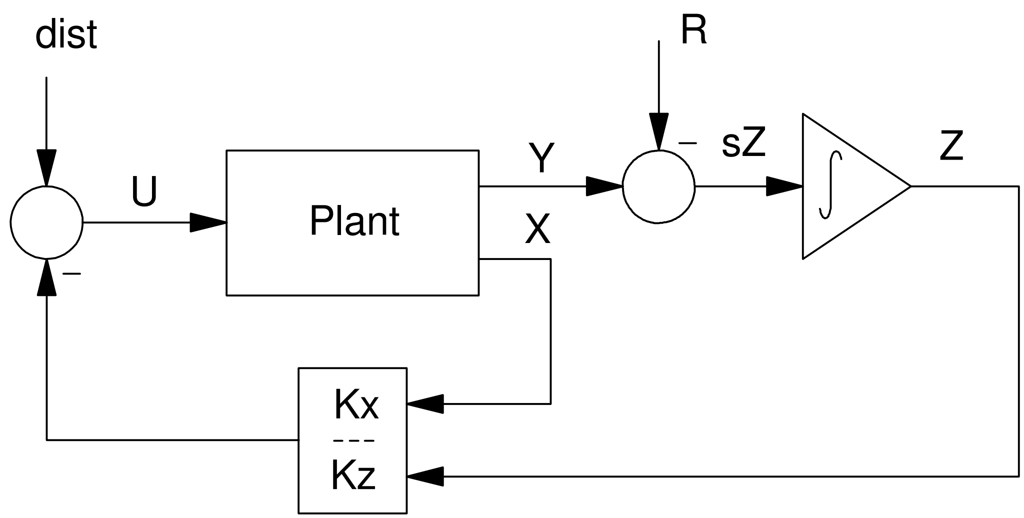
Constant Disturbance:

Problem: Suppose a system has a constant disturbance.

- Design a feedback control law that forces Y=R in spite of this disturbance.

Solution: Add a servo compensator

- At steady state, sZ = 0
- Z is an integration constant
  - Cancels the disturbance and
  - Forces Y = R

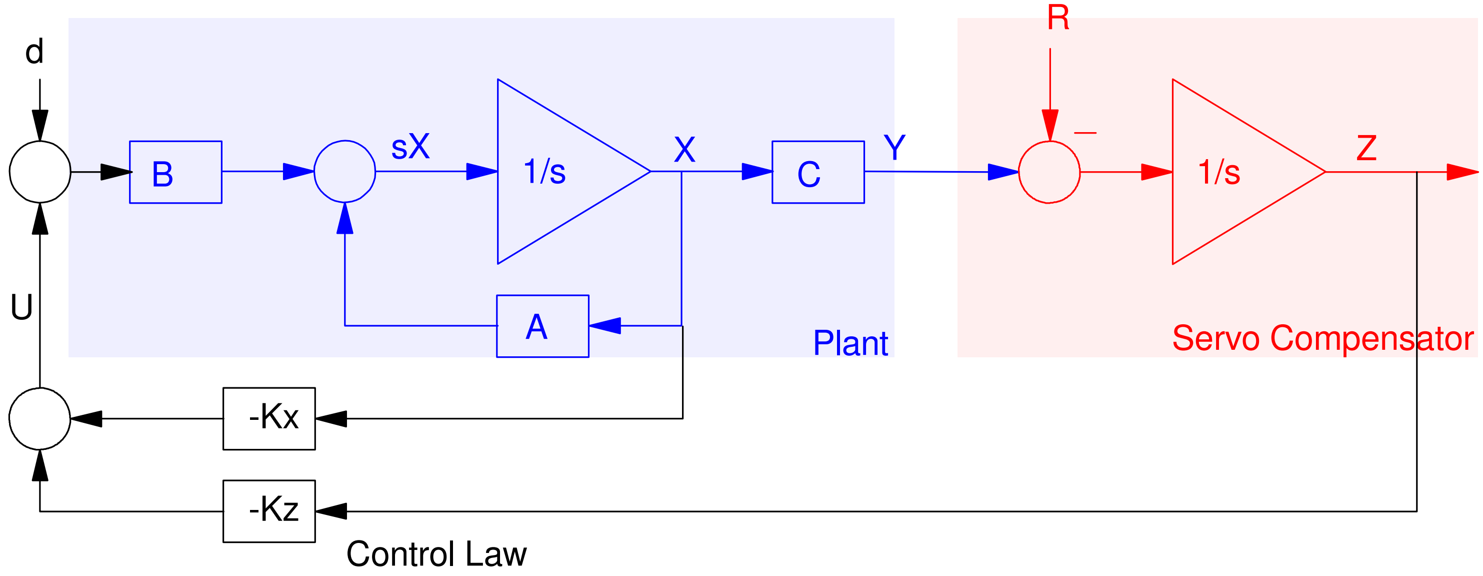


Step Response with Respect to What?

Note: The B matrix determines *what* is being stepped.

The closed-loop plant + setpoint + disturbance dynamics are...

$$s\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} A - BK_x & -BK_r \\ C & 0 \end{bmatrix}\begin{bmatrix} X \\ Z \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} R + \begin{bmatrix} B \\ 0 \end{bmatrix} d$$



Step Response with Respect to the Set Point (R)

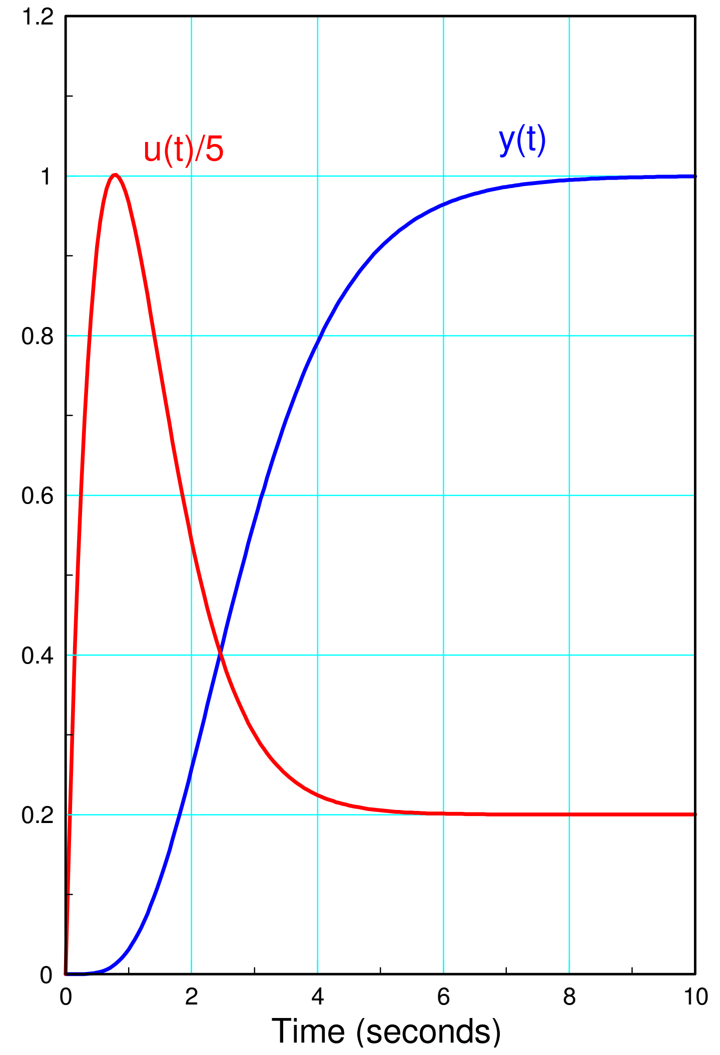
Use the B matrix with repect to R

- B = Br

```
Br = [0*B;-1];
G5 = ss(A5 - B5*K5, Br, C5, D5);

y = step(G5,t);
plot(t,y);
```

The system tracks a constant set point

- The servo compensator figures out what u(t) needs to be to force $y(t) \rightarrow 1$



Step Response with Respect to the Disturbance (d)

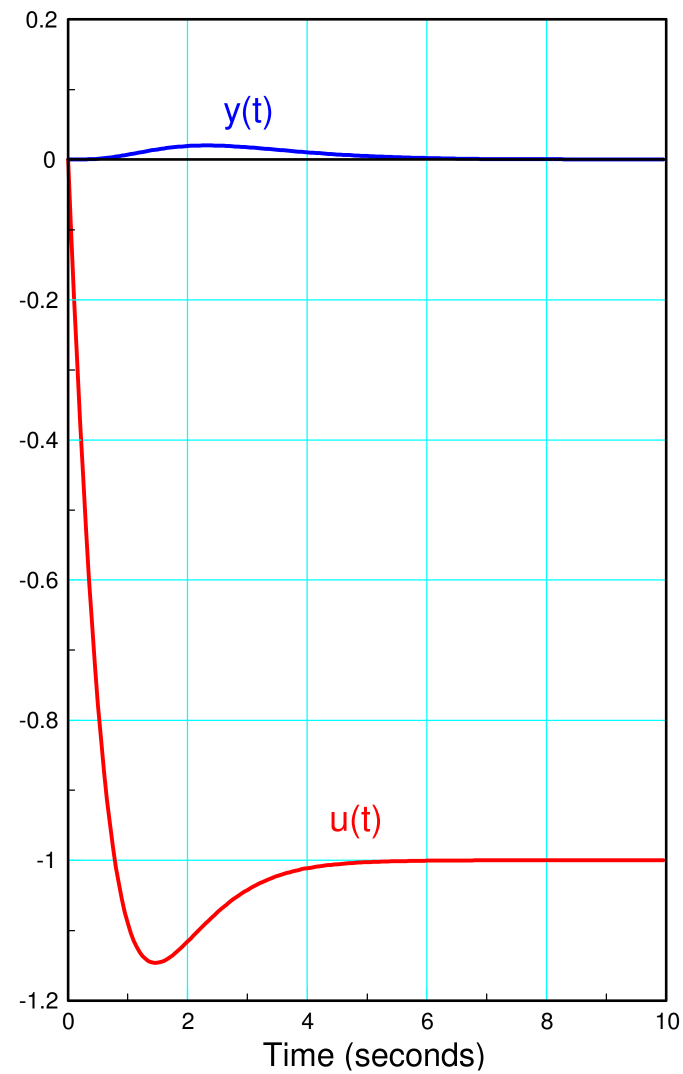
Use the B matrix with respect to d

- Bd = [B ; 0]

```
>> Bd = [B ; 0];
>> G5 = ss(A5 - B5*K5, Bd, C5, D5);
>> y = step(G5,t);
>> plot(t,y);
```

The servo compensator searches to find the constant that forces Y = R

- u(t) = -1 cancels with the disturbance
- u(t) + d(t) = 0
- (-1) + (+1) = 0



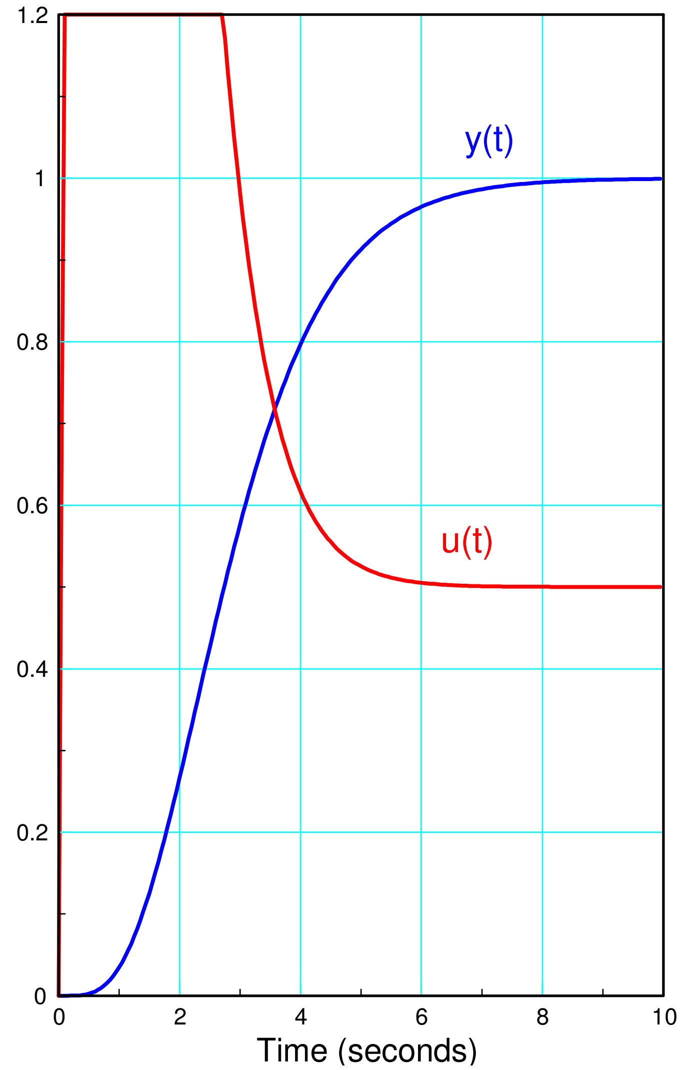
Step Response with Respect to both R & d

Let

- R = 1
- d = 0.5

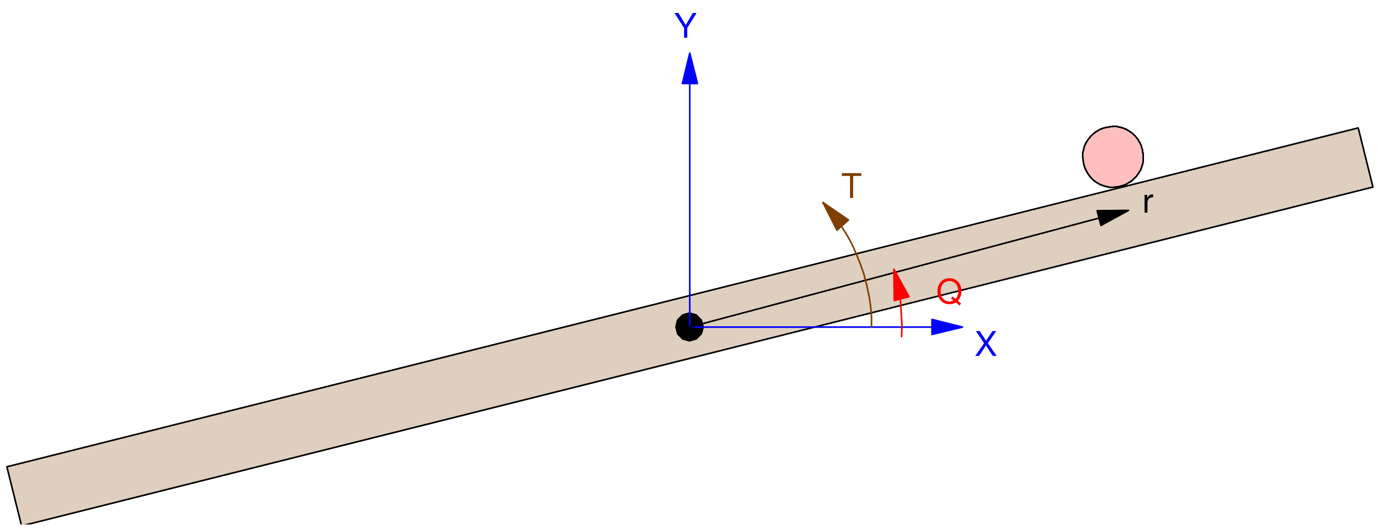
Use the B matrix with respect to d

```
>> Br = [0*B ; -1];
>> Bd = [B ; 0];
>> G5 = ss(A5 - B5*K5, Br+0.5*Bd, C5, D5);
>> y = step(G5,t);
>> plot(t,y);
```



Example 2: Ball and Beam System

- m = 1kg
- J = 0.2 kg m^2

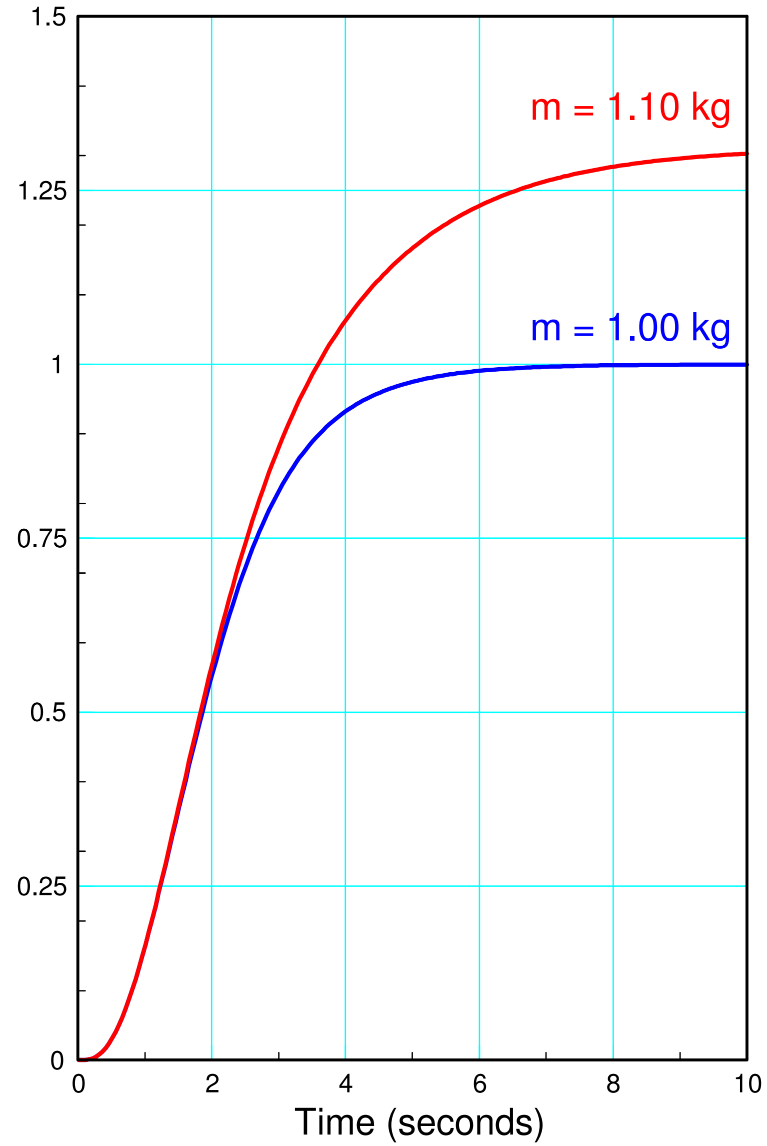


Full-State Feedback

- No servo compensator
- Poles at {-1, -2, -3, -4}
- $U = K_r R - K_x X$

Kr makes the DC gain 1 when m = 1kg

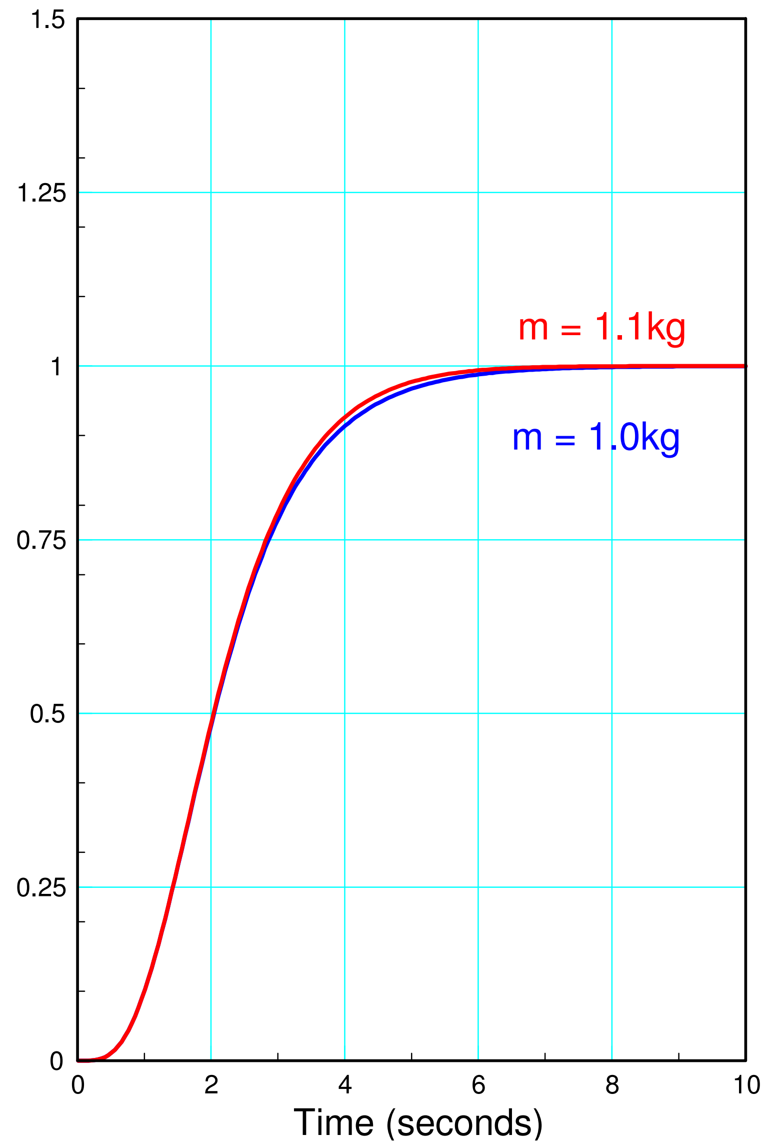
- Increase the mass to 1.1kg
- Extra torque on the beam acts as a constant disturbance
- No longer tracks



Add a Servo Compensator

- Poles at {-1, -2, -3, -4, -5}
- Now tracks a constant set point
- Now rejects a constant disturbance

```
X = [0, 0, 0, 0]';
Z = 0;
dt = 0.01;
t = 0;
Kx = [-56.774 102.004 -38.573 18.00];
Kz = -20.5723;
Ref = 1;
y = [];

while(t < 10)
   U = -Kz*Z - Kx*X;
   dX = BeamDynamics(X, U);
   dZ = X(1) - Ref;
   X = X + dX * dt;
   Z = Z + dZ*dt;
   y = [y ; Ref, X(1)];
   t = t + dt;
   BeamDisplay(X, Ref);
   end
```



Summary

A servo compensator creates an augmented system:

$$\begin{bmatrix} sX \\ sZ \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} X \\ Z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} U + \begin{bmatrix} 0 \\ -1 \end{bmatrix} R$$

With a servo compensator, you can

- Track a constant set point, and
- Reject constant disturbances