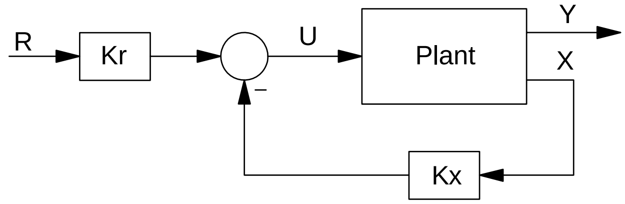
Servo-Compensator Design

Problem: How do you force a system to track a set point? If the set point is a constant, how do you force the DC gain to be one?

The solution we looked at before is to add an input gain, Kr. This gain is adjusted so that the DC gain is one.

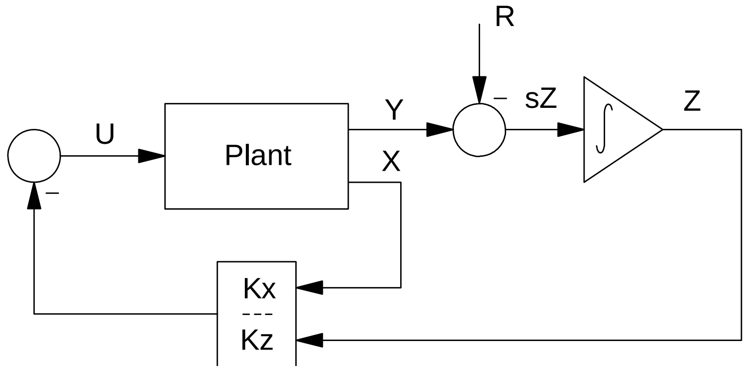


Solution 1: Select Kr to force the DC gain to be one.

Problem: How do you force the DC gain to be one if the plant model is uncertain?

Solution: Assuming the system is not changing with time, the system has a DC gain. We may not know what it is before-hand, but it is something. This means that there is some constant input, U, which forces the output to one. The problem is we don't know what that constant is.

One way to force Y→R is to integrate the difference, creating a new state, Z. This integrator is termed a *servo compensator*.



Solution 2: Integrate the error between Y and R. At steady-state, sZ = 0.

$$Z = \int_0^t (Y - R)d\tau$$

$$sZ = (Y - R)$$

Then, make the feedback control law

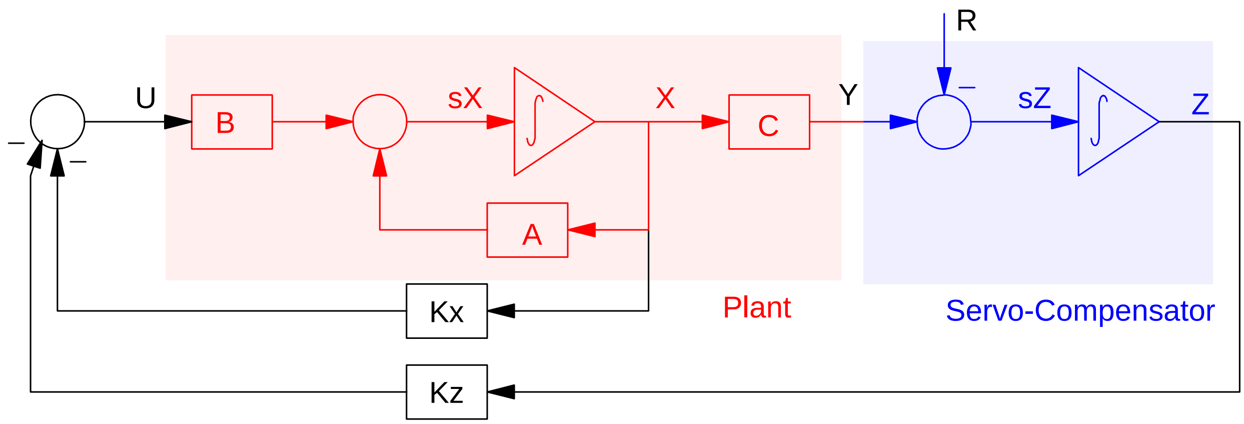
$$U = -K_z Z - K_x X$$

If you make the system stable and R is a constant, then the system will reach a steady-state value with all states being constants - meaning s=0.

$$sZ = 0 = R - Y$$

The control law then forces the output to track the set point.

In state-space, the system looks like the following



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

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

The closed-loop system is then

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

Example: 4th-Order Heat Equation

Design a feedback control law for the following system 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.

To force the DC gain to one, add an integrator (servo compensator).

To force the settling time to 4 seconds with no overshoot, place the closed-loop dominant pole at s = -1. Somewhat arbitrarily, place all 5 poles at {-1, -2, -3, -4, -5}

The system is:

$$sX = \begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} X + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} U$$

$$Y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} X$$

Solution: Input the system's (A,B,C,D) in to MATLAB:

```
-->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

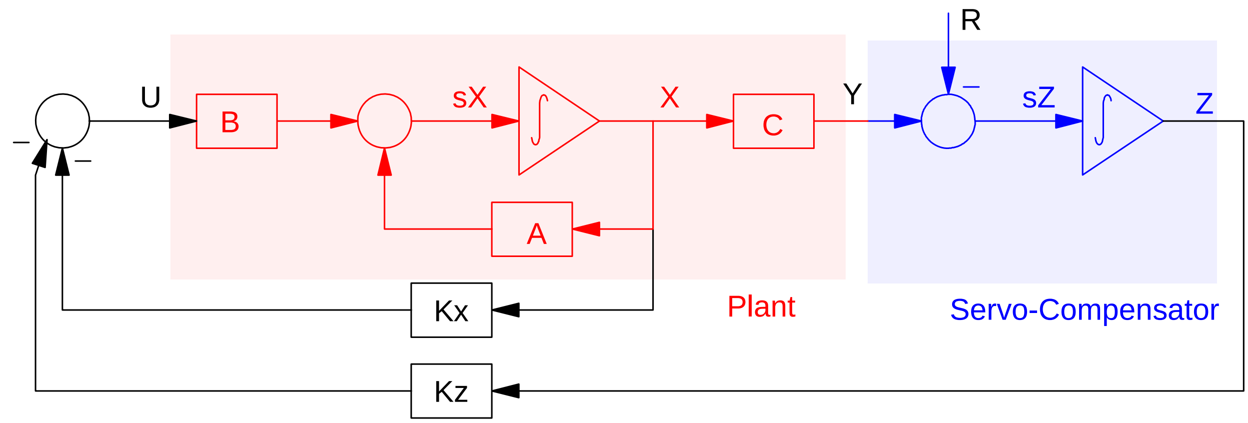
```
>> Kx = K5(1:4)

    2.0000    7.0000   13.0000   25.0000

>> Kz = K5(5)

   16.0000
```

Answer:

$$\boxed{\begin{array}{c} K_x = \left[\begin{array}{cccc} 2 & 7 & 13 & 25 \end{array}\right] \\ K_z = 16 \end{array}}$$

Just for fun, plot the step-response of the system with feedback along with the control input, U.

In Matlab, the closed-loop system is



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

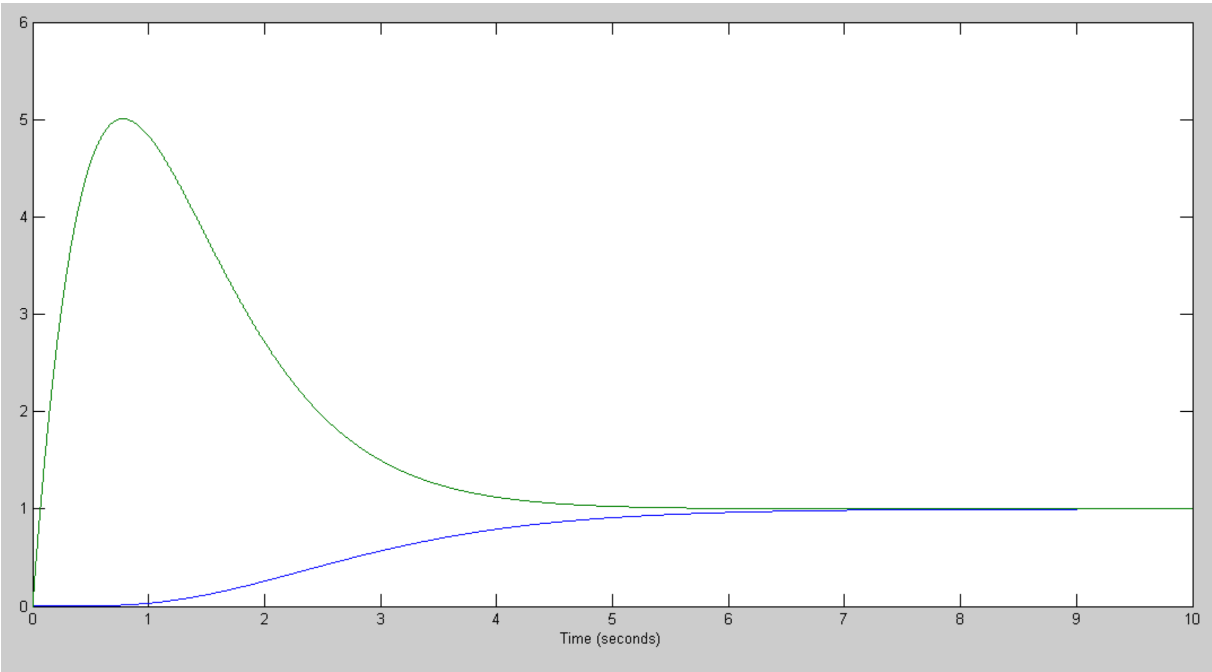
To plot the step response from R to both Y and U, define the output to be

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

In Matlab:

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

         0         0         0    1.0000         0
   -2.0000   -7.0000  -13.0000  -25.0000  -16.0000

>> D5 = [0; 0]
```

```
     0
    0

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



Step Response for a Servo-Compensator System with Closed-Loop Poles at { -1, -2, -2, -2, -2 }
Output = Blue, Input = Green. Note that the input goes to 1.000 - which is the voltage needed to hold the output at 1.00

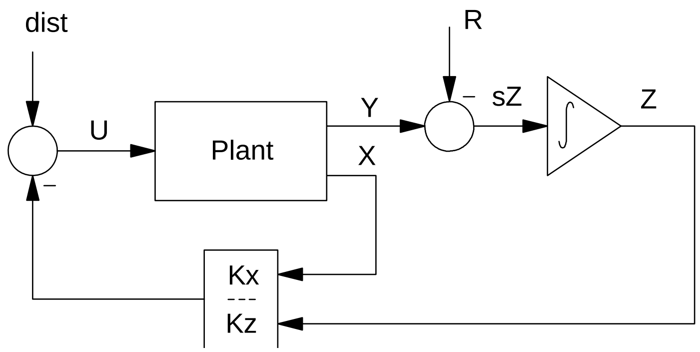
Constant Disturbance:

Problem: Suppose a system has a constant disturbance. Design a feedback control law that forces Y=R in spite of this disturbance.

Solution: In order to overcome the constant disturbance, the input, U, at steady-state again needs to be a constant. We may not know what that constant is, but it will be a constant. This problem then becomes one of estimating the constant which forces

$$E = R - Y = 0$$

This is exactly the same probelem as before and is the same solution. All that changes is the integrator provides an offset for U which drives the constant output as well as cancelling the disturbance.



Servo-Compensator Design for a Plant with a Constant Disturbance

Option 1: Include the disturbance in with the B matrix (since R is a step input).

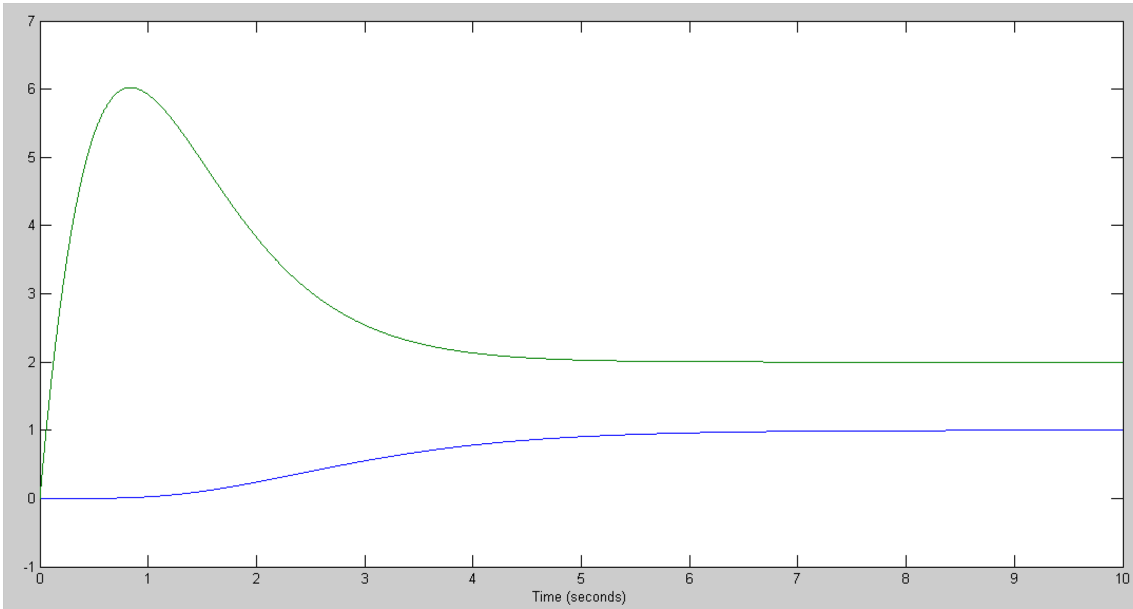
Example: Find the response when the input has a disturbance of -1:

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

Lump the disturbance in with R

$$\begin{bmatrix} 0 \\ -1 \end{bmatrix}1 + \begin{bmatrix} B \\ 0 \end{bmatrix}(-1) = \begin{bmatrix} -B \\ -1 \end{bmatrix}$$

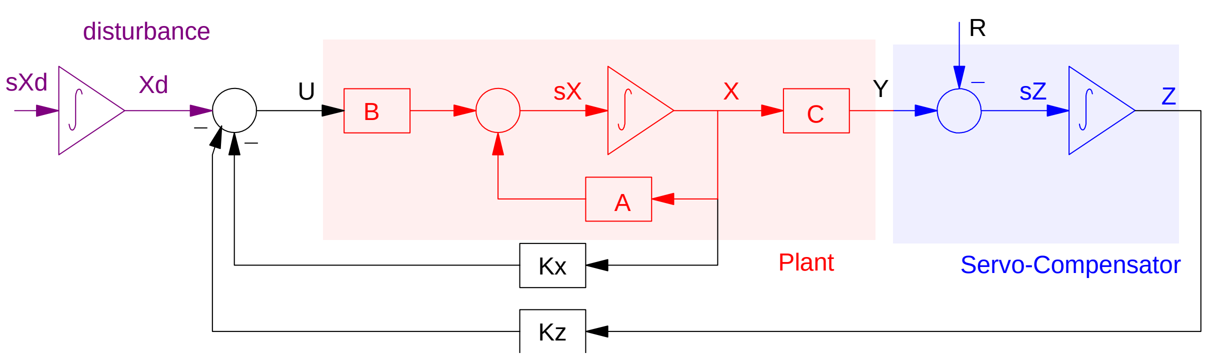
```
>> Br = [-B;-1]

Br =

    -1
     0
     0
     0
    -1

>> G5 = ss(A5 - B5*K5, Br, C5, D5);
>> y = step(G5,t);
>> plot(t,y)
>> xlabel('Time (seconds)');
```



Step Response for a Servo-Compensator System with A Disturbance of -1.
Note that the input goes to 2.000 in steady state: 1.00 to cancel the disturbance, another 1.0 to drive the output

Option 2: Augement the system with a disturbance state

The state-space model for the plant : servo compensator : disturbance becomes a little more complicated, however. To model a constant disturbance, add an integrator: the integration constant is the disturbance.



Block Diagram for a Plant plus Servo Compensator plus Constant Disturbance

In state-space, the open-loop system looks like:

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

or

$$sX_5 = A_5 X_5 + B_u U + B_r R$$

With feedback

$$U = -K_5 X_5$$

this becomes

$$sX_5 = (A_5 - B_u K_5) X_5 + B_r R$$

Augment the system with two integrator states:

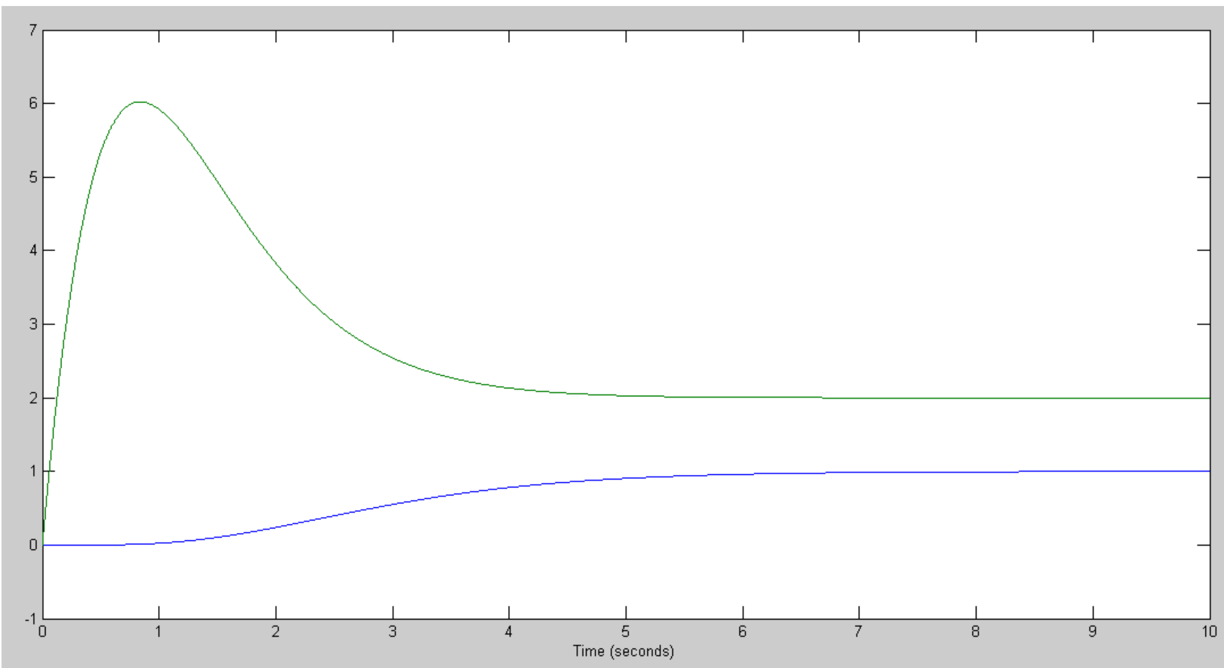
- Xd is the input disturbance
- Xr is the reference input (set point)

$$s\begin{bmatrix} X_5 \\ X_d \\ X_r \end{bmatrix} = \begin{bmatrix} A_5 - B_5 K_5 & B_u & B_r \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} X_5 \\ X_d \\ X_r \end{bmatrix} + \begin{bmatrix} X_5(0) \\ X_d(0) \\ X_r(0) \end{bmatrix} \delta(t)$$

This has no input but has an initial condition - coming from the delta input. To simulate, you use the MATLAB command *impulse()*

```
>> Bu = [B; 0]

     1
     0
     0
     0
     0

>> Br = [0;0;0;0;-1]

     0
     0
     0
     0
    -1

>> A7 = [A5-B5*K5, Bu, Br;  0,0,0,0,0,0,0; 0,0,0,0,0,0,0]

   -4.0000   -6.0000  -13.0000  -25.0000  -16.0000    1.0000         0
    1.0000   -2.0000    1.0000         0         0         0         0
         0    1.0000   -2.0000    1.0000         0         0         0
         0         0    1.0000   -1.0000         0         0         0
         0         0         0    1.0000         0         0   -1.0000
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
```

```
>> C7 = [C, 0, 0, 0; -K5, 0, 0]

         0         0          0    1.0000         0         0         0
   -2.0000   -7.0000   -13.0000  -25.0000  -16.0000         0         0

>> D7 = [0;0]

     0
     0

>> G7 = ss(A7, X70, C7, D7);
>> y = impulse(G7,t);
>> plot(t,y)
>>
```



Closed-Loop Response to Ref = +1 and an input disturbance of -1.

Note the the response is exactly the same as we computed before.