Servo-Compensators Generalized Case

NDSU ECE 463/663

Lecture #18

Inst: Jake Glower

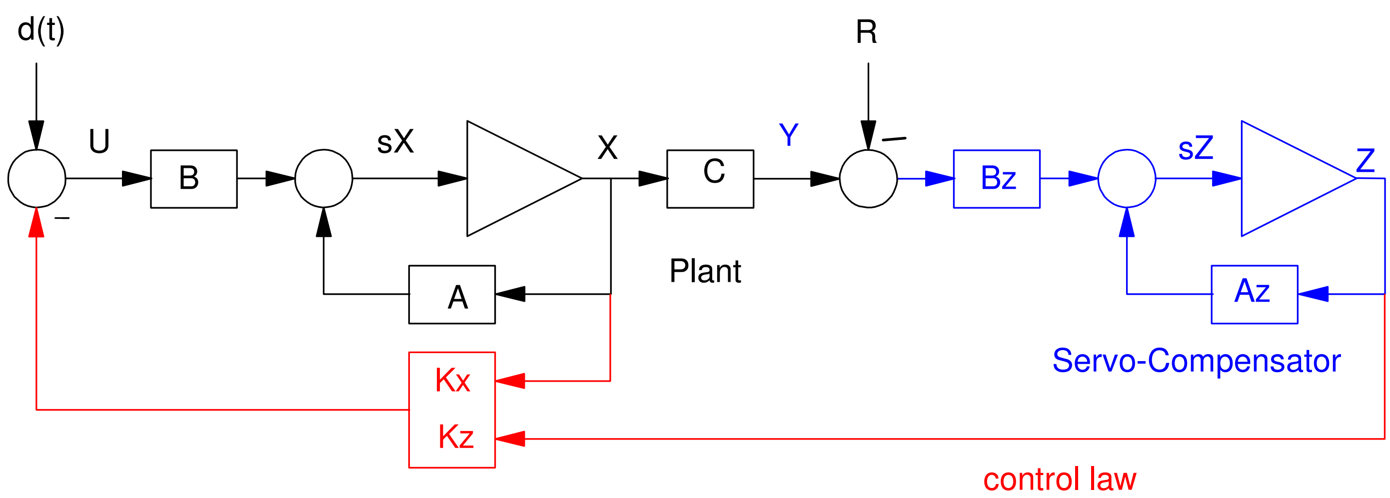
Please visit Bison Academy for corresponding lecture notes, homework sets, and solutions

Recap

- If tracking or rejecting a constant, add a servo compensator with a pole at s = 0
- If tracking or rejecting a sinusoid, add a servo compensator with poles at $\pm j\omega$

Not surprisingly,

- If tracking or rejecting signals with poles at $\{0,\ j\omega,\ -j\omega\}$, add a servo compensator with poles at $\{0,\ j\omega,\ -j\omega\}$.



State-Space Formulation: Let the plant be

$$sX = AX + BU + Bd$$

$$Y = CX$$

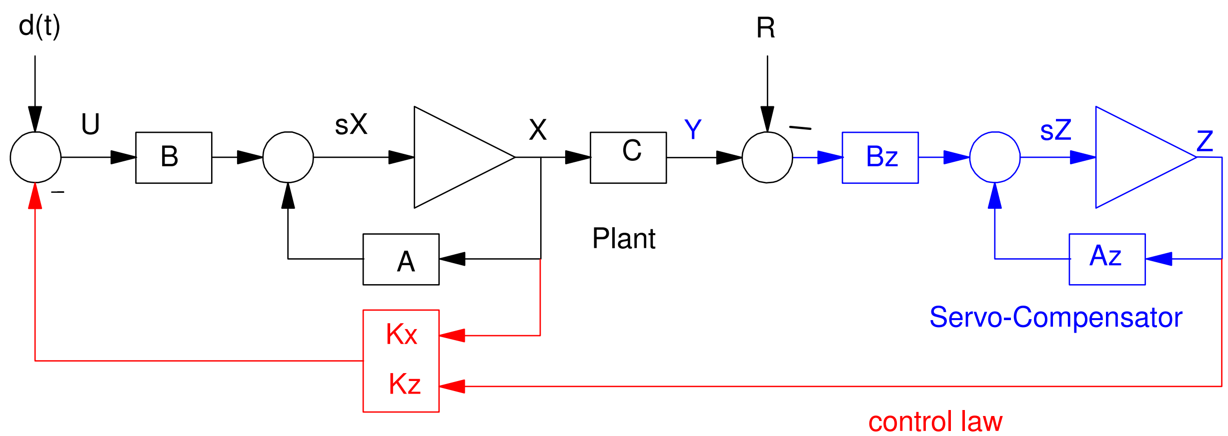
Define a servo-compensator

$$sZ = A_z Z + B_z (Y - R)$$

so that the eigenvalue of Az are

$$eig(A_z) = 0, \pm j\omega$$

Feed the servo-compensator with the difference between Y and the set point R



In state-space, the plant plus servo-compensator looks like the following:

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

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

or you can write this as

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

Example:

Assume a 4th-order heat equation:

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

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

Design a feedback control law for the following system so that

- The 2% settling time is 13 seconds,
- There is no overshoot for a step input,
- Y tracks a constant setpoint (R = 1), and
- Y rejects a sinusoidal disturbance at 1 rad/sec

Note: The result works for any combination of DC and 1 rad/sec:

$$R(t) = a_1 + b_1 \cos(t) + c_1 \sin(t)$$

$$d(t) = a_2 + b_2 \cos(t) + c_2 \sin(t)$$

Step 1: Add a servo compensator:

- Controllable
- Poles at $\{0, j, -j\}$

$$sZ = \left[\begin{array}{cc:c} 0 & 1 & 0 \\ -1 & 0 & 0 \\ \cdots & \cdots & \cdots \\ 0 & 0 & 0 \end{array}\right] Z + \left[\begin{array}{c} 1 \\ 1 \\ \cdots \\ 1 \end{array}\right] (R - Y)$$

Step 2: Create the augmented system: plant + servo compensator

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

$$s\begin{bmatrix} X \\ \cdots \\ Z \end{bmatrix} = \left[\begin{array}{cccc:ccc} -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right]\begin{bmatrix} X \\ \cdots \\ Z \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \cdots \\ 0 \\ 0 \\ 0 \end{bmatrix} U + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \cdots \\ -1 \\ -1 \\ -1 \end{bmatrix} R + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \cdots \\ 0 \\ 0 \\ 0 \end{bmatrix} d$$

Design a full-state feedback control law to meet the design specs.

- Dominant pole at s = -1
- Place the poles at {-1, -2, -2.2, -2.3, -2.4, -0.3+j, -0.3-j} using Bass Gura

In Matlab:

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

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

B = [1;0;0;0]

     1
     0
     0
     0

C = [0,0,0,1]

     0     0     0     1
```

```
Az = [0,1,0;-1,0,0;0,0,0]

     0      1      0
    -1      0      0
     0      0      0

Bz = [1;1;1]

     1
     1
     1

A7 = [A, zeros(4,3) ; Bz*C, Az]

    -2      1      0      0  :   0      0      0
     1     -2      1      0  :   0      0      0
     0      1     -2      1  :   0      0      0
     0      0      1     -1  :   0      0      0
    -----------------------------------------
     0      0      0      1  :   0      1      0
     0      0      0      1  :  -1      0      0
     0      0      0      1  :   0      0      0
```

```
B7u = [B ; zeros(3,1)]

      1
      0
      0
      0
  - - - - -
      0
      0
      0


K7 = ppl(A7, B7u, [-1, -2, -2.2, -2.3, -2.4, -0.3+j, -0.3-j])

    3.5000   12.0900   29.6810   63.4358    0.5236   21.3719   26.4739
```

This gives

Kx = [3.5000   12.0900   29.6810   63.4358]

Kz = [0.5236   21.3719   26.4739]

Step Responses

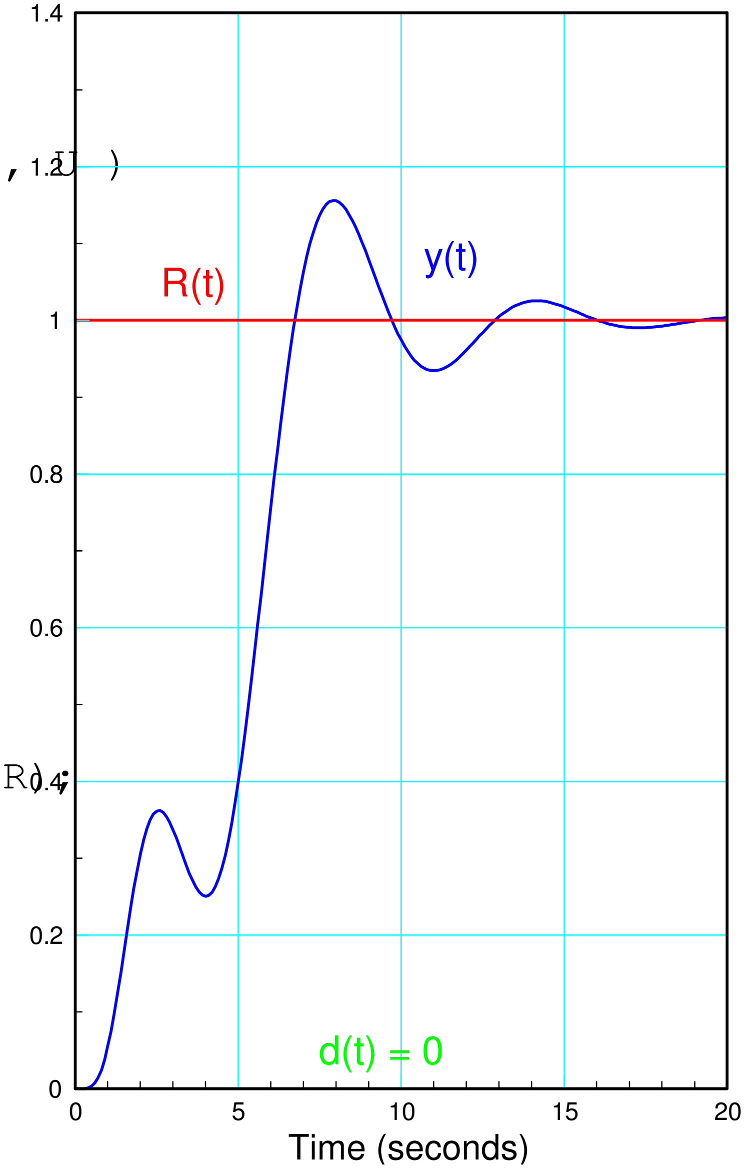
- Use the *step2* command from before

```
function [ y ] = step3( A, B, C, D, t, X0, U )
```

Case 1: Step Response with Respect to R

- The system tracks a constant set point:

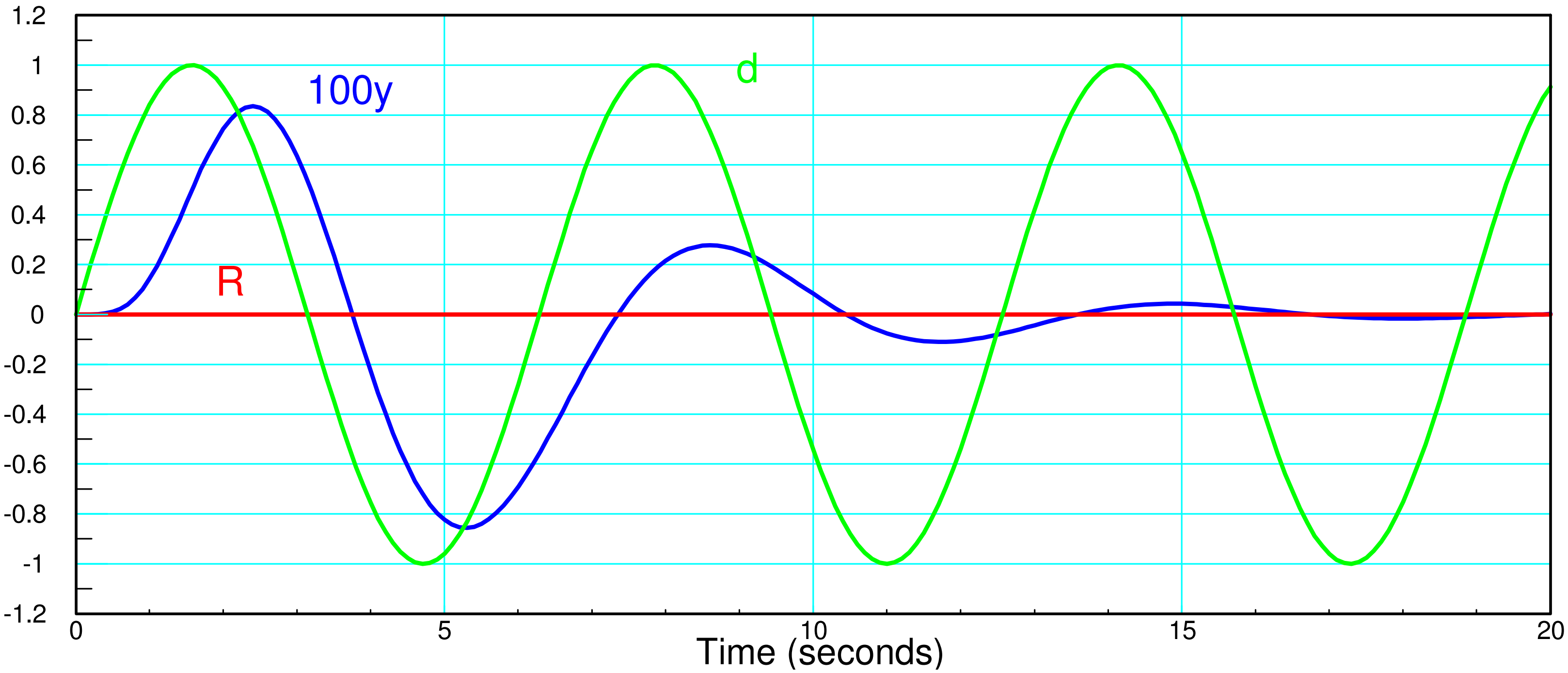
```
B7r = [0*B ; -Bz];
C7 = [C, zeros(1,3)];
D7 = 0;
X0 = zeros(7,1);
t = [0:0.1:20]';
R = 0*t + 1;
y = step3(A7-B7u*K7, B7r, C7, D7, t, X0, R);
plot(t,R,'r',t,y,'b')
```



Case 2: Response to disturbance, d

- Rejects a 1 rad/sec disturbance

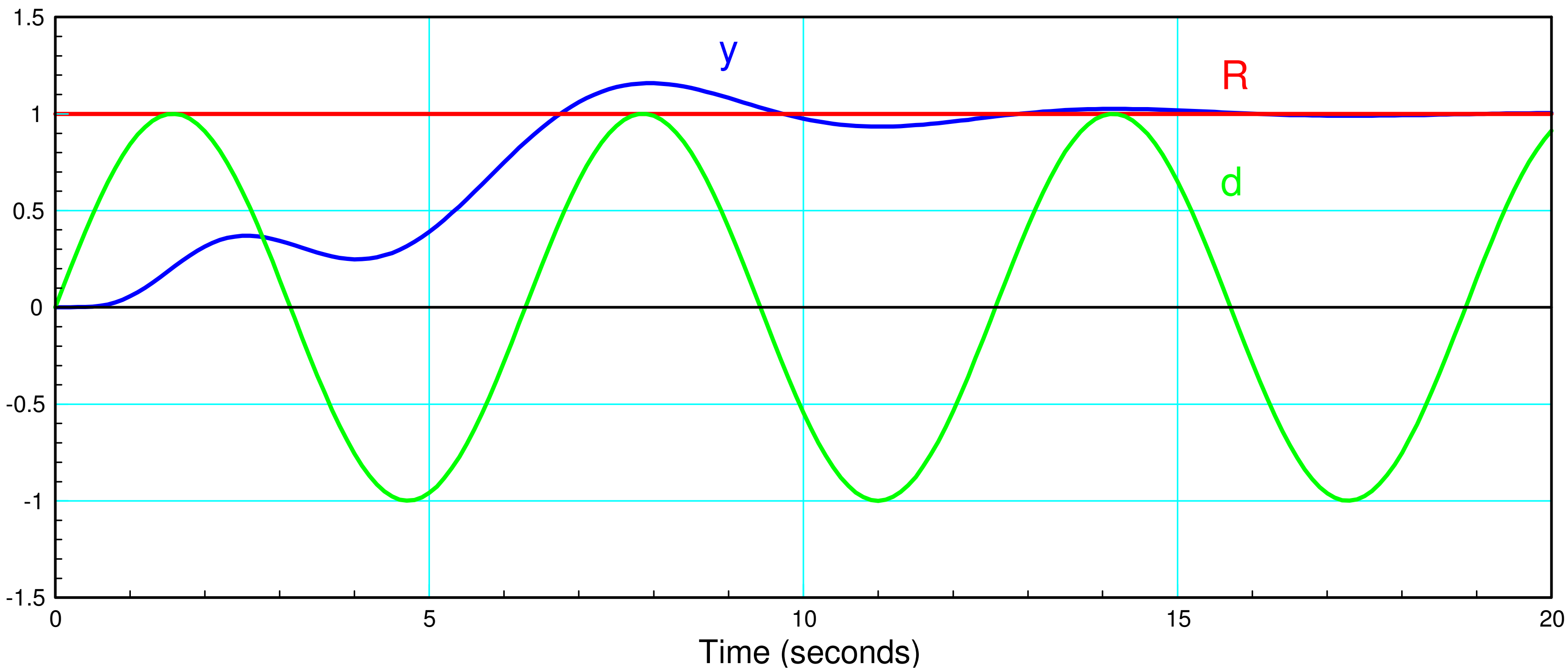
```
B7d = [B ; 0*Bz];
R = 0*t;
d = sin(t);
y = step3(A7-B7u*K7, B7d, C7, D7, t, X0, d);
plot(t,R,'r',t,y,'b',t,d,'g')
```



Case 3: Constant Set Point & 1 rad/sec Noise

- Tracks a constant set point, rejects a 1 rad/sec disturbance

```
d = sin(t);
R = 0*t + 1;
y = step3(A7-B7u*K7, [B7r, B7d], C7, [0,0], t, X0, [R,d]);
plot(t,R,'r',t,y,'b',t,d,'g')
```

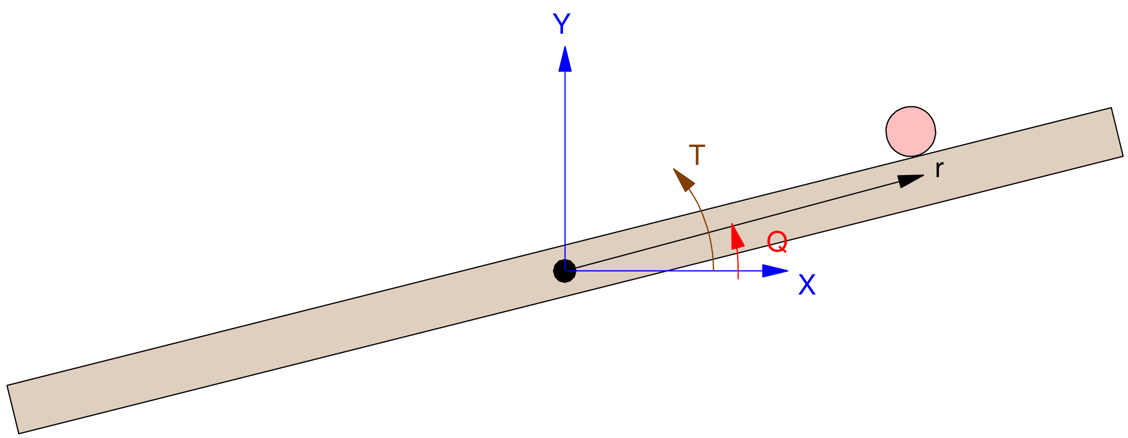


Example 2: Ball and Beam System

- m = 1.1kg (0.1kg more than than the model)
- R(t) = 1.0 - 0.4 cos(t)

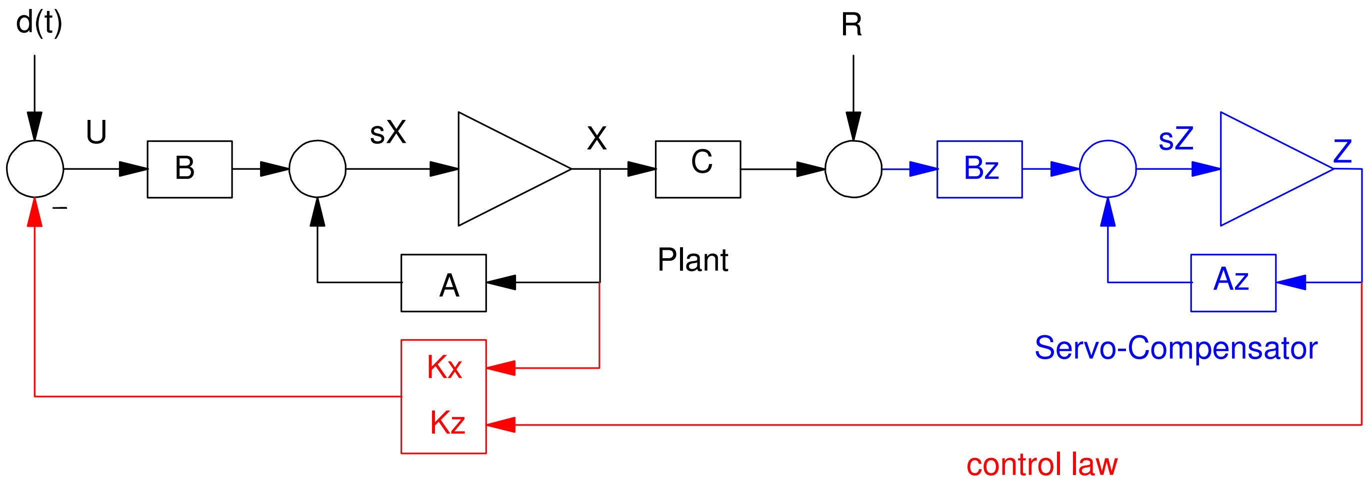
Extra mass creates

- A constant disturbance due to gravity, plus
- A 1 rad/sec disturbance



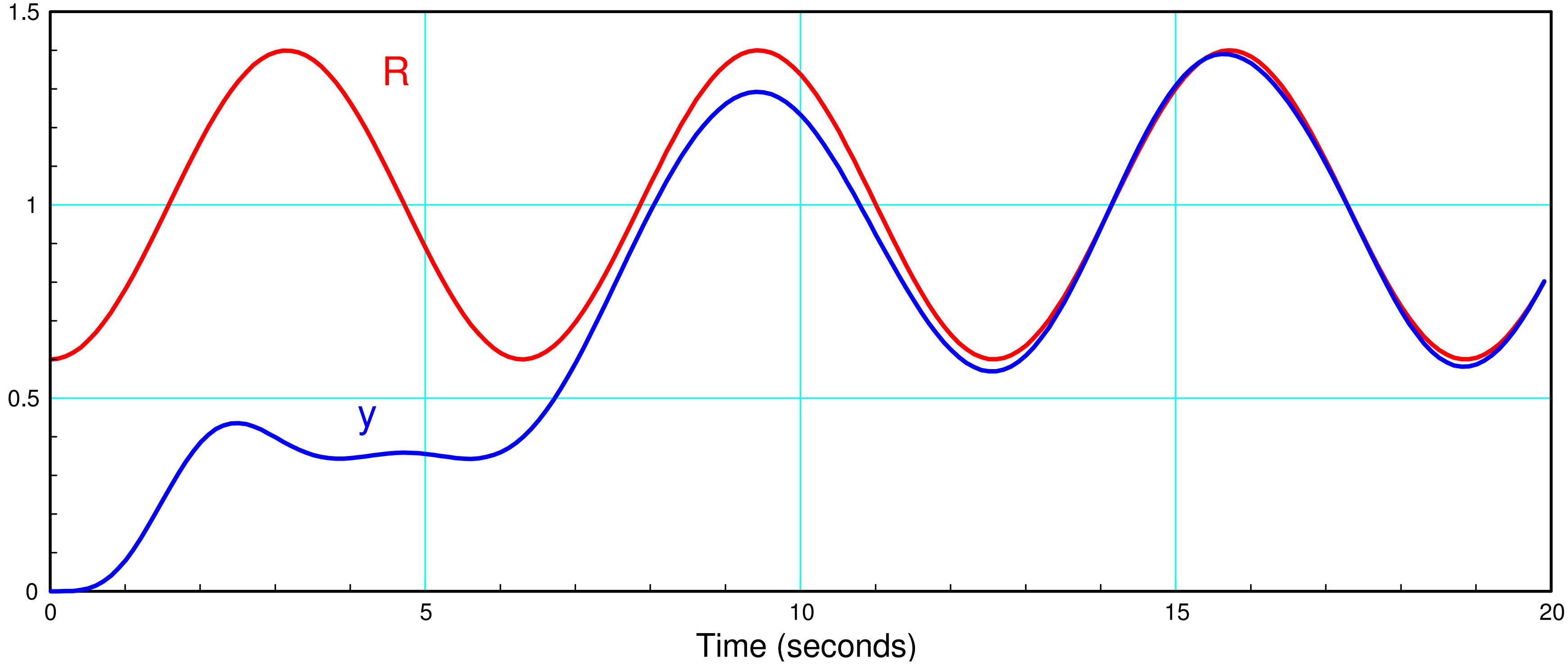
Same solution as before

- Add a servo compensator
- Pick Az to have poles at {0, -j, +j}
- Pick Kx and Kz to place the closed-loop poles



Result: m = 1.1kg

- Tracks a constant & 1 rad/sec sine wave
- Rejects disturbances at DC and 1 rad/sec

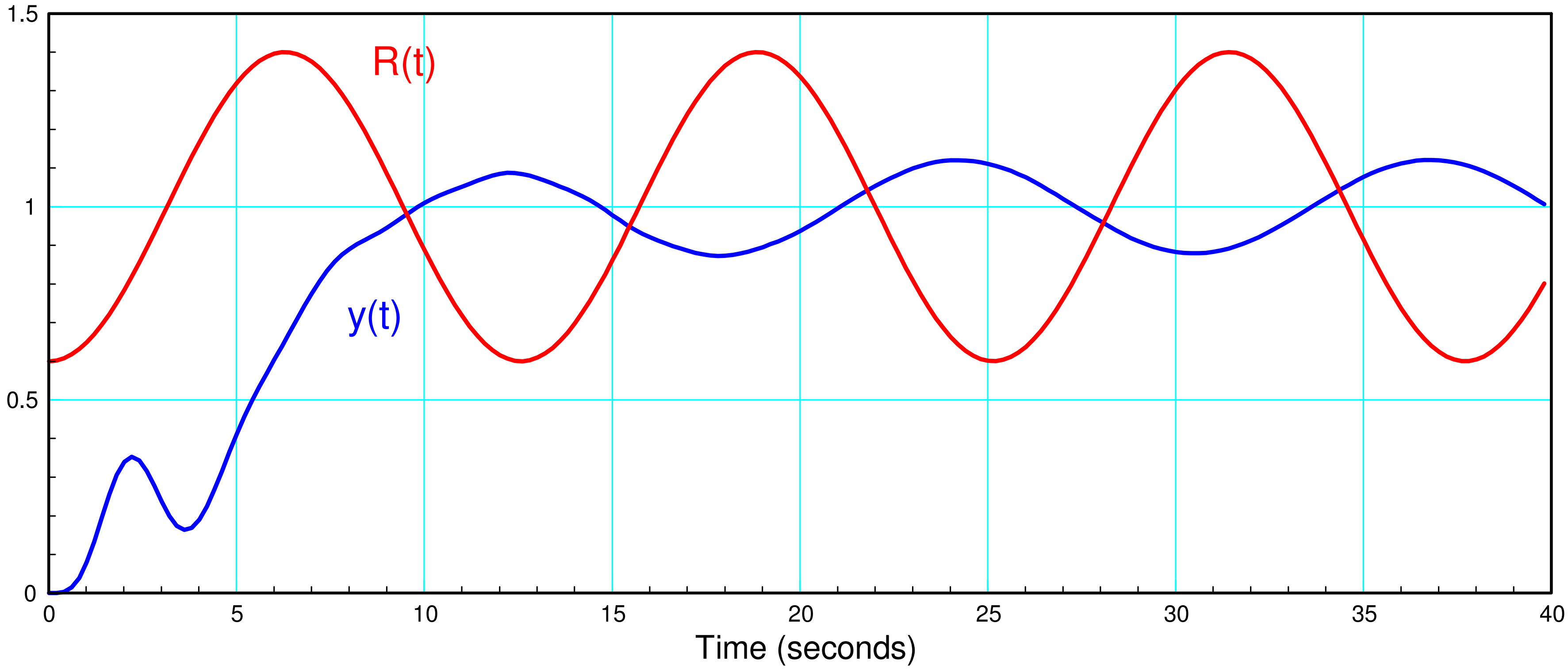


Sidelight: What is the frequency changes?

- Change the frequency to 0.5 rad/sec

No longer tracks

- Not designed for 0.5 rad/sec



What to do?

Option 1:

- Determine the frequency of R(t)
- Redesign the servo compensator

Option 2:

- Pick Az to have poles at {0, j0.5, j, j1.5, j2}
- Results in a 9th-order compensator

Option 3: Adaptive Control

- Estimate the frequency of R(t) in real time
- Change Az, Kx, Kz accordingly

Option 4: Adaptive Control (Self-Tuning Regulator)

- Estimate the next value of R(t) in real time
- Determine the input to drive y(t) to R(t) in real time

There are more options...

Summary

If you are trying to track a constant and/or reject a constant disturbance

- Add a servo compensator with a pole at s = 0.

If you are trying to track and/or reject a sinusoidal

- Add a servo compensator with poles at {+jw, -jw}

If you are trying to do both

- Add a servo compensator with poles at {0, +jw, -jw}

This can result in a very high order compensator

- The step response can be pretty squirrelly as the compensator tries to figure out what you're trying to track