ECE 461 - Homework #11

Discre-Time Compensator Design. Due Monday, November 23rd

Each problem is 20 points

The transfer function for a system is

$$G(s) = \left(\frac{625}{(s+1.31)(s+5.71)(s+12.45)(s+18.37)} \right)$$

(heat equation from Homework #5 and #10)

Assume a sampling rate of T = 0.1 second.

1) Design a discrete-time compensator of the form

K(z) = k

which results in

- 20% overshoot for a step input.

Check your design in VisSim

For 20% overshoot, the solution lies along the line

s = -1 + j2

Doing a numerical search

$$\left(\left(\frac{625}{(s+1.31)(s+5.71)(s+12.45)(s+18.37)} \right) (e^{-sT/2})(k) \right)_{s=\alpha(-1+j2)} = 1\angle 180^0$$

s = -1.6197 + j3.2395

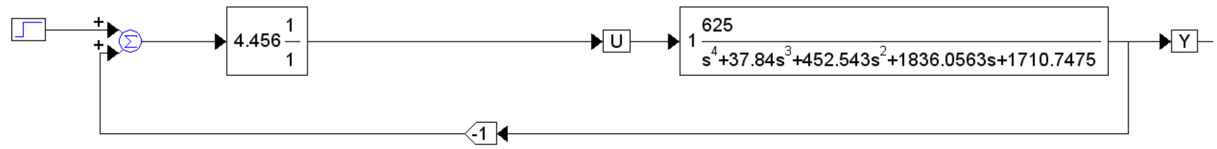
z = 0.5767 + j0.4365

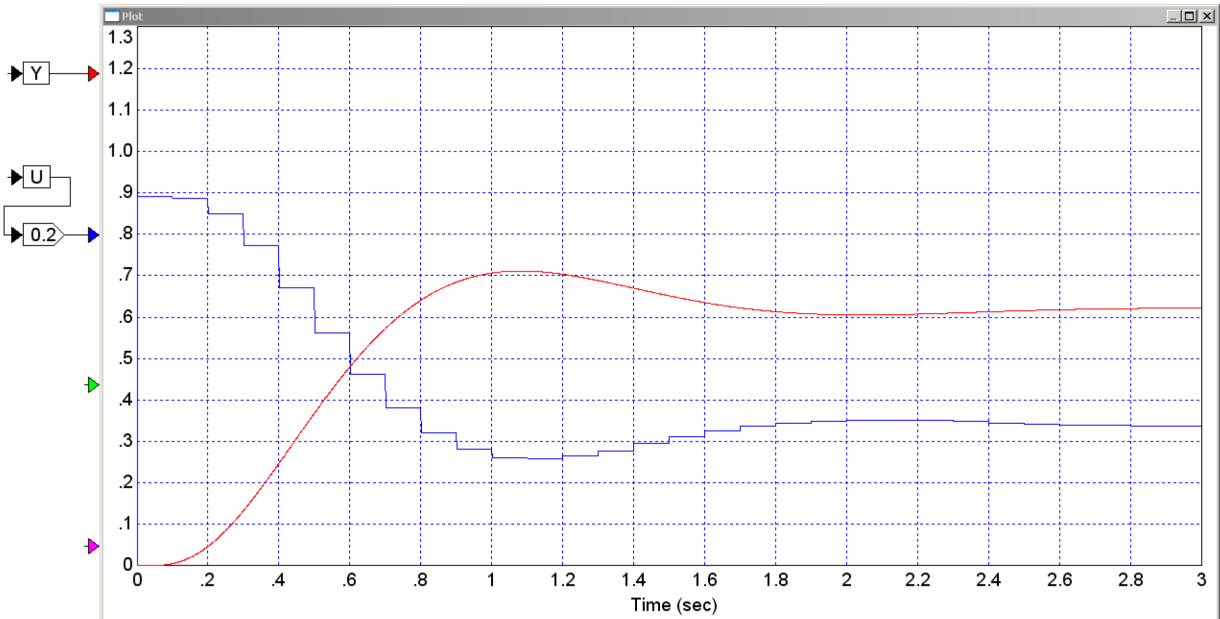
At this point

$$\left(\left(\frac{625}{(s+1.31)(s+5.71)(s+12.45)(s+18.37)} \right) (e^{-sT/2}) \right)_{s=-1.6197+j3.2395} = 0.2244\angle 180^0$$

so

$$k = \frac{1}{0.2244} = 4.4560$$


+
+
4.456
1
1
U
1
625
s4+37.84s3+452.543s2+1836.0563s+1710.7475
Y
-1


Plot
Y
U
0.2
1.3
1.2
1.1
1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
0
0
.2
.4
.6
.8
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
Time (sec)

2) Design a discrete-time PI compensator of the form

$$K(z) = k\left(\frac{z-a}{z-1}\right)$$

which results in

- No error for a stp input and
- 20% overshoot for a step input.

Check your design in VisSim

Pick 'a' to cancel the pole at

s = -1.31

z = e^{sT} = 0.8772

$$K(z) = k\left(\frac{z-0.8772}{z-1}\right)$$

Find the point where

$$\left(\left(\frac{625}{(s+1.31)(s+5.71)(s+12.45)(s+18.37)}\right)(e^{-sT/2})\left(k\left(\frac{z-0.8772}{z-1}\right)\right)\right)_{s=\alpha(-1+j2)} = 1\angle 180^0$$

Iterating

s = -1.4659 + j2.9318

z = 0.6213 + j0.4127

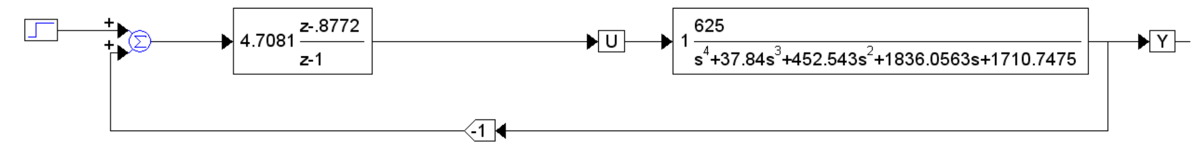
At this point

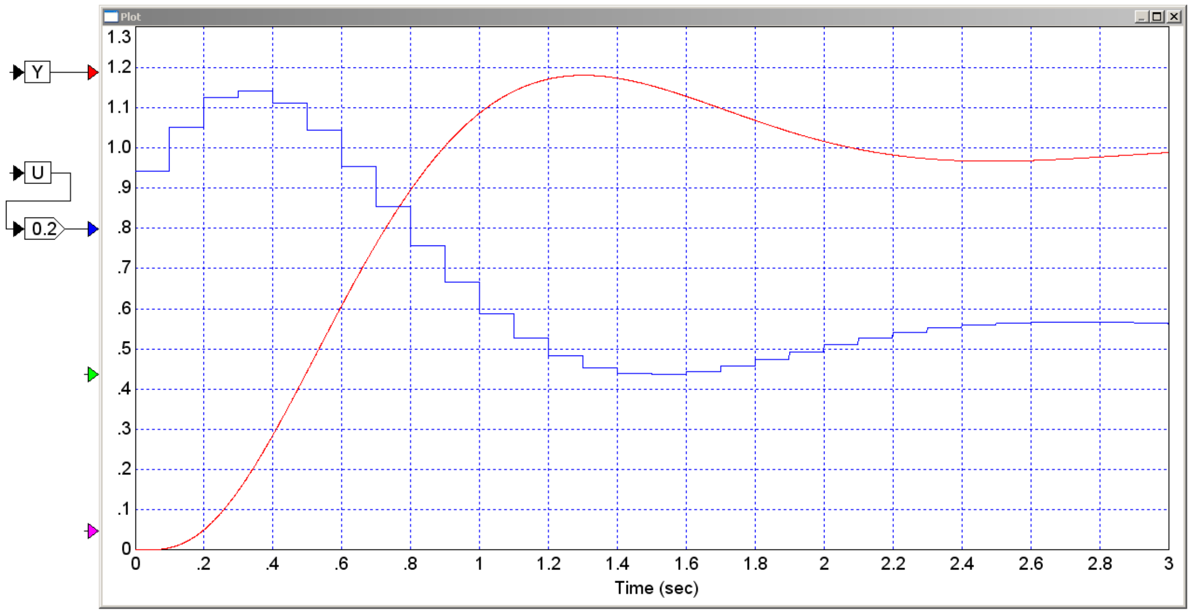
$$\left(\left(\frac{625}{(s+1.31)(s+5.71)(s+12.45)(s+18.37)}\right)(e^{-sT/2})\left(\frac{z-0.8772}{z-1}\right)\right)_{s=-1.4659+j2.9318} = -0.2124$$

so

$$k = \frac{1}{0.2124} = 4.7081$$

Checking in SciLab





The input peaks at only 2x its steady-state value. This means you need to size the motor a little more than what's required for steady-state operation. It also suggests you can speed up the system a little more.

3) Design a discrete-time compensator K(z) which results in

- No error for a stp input and
- 20% overshoot for a step input.
- A 2% settling time of 1 second

Check your design in VisSim

Translating

- Add a pole at s = 0 to make it type-1
- Place the dominant pole at s = -4 + j8

or in the z-plane

- Add a pole at z = +1 to make it type-1
- Place the dominant pole at z = 0.4670 + j0.4809

Start with cancelling two poles and see if that works

$$K(z) = \left(\frac{(z-0.8772)(z-0.5650)}{(z-1)(z-a)}\right)$$

Evaluating at the design point:

$$\left(\left(\frac{625}{(s+1.31)(s+5.71)(s+12.45)(s+18.37)}\right)(e^{-sT/2})\left(\frac{(z-0.8772)(z-0.5650)}{(z-1)}\right)\right)_{s=-4+j8} = 0.0255\angle -187.93^0$$

The phase is past 180 degrees - so it won't work. Try cancelling another pole

$$K(z) = \left(\frac{(z-0.8772)(z-0.5650)(z-0.2879)}{(z-1)(z-a)^2}\right)$$

Evaluating at the design point:

$$\left(\left(\frac{625}{(s+1.31)(s+5.71)(s+12.45)(s+18.37)}\right)(e^{-sT/2})\left(\frac{(z-0.8772)(z-0.5650)(z-0.2879)}{(z-1)}\right)\right)_{s=-4+j8} = 0.0128\angle -118^0$$

Since three zeros are added, add three poles (one at z = +1, the others at....)

The angle 61.63 degrees away from 180 degrees

Each of the two poles at 'a' add 30.81 degrees

$$a = 0.4670 - \left(\frac{0.4809}{\tan\left(30.81^0\right)}\right) = -0.3392$$

so

$$K(z) = \left(\frac{(z-0.8772)(z-0.5650)(z-0.2879)}{(z-1)(z+0.3392)^2}\right)$$

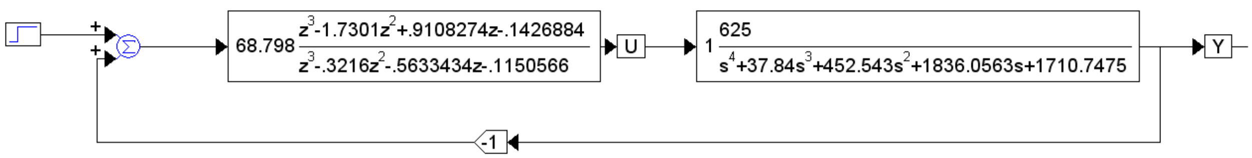
To find k

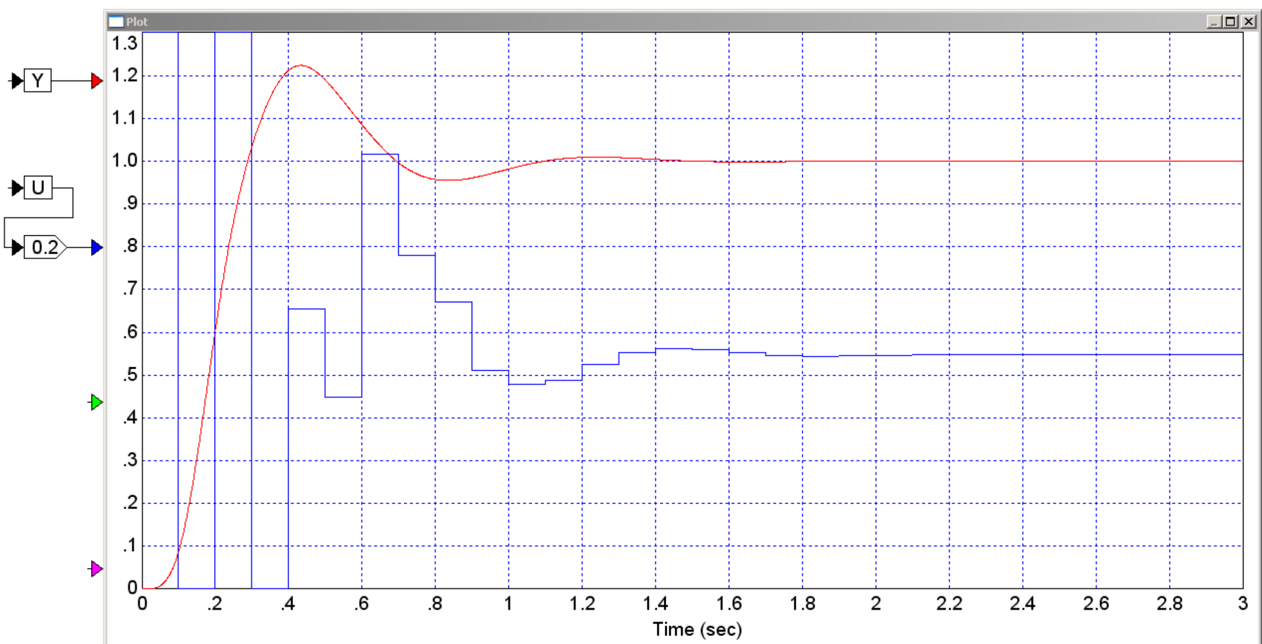
$$\left(\left(\frac{625}{(s+1.31)(s+5.71)(s+12.45)(s+18.37)}\right)\left(e^{-sT/2}\right)\left(\frac{(z-0.8772)(z-0.5650)(z-0.2879)}{(z-1)(z+0.3392)^2}\right)\right)_{s=-4+j8} = -0.0145$$

$$k = \frac{1}{0.0145} = 68.798$$

$$K(z) = 68.798\left(\frac{(z-0.8772)(z-0.5650)(z-0.2879)}{(z-1)(z+0.3392)^2}\right)$$

Checking in VisSim





Note that to speed up the system 3x (3 seconds in problem 2 to 1 second in problem 3) the input went from a peak of

- 5.5 in problem #2 (1.1 x 5) to
- 344 in problem #3 (68 x 5)

(it took 62 times more input for a factor of 3 increase in speed).

You can make a system faster than it's open-loop step response - but at a high cost.

A settling time of 1 second works on paper but probably won't work in practice.

4) Write a program to implement the compensator for problem #3

```
while(1) {
   x3 = x2;
   x2 = x1;
   x1 = x0;
   x0 = A2D_Read(0);

   y3 = y2;
   y2 = y1;
   y1 = y0;
   y0 = 0.3216*y1 + 0.5633434*y2 + 0.1150566*y3 +
        68.798*(x0  -1.7301*x1 + 0.9108274*x2  - 0.1426884*x3);

   D2A(y0);

   Wait_100ms();
   }
```