Homework #10: ECE 461/661

z-Transforms, s to z conversion, Root Locus in the z-Domain

z-Transforms

1) Determine the difference equation that relates X and Y

$$Y = \left(\frac{0.005z}{(z-0.98)(z-0.95)(z-0.8)}\right)X$$

n Matlab

```
>> poly([0.98,0.95,0.8])

ans =    1.0000   -2.7300    2.4750   -0.7448
```

$$Y = \left(\frac{0.005z}{z^3-2.73z^2+2.475z-0.7448}\right)X$$

Cross multiply

$$(z^3 - 2.73z^2 + 2.475z - 0.7448)Y = (0.005z)X$$

'zY' means 'y(k+1)' or 'the next value of Y'

y(k+3) - 2.73 y(k+2) + 2.475 y(k+1) - 0.7448 y(k) = 0.005 x(k+1)

If you don't like using future values, you can do a change of variable and shift time by 3

y(k) - 2.73 y(k-1) + 2.475 y(k-2) - 0.7448 y(k-3) = 0.005 x(k-2)

Either answer is correct

2) Determine y(k) assuming

$$Y = \left(\frac{0.005z}{(z-0.98)(z-0.95)(z-0.8)}\right)X \qquad x(t) = 2\cos(3t) + 4\sin(3t)$$

$$T = 0.1$$

This is a phasor problem: $-\infty < t < \infty$

$$s = j3$$

$$z = e^{sT} = e^{j0.3} = 1\angle 0.3 \text{ rad}$$

$$X = 2 - j4 \qquad \textit{real = cosine, -imag = sine}$$

$$Y = \left(\frac{0.005z}{(z-0.98)(z-0.95)(z-0.8)}\right)_{z=1\angle 0.3}(2-j4)$$

$$Y = 0.2894 + j0.7072$$

which means (real = cosine, -imag = sine)

$$\boxed{y(t) = 0.2894\cos(3t) - 0.7072\sin(3t)}$$

3) Determine y(k) assuming

$$Y = \left(\frac{0.005z}{(z-0.98)(z-0.95)(z-0.8)}\right)X \qquad\qquad x(k) = u(k)$$

Plug in the z-transform for x(k)

$$Y = \left(\frac{0.005z}{(z-0.98)(z-0.95)(z-0.8)}\right)\left(\frac{z}{z-1}\right)$$

Do partial fractions

$$Y = \left(\frac{0.005z}{(z-1)(z-0.98)(z-0.95)(z-0.8)}\right)z$$

$$Y = \left(\left(\frac{25}{z-1}\right) + \left(\frac{-45.3704}{z-0.98}\right) + \left(\frac{21.1111}{z-0.95}\right) + \left(\frac{-0.7407}{z-0.8}\right)\right)z$$

$$Y = \left(\frac{25z}{z-1}\right) + \left(\frac{-45.3704z}{z-0.98}\right) + \left(\frac{21.1111z}{z-0.95}\right) + \left(\frac{-0.7407z}{z-0.8}\right)$$

Convert back to time

$$\boxed{y(k) = \left(25 - 45.3704(0.98)^k + 21.1111(0.95)^k - 0.7407(0.8)^k\right)u(k)}$$

s to z conversion

4) Determine the discrete-time equivalent of G(s). Assume T = 0.5 second

$$G(s) = \left(\frac{170}{(s+0.47)(s+3.40)(s+9.00)(s+16.77)}\right)$$

s and z are related by

$$z = e^{sT}$$

s = -0.47	z = 0.7906
s = -3.40	z = 0.1827
s = -9.00	z = 0.0111
s = -16.77	z = 0.0002

meaning

$$G(z) \approx \left(\frac{kz^n}{(z-0.7906)(z-0.1827)(z-0.0111)(z-0.0002)}\right)$$

To find 'k', math the gain at s = 0

$$G(s=0) = 0.7048$$

$$G(z=1) = 5.9098k$$

$$k = \left(\frac{0.7048}{5.9098}\right) = 0.1193$$

To find 'n', pick a frequency, such as 0.1 rad/sec and match the phase

$$G(s=j0.1) = 0.6892\angle-14.6744^0$$

$$G(z=e^{sT}) = (0.6894\angle-22.7688^0)\cdot(1\angle n\cdot 2.8648^0)$$

To match the phase

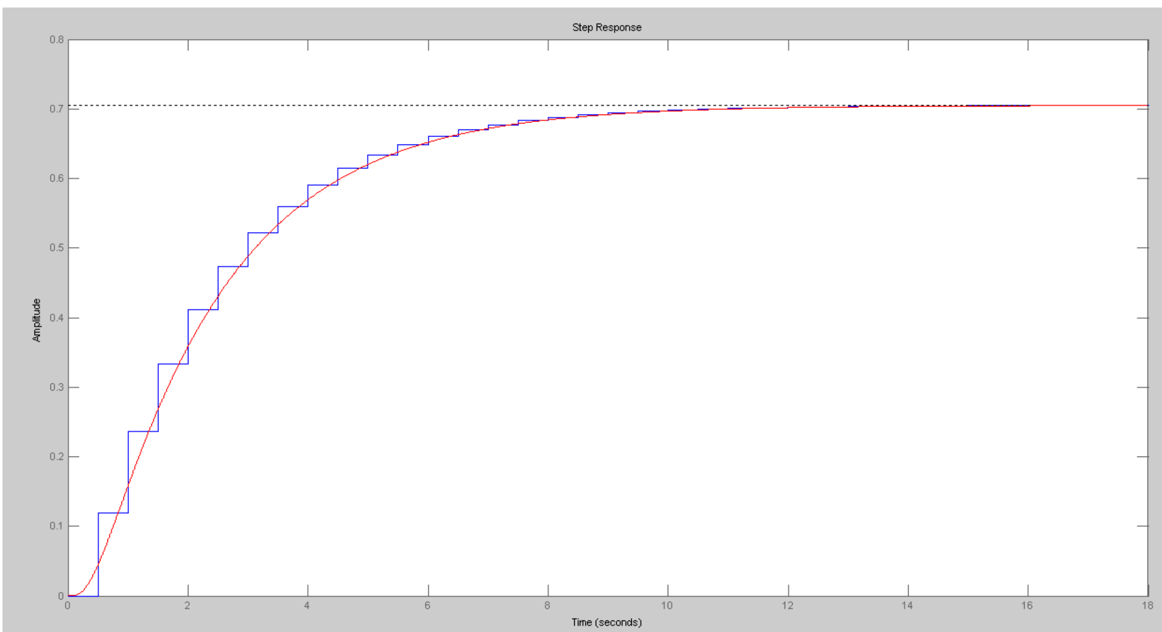
$$n = \left(\frac{22.7688^0-14.6744^0}{2.8648^0}\right) = 2.82$$

Round to n=3

$$\boxed{G(z) \approx \left(\frac{0.1193z^3}{(z-0.7906)(z-0.1827)(z-0.0111)(z-0.0002)}\right)}$$

Checking in Matlab

```
>> T = 0.5;
>> Gz = zpk([0,0,0],[0.7906,0.1827,0.0111,0.0002],0.1193,T)

                 0.1193 z^3
-------------------------------------------
(z-0.7906) (z-0.1827) (z-0.0111) (z-0.0002)

Sampling time (seconds): 0.5

>> step(Gz)
>> hold on;
>> Gs = zpk([],[-0.47,-3.40,-9,-16.77],170)

               170
--------------------------------
(s+0.47) (s+3.4) (s+9) (s+16.77)

>> t = [0:0.01:18]';
>> Ys = step(Gs,t);
>> plot(t,Ys,'r');
>>
```



5) Determine the discrete-time equivalent of G(s). Assume T = 0.1 second

$$G(s) = \left(\frac{170}{(s+0.47)(s+3.40)(s+9.00)(s+16.77)}\right)$$

In Matlab

```
>> s = [-0.47,-3.40,-9,-16.77]

s =   -0.4700   -3.4000   -9.0000  -16.7700

>> T = 0.1;
>> z = exp(s*T)

z =    0.9541    0.7118    0.4066    0.1869


>> Gs = zpk([],[-0.47,-3.40,-9,-16.77],170)

              170
--------------------------------
(s+0.47) (s+3.4) (s+9) (s+16.77)

>> Gz = zpk([],exp(s*T),1)

                     1
-------------------------------------------
(s-0.9541) (s-0.7118) (s-0.4066) (s-0.1869)
```

Find k to match the DC gain

```
>> k = evalfr(Gs,0) / evalfr(Gz,1)

k =   0.004500520667663
```

Add n zeros at z=0 to match the phase at s = j0.1 rad/sec (somewhat arbitrary)

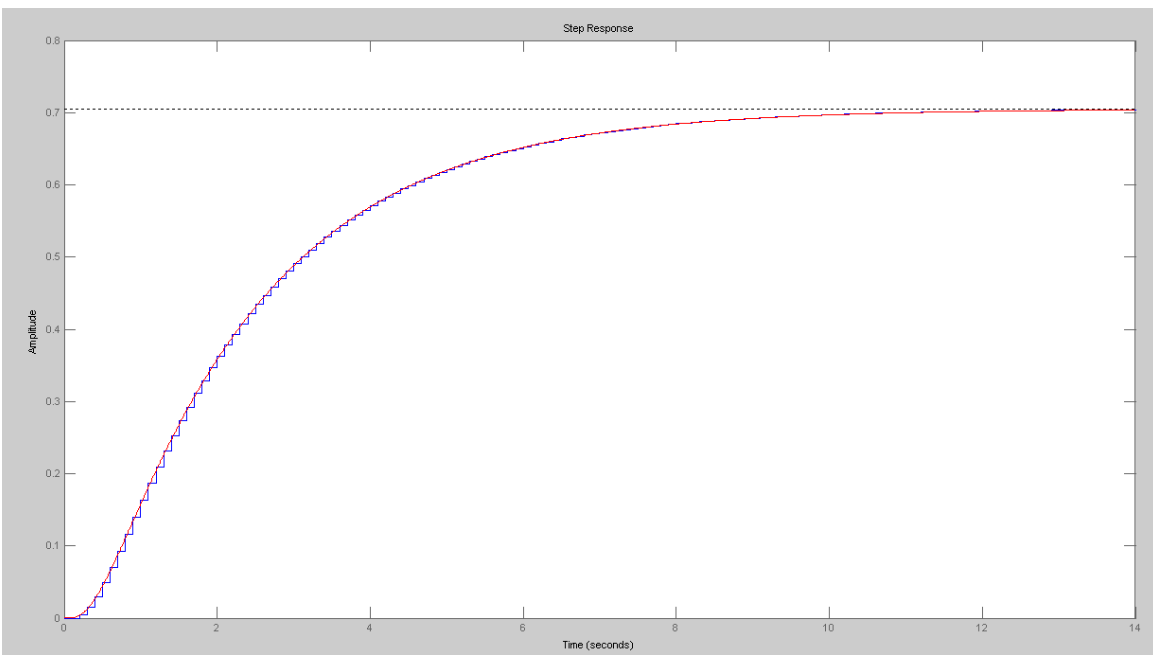
```
>> s = j*0.1;
>> T = 0.1;
>> z = exp(s*T);

>> n = (angle(evalfr(Gs,s)) - angle(evalfr(Gz,z))) / angle(z)

n =    2.2398
```

Round to n = 2

```
>> Gz = zpk([0,0],exp(s*T),k)


               0.004500 z^2
-------------------------------------------
(z-0.9541) (z-0.7118) (z-0.4066) (z-0.1869)
```

Checking in Matlab

```
>> Gs = zpk([],[-0.47,-3.40,-9,-16.77],170)

              170
--------------------------------
(s+0.47) (s+3.4) (s+9) (s+16.77)

>> Gz = zpk([0,0],[0.9541,0.7118,0.4066,0.1869],0.0045,T);
>> T = 0.1;
>> Gz = zpk([0,0],[0.9541,0.7118,0.4066,0.1869],0.0045,T)

                0.0045 z^2
-------------------------------------------
(z-0.9541) (z-0.7118) (z-0.4066) (z-0.1869)

Sampling time (seconds): 0.1

>> hold off;
>> step(Gz)
>> hold on
>> t = [0:0.01:14]';
>> Ys = step(Gs,t);
>> plot(t,Ys,'r');
>>
```



Note: If you change the sampling rate, you change G(z). That affects everything from this point onward (compensator design, gain computations, etc.). Changing the sampling rate is a *big* deal.

Root Locus in the z-Domain

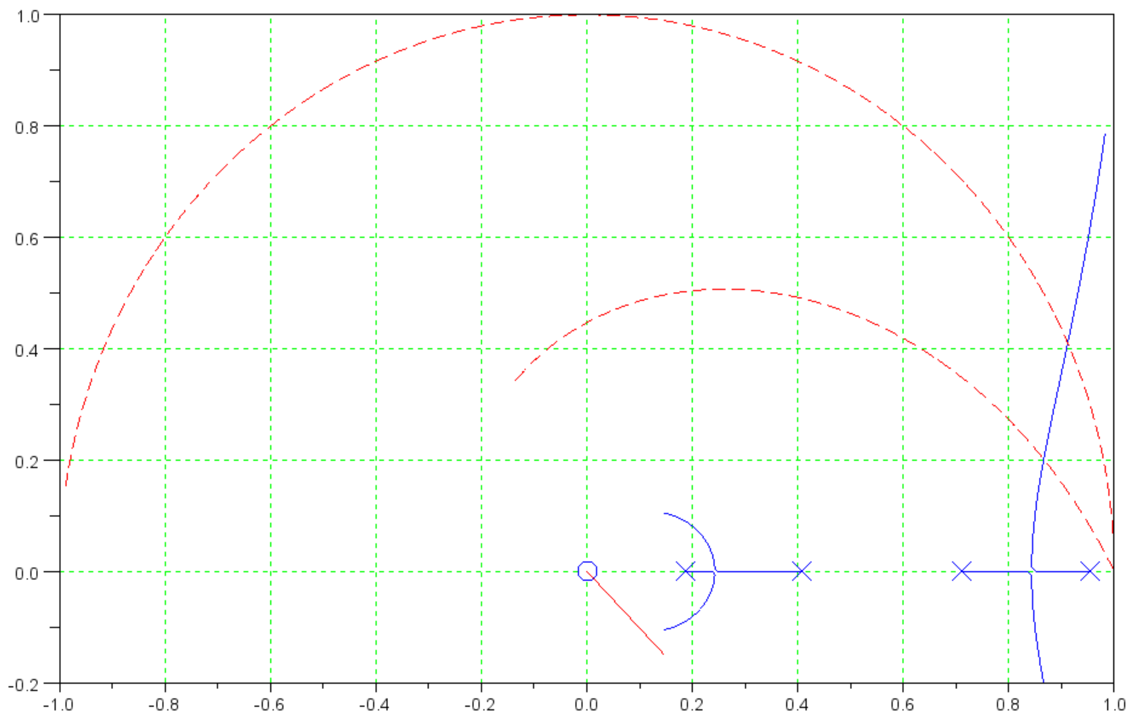
Assume T = 0.1 seconds.

$$G(s) = \left(\frac{170}{(s+0.47)(s+3.40)(s+9.00)(s+16.77)}\right)$$

6) Draw the root locus for G(z)

From problem #4

$$G(z) = \left(\frac{0.0045z^2}{(z-0.9541)(z-0.7118)(z-0.4066)(z-0.1869)}\right)$$



7) Find k for no overshoot in the step response

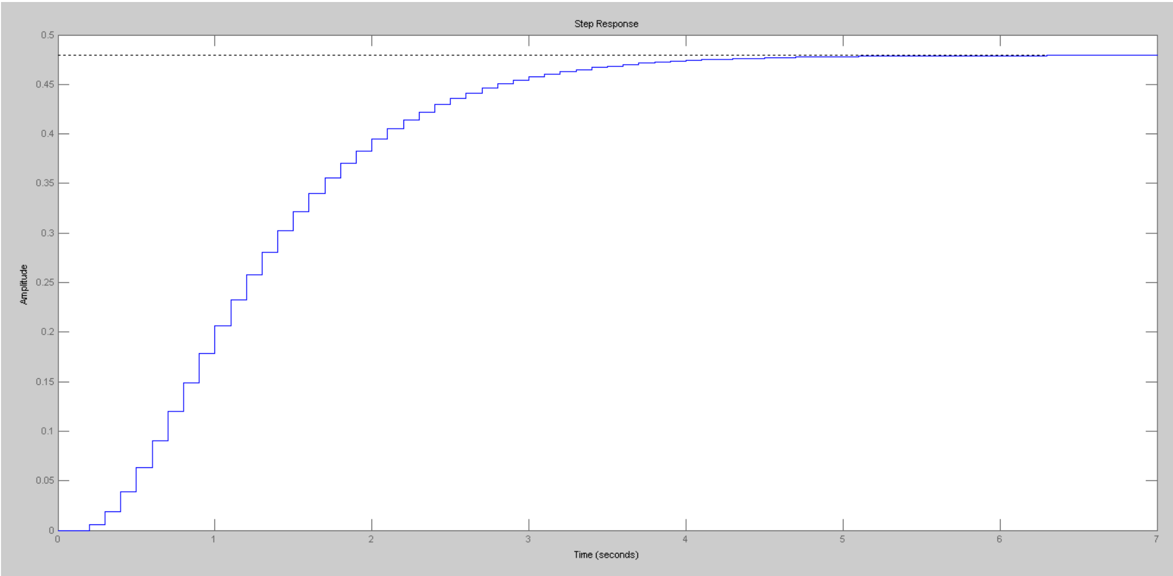
- Simulate the closed-loop system's step response

```
-->evalfr(G,0.843)

  - 0.7661403

-->k = 1/abs(ans)
   k = 1.305244
```

Simulate the closed-loop system:

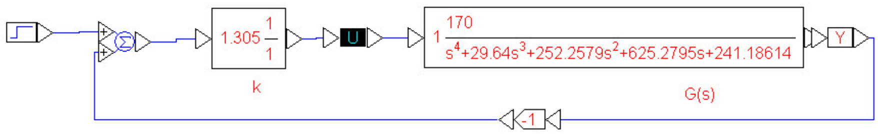
Using Matlab

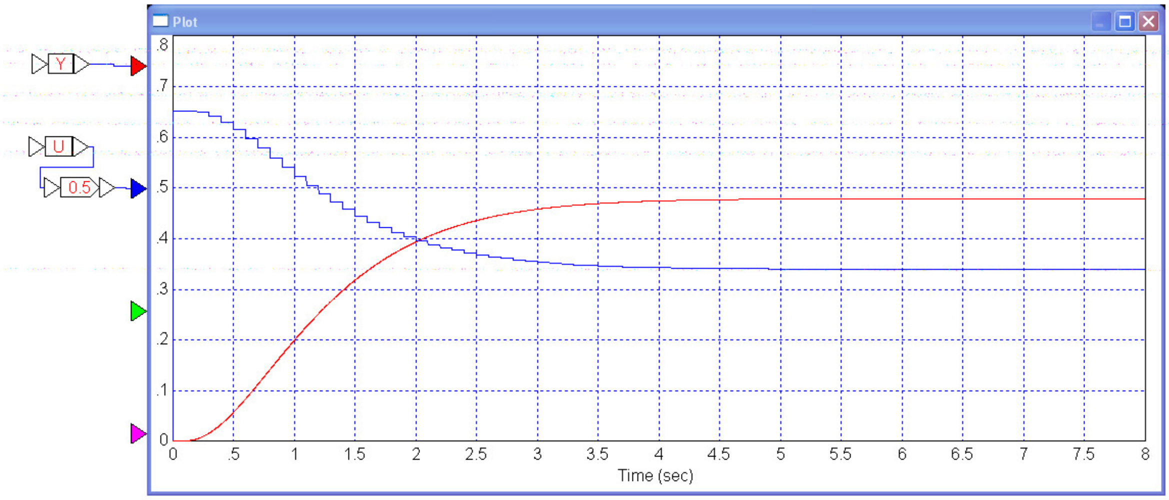
```
>> T = 0.1;
>> Gz = zpk([0,0],[0.9541,0.7118,0.4066,0.1869],0.0045,T)

                0.0045 z^2
-------------------------------------------
(z-0.9541) (z-0.7118) (z-0.4066) (z-0.1869)

Sampling time (seconds): 0.1

>> k = 1.305244;
>> Gcl = minreal(Gz*k / (1+Gz*k));
>> eig(Gcl)

   0.8434 + 0.0013i
   0.8434 - 0.0013i
   0.3832
   0.1893

>> step(Gcl)
```



Note: VisSim (or Simulink) is a better way to check the closed-loop system. These allow you to simulate:

- K(z) with a sampling rate of T
- G(s) as a continuous-time system, and
- The resulting hybrid system





To do this in Matlab, you'd have to write your own script to mix a discrete-time system, Kz, along with a continuous-time system, G(s). Doable, but not easy.

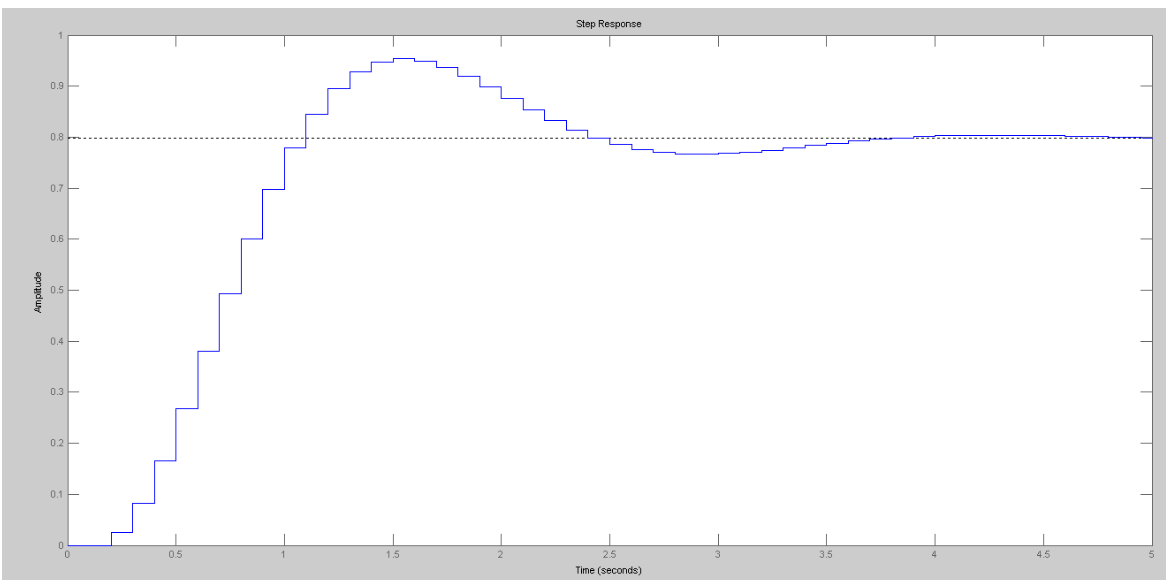
8) Find k for 20% overshoot for a step response (damping ratio = 0.4559)

- Simulate the closed-loop system's step response

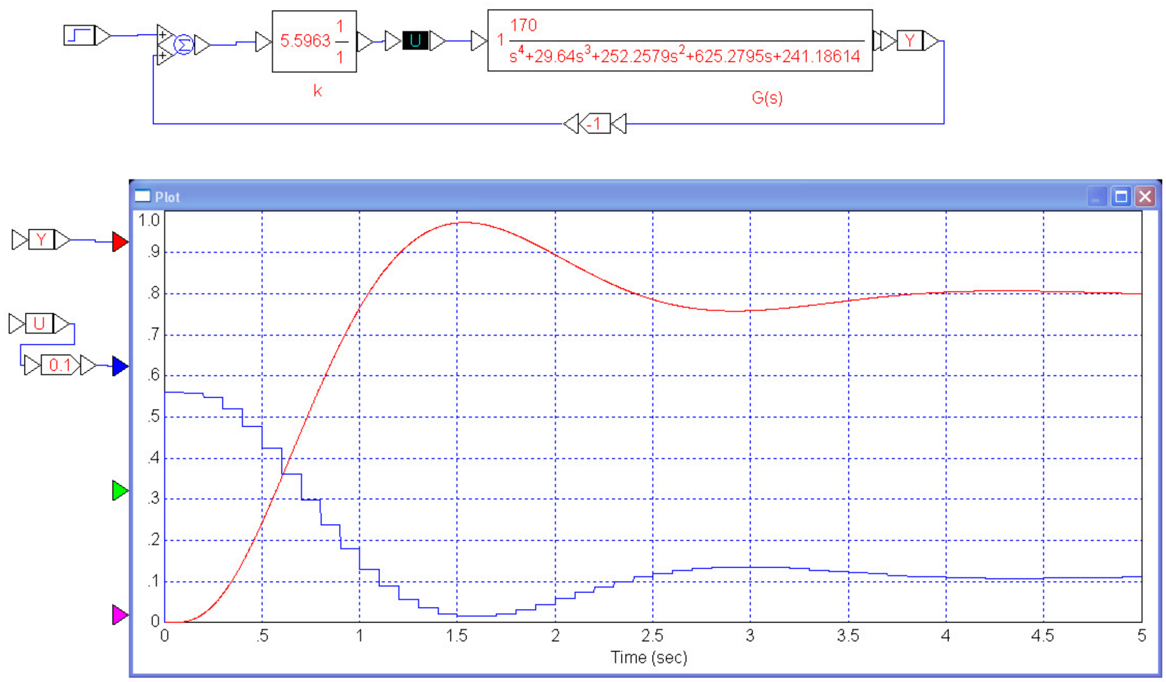
```
-->z = 0.866 + j*0.202;

-->evalfr(G,z)

  - 0.178687 + 0.0007450i

-->k = 1/abs(ans)

   k = 5.5963293
```

Checking in Matlab

```
>> T = 0.1;
>> Gz = zpk([0,0],[0.9541,0.7118,0.4066,0.1869],0.0045,T);
>> k = 5.5963293;
>> Gcl = minreal(Gz*k / (1+Gz*k));
>> eig(Gcl)

   0.8665 + 0.2019i
   0.8665 - 0.2019i
   0.3270
   0.1994

>> step(Gcl)
```



Sidelight: This is almost the same result you get with Simulink (or VisSim):



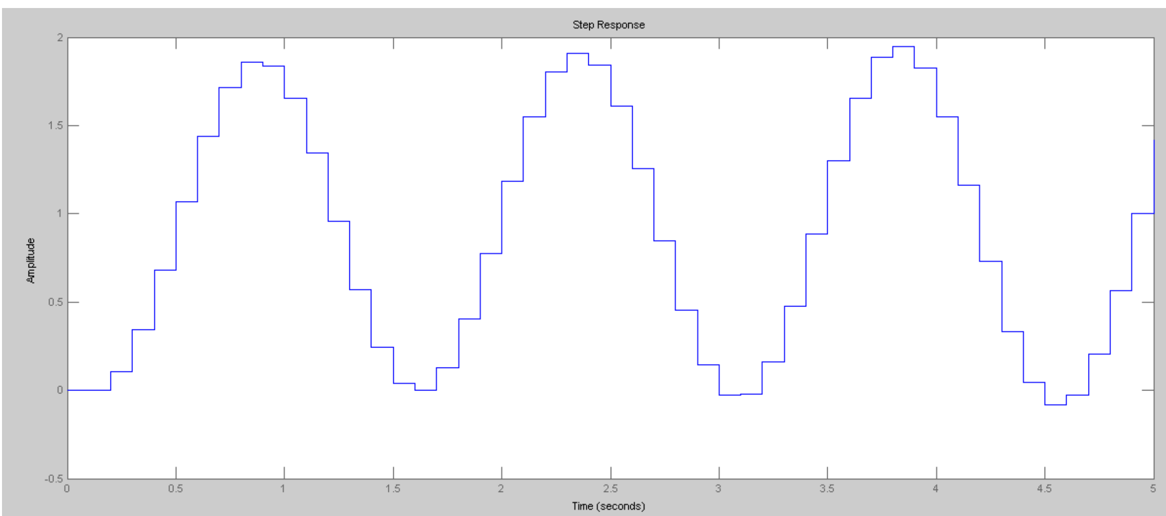
9) Find k for a damping ratio of 0.00

- Simulate the closed-loop system's step response

```
-->evalfr(G,0.912 + j*0.415)

  - 0.0430222 + 0.0002051i

-->k  = 1/abs(ans)

 k =  23.243543
```

Checking in Matlab

```
>> T = 0.1;
>> Gz = zpk([0,0],[0.9541,0.7118,0.4066,0.1869],0.0045,T);
>> k = 23.243543;
>> Gcl = minreal(Gz*k / (1+Gz*k));
>> eig(Gcl)

   0.9129 + 0.4148i
   0.9129 - 0.4148i
   0.2168 + 0.0658i
   0.2168 - 0.0658i

>> t = [0:T:5]';
>> step(Gcl,t);
```



Repeating using VisSim (similar to Simulink)