Gain Compensation using Root Locus

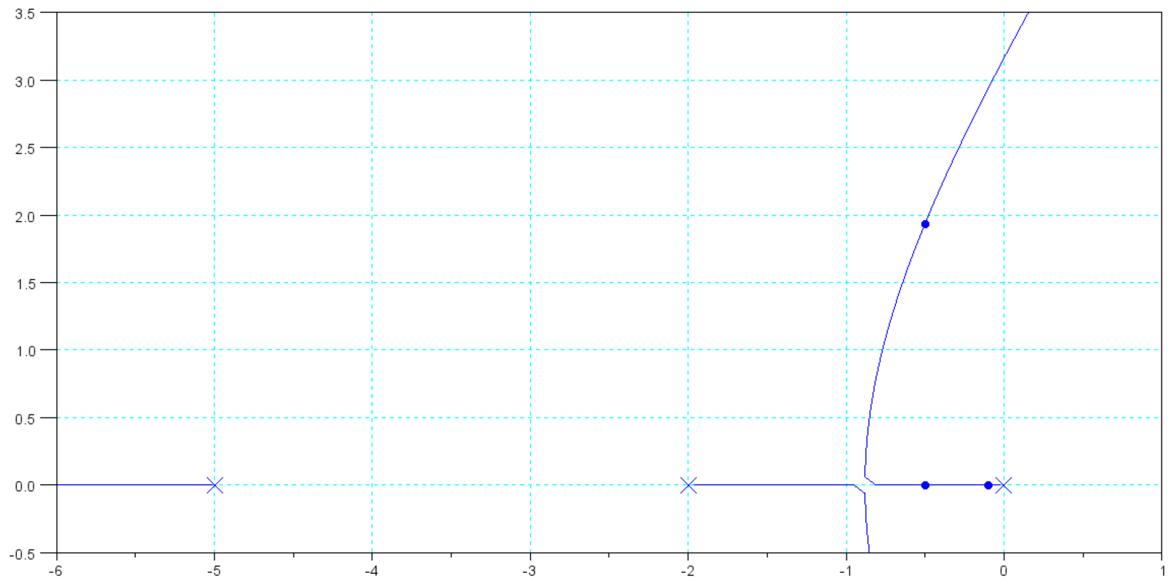
Objectives:

- Pick a feedback gain t give a desired response (such as %OS)
- Specify how the resulting closed-loop system will behave

Introduction

Assume $G(s) = \left(\frac{20}{s(s+2)(s+5)}\right)$. Determine where you can place the closed-loop poles of $\left(\frac{GK}{1+GK}\right)$.

Solution: Draw the root-locus plot. (Matlab used here)



Root Locus of G(s) along with target points at s = { -0.1, -0.5, -0.5 + j1.93 }

i) Determine K to place the closed-loop dominant pole at s = -0.1.

To do this,

$$\left(\frac{20k}{s(s+2)(s+5)}\right)_{s=-0.1} = 1\angle 180^0$$

$$(-21.482) \cdot k = 1\angle 180^0$$

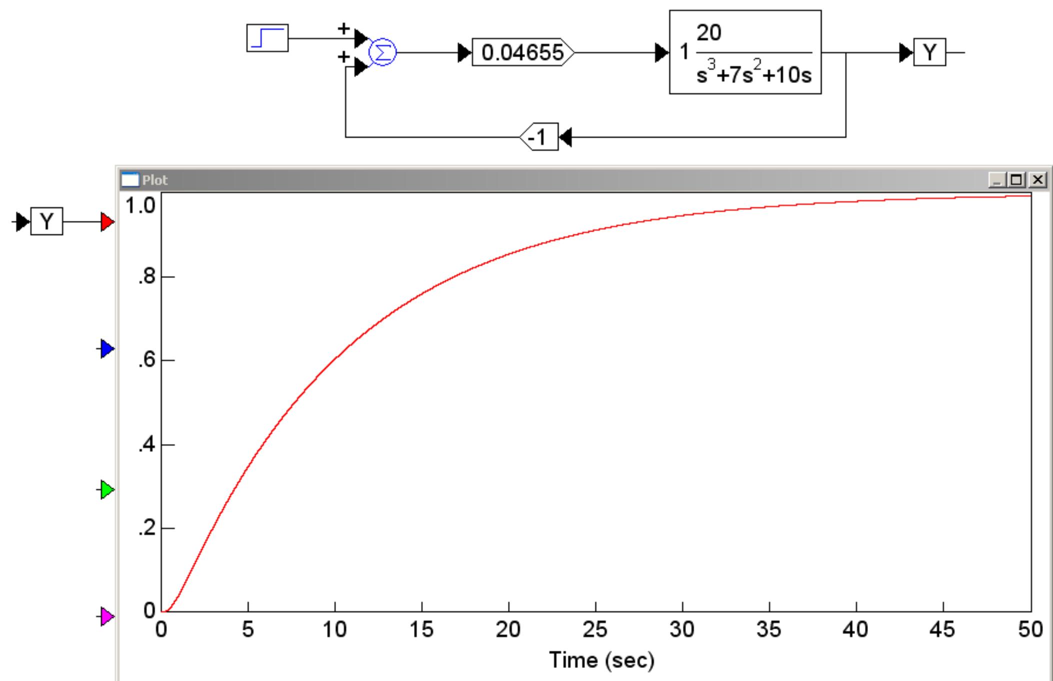
$$k = \frac{1}{21.482} = 0.04655$$

With this value of k, the step response should

- Have no error for a step input (it's a type-1 system)
- No overshoot (the dominant pole is pure real), and

- A 2% settling time of 40 seconds (4/0.1 = 40)

Verifying this in VisSim:



Verifying in Matlab:

```
G = zpk([],[0,-2,-5],20);
x = evalfr(G, -0.1)

 -21.4823

k = -1/x

    0.0465

Gcl = minreal(G*k / (1+G*k));
zpk(Gcl)

            0.931
-------------------------
(s+1.84) (s+5.06) (s+0.1)
```

The closed-loop system has a dominant pole at s = -0.1 (expected since that's where you put it)

ii) Determine K to place the closed-loop dominant pole at s = -0.5

$$\left(\frac{20k}{s(s+2)(s+5)}\right)_{s=-0.5} = 1\angle 180^0$$

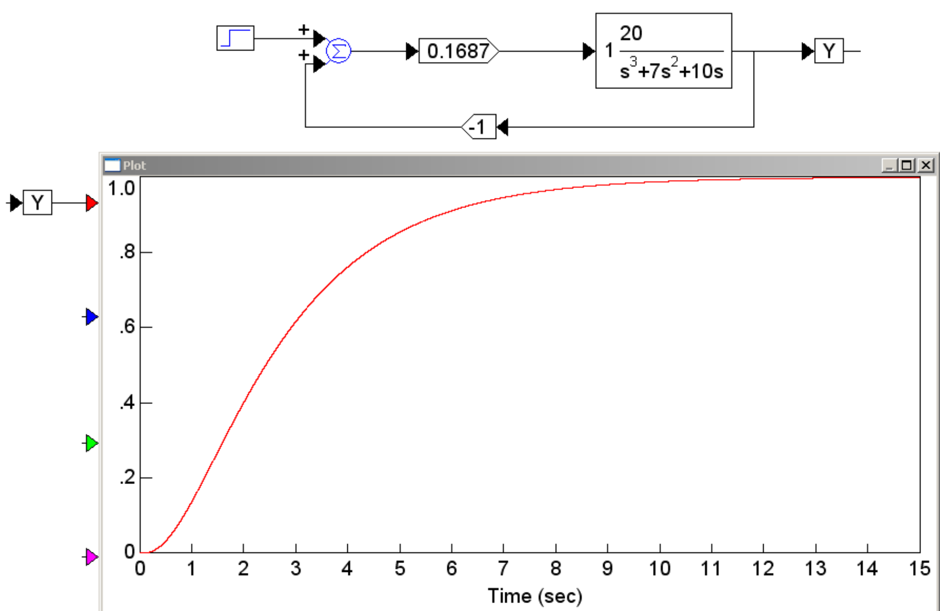
In Matlab:

```
x = evalfr(G, -0.5)

  -5.9259

k = -1/x

   0.1687

Gcl = minreal(G*k / (1+G*k));
zpk(Gcl)

             3.375
 ---------------------------
 (s+1.297) (s+0.5) (s+5.203)
```

With this value of k, the step response should

- Have no error for a step input (it's a type-1 system)
- No overshoot (the dominant pole is pure real), and
- A 2% settling time of 8 seconds (4/0.5 = 40)

In VisSim:

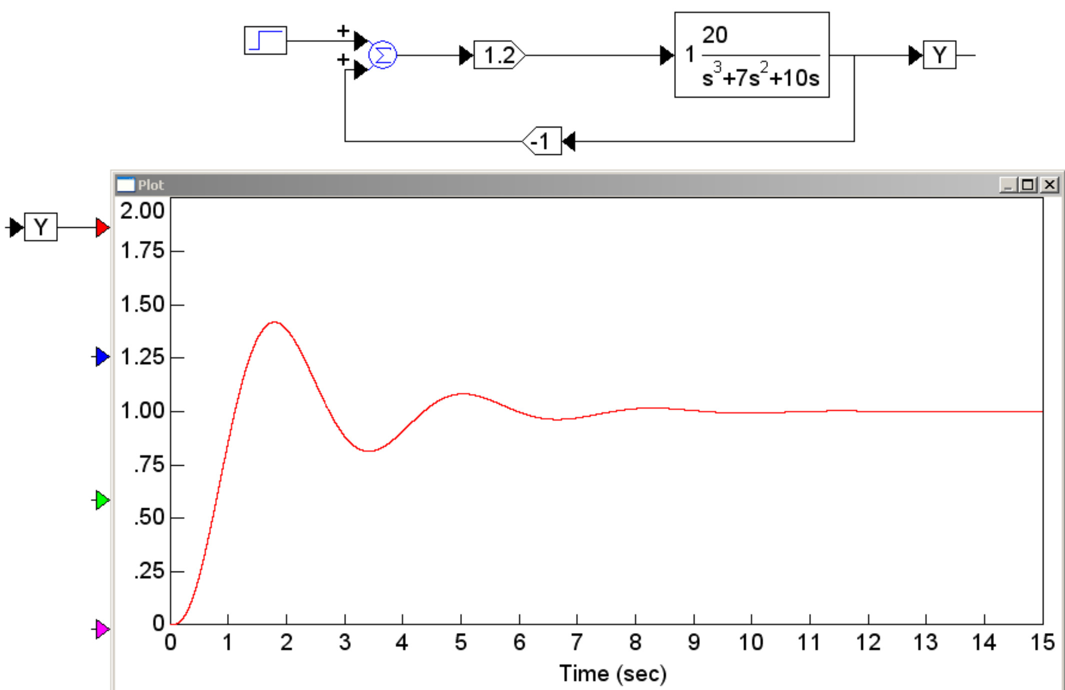


Step Response with the Closed-Loop Dominant Poles Placed at s = -0.5

iii) Place the closed-loop dominant pole at s = -0.5 + j1.9365

```
x = evalfr(G, -0.5 + j*1.9365)

 -0.8333 + 0.0000i

k = -1/x

  1.2000 + 0.0000i

k = 1.2;
Gcl = minreal(G*k / (1+G*k));
zpk(Gcl)

          24
-------------------
(s+6)  (s^2 + s + 4)
```

In VisSim:



Step Response with the Closed-Loop Dominant Poles Placed at s = -0.5 + j1.9365

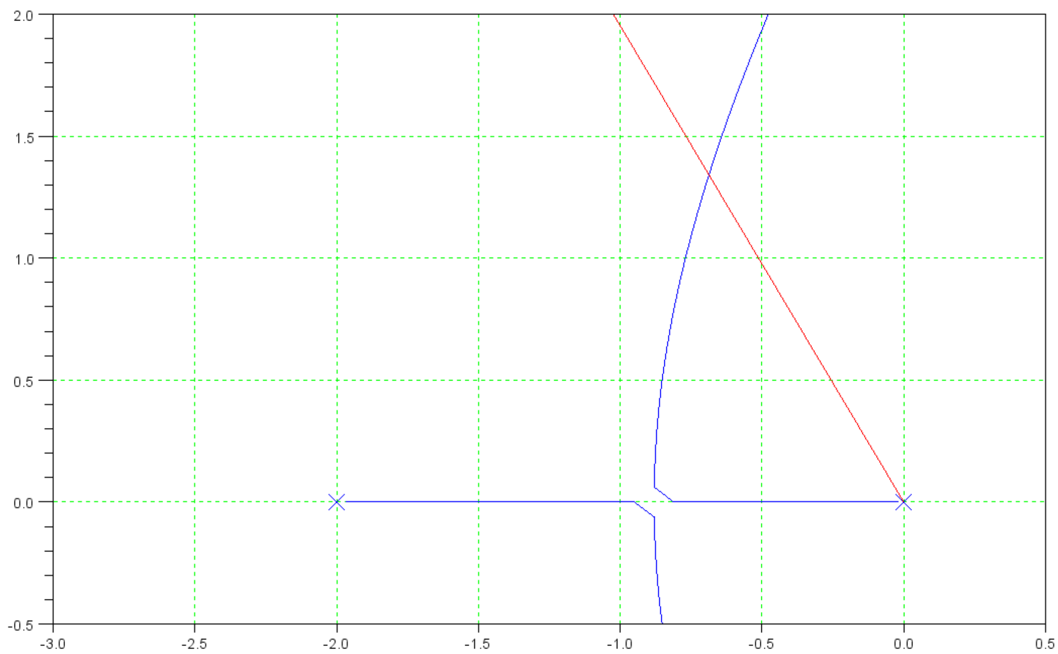
Note: The root locus does in fact predict how the system will behave for various values of k.

iv) Determine k so that

- k is as large as possible, but
- The overshoot for a step input is 20% or less.

Solution: Pick k so that the damping ratio is 0.4559 (20% overshoot)

Find the spot on the root locus which intersects the 0.4559 damping line



Root Locus with 0.4559 damping line

The solution is

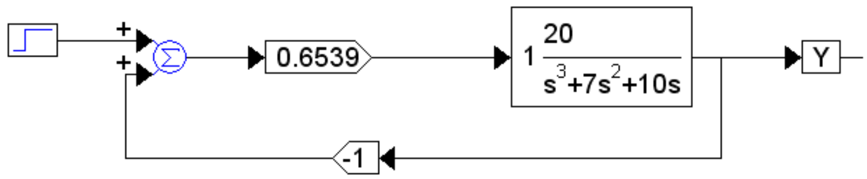
$$s = -0.6811 + j1.3623$$

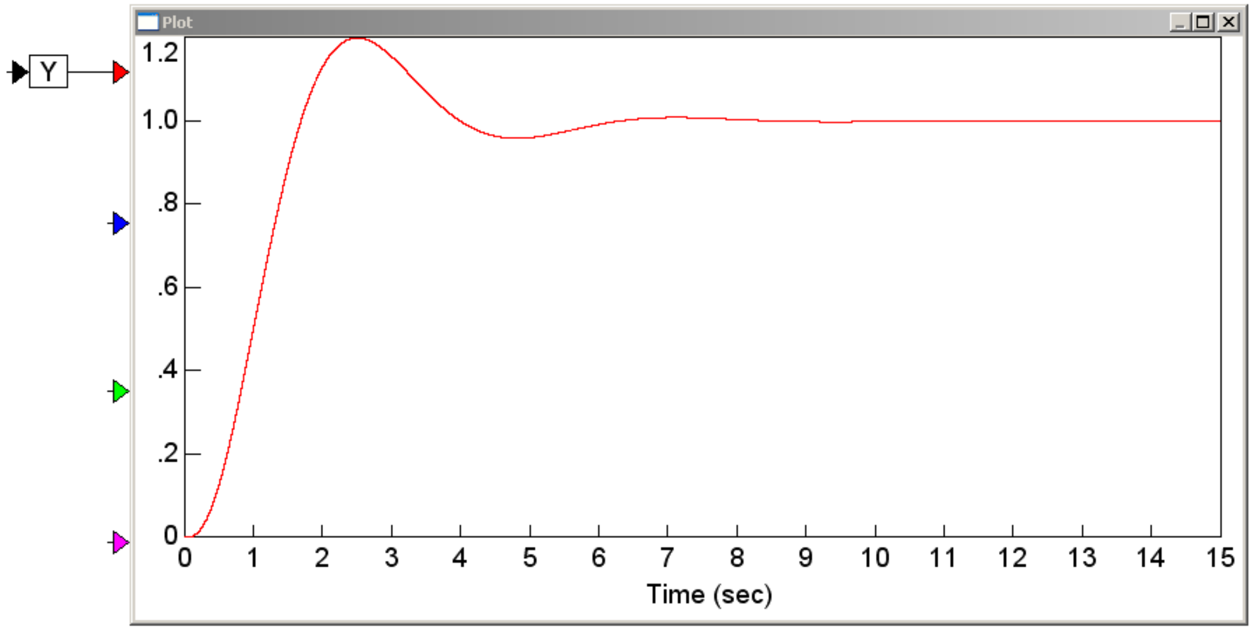
To find k:

```
s = -0.6811 + j*1.3623;
x = evalfr(G, s)

 -1.5292 - 0.0001i

k = 1 / abs(x)

   0.6539

Gcl = minreal(G*k / (1+G*k));
zpk(Gcl)

              13.0786
-------------------------------
(s+5.638) (s^2 + 1.362s + 2.32)
```

```
eig(Gcl)

  -0.6811 + 1.3623i
  -0.6811 - 1.3623i
  -5.6377
```

Checking in VisSim:



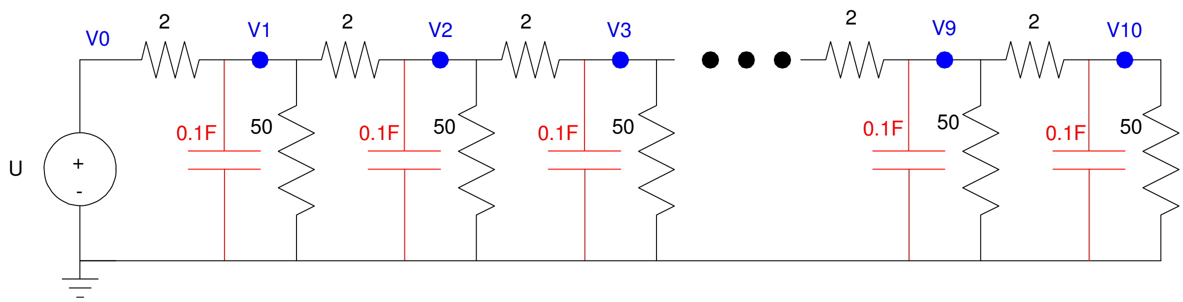


Note that the overshoot is 20% as expected

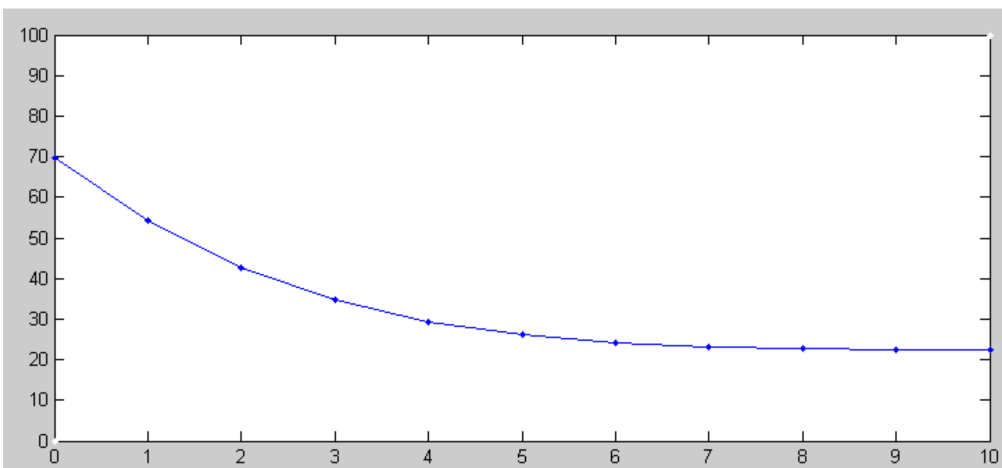
Example 2: Heat Equation

Control the tip temperature of the following 10th-order RC filter (heat.m) so that

- a) The system is as fast as possible with no overshoot, or
- b) There is 20% overshoot (or less) in the step response



Temperature along a metal bar modeled as a 10th order RC filter



Temperature along a metal bar. The feedback control law sets V0 = k (R - V10)

The transfer function from V0 to V10 is found as follows:

```
A = zeros(10,10);
for i=1:9
  A(i,i) = -20.1;
  A(i,i+1) = 10;
  A(i+1,i) = 10;
  end
A(10,10) = -10.1;
B = zeros(10,1);
B(1) = 10;
C = zeros(1,10);
C(10) = 1;
D = 0;
G = ss(A,B,C,D);
zpk(G)
```

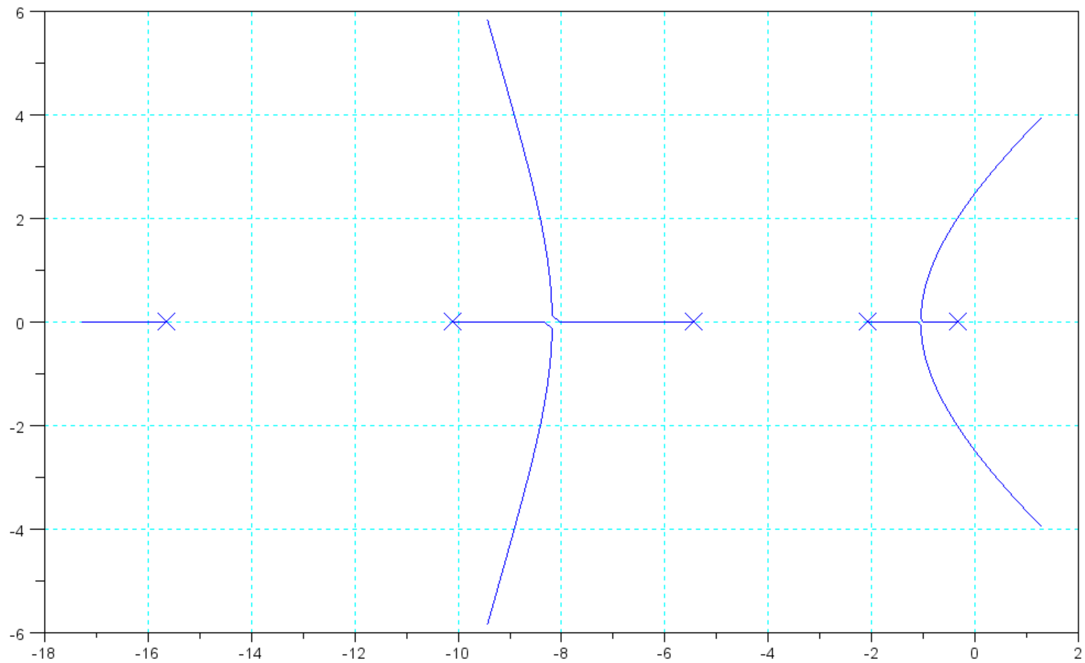
```
                                                  10000000000
---------------------------------------------------------------------------------------------------
(s+39.21) (s+36.62) (s+32.57) (s+27.41) (s+21.59) (s+15.65) (s+10.1) (s+5.439) (s+2.081) (s+0.3234)
```

A 5th-order approximation to this system

- Keeps the five most dominant poles, and
- Matches the DC gain

```
DC = evalfr(G,0)

   0.6245

P = eig(G)

 -39.2115
 -36.6248
 -32.5698
 -27.4068
 -21.5946
 -15.6496
 -10.1000
  -5.4390
  -2.0806
  -0.3234


G5 = zpk([],P([6,7,8,9,10]),1);
k = DC / evalfr(G5,0)

 361.2378

G5 = zpk([],P([6,7,8,9,10]),k);
zpk(G5)


                    361.2378
-------------------------------------------------
(s+15.65) (s+10.1) (s+5.439) (s+2.081) (s+0.3234)
```

The root-locus for this system is (using Matlab)



Root-Locus Plot for the 5th-order model

a) Find k so that

- There is no oveshoot in the step response (meaning the closed-loop dominant pole is on the real axis), and
- The system is as fast as possible (meaning the dominant pole is as far left as possible)

The point on the root locus that meets these requirements is at the breakaway point

$$s = -1.05$$

To find k at this point

$$GK = -1$$

In matlab

```
evalfr(G5, -1.05)

  -0.8318

k = 1/abs(ans)

   1.2022
```

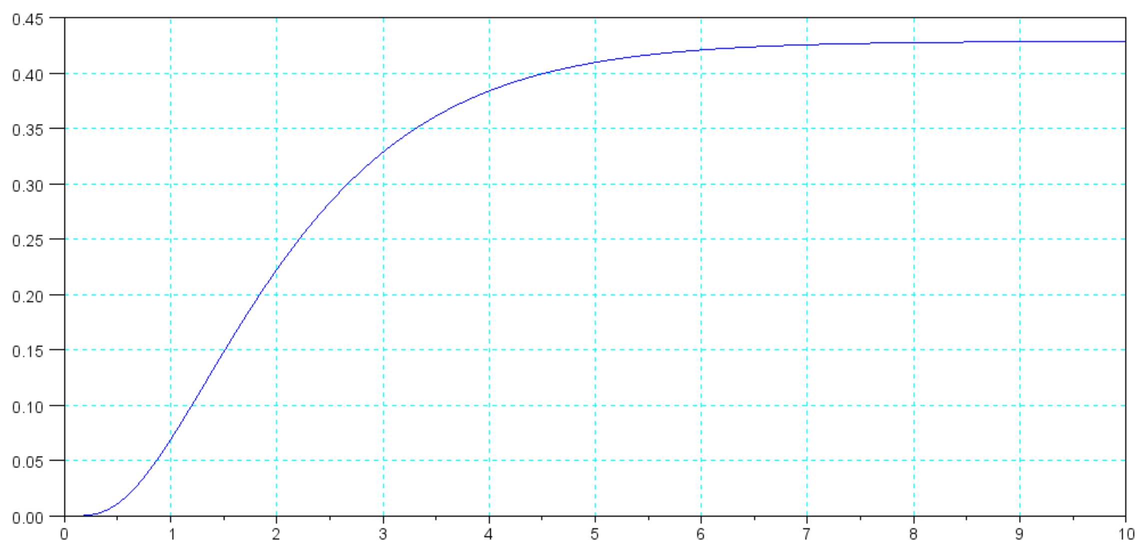
For this value of k, we expect the step response to have

- No overshoot (the dominant pole is real
- A 2% settling time of 3.81 seconds (4 / 1.05)
- A steady-state error of 0.5712

```
DC = evalfr(G5,0)

   0.6245

Kp = DC * k

   0.7508

Estep = 1 / (Kp + 1)

   0.5712
```

Checking the step response in Matlab:

```
Gcl = minreal(G5*k / (1+G5*k));
eig(Gcl)

 -15.6859
  -9.8721
  -5.9352
  -1.0500              note that the dominant pole is where we placed it
  -1.0494

t = [0:0.01:5]';
y = step(Gcl, t);
plot(t,y);
```



Step Response with k = 1.2011 (s = -1.05)

In the Matlab simulation (heat.m), this control law is implemented with the following code:

.

```
.
.
R = 100;

while(t < 100)

   V0 = 1.2011 * ( R - V(10) );

   dV(1)  = 10*V0    - 20.1*V(1) + 10*V(2);
   dV(2)  = 10*V(1) - 20.1*V(2) + 10*V(3);
   .
   .
   .
```

b) Find k so that

- The feedback gain, k, is as large as possible, but
- There is 20% overshoot (or less) in the step response

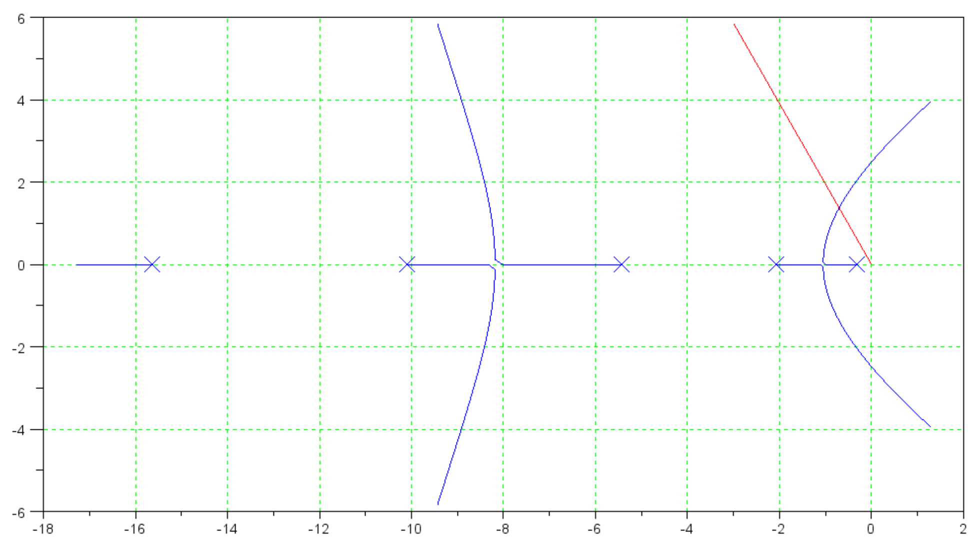
This is the same as before, but pick a different point on the root locus: the point which intersects the 20% overshoot damping line.

First, find the overshoot that results in 20% overshoot

$$\exp\left(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}\right) = 0.2$$

$$\zeta = 0.4559$$

Find the point on the root locus that intersects with this damping ratio



Root Locus along with the 0.4559 damping line

Zooming in allows you to find this point to be s = -0.7 + j1.36

Find k at this point

$$(GK)_{s=-0.7+j1.36} = -1$$

In Matlab

```
s = -0.7 + j*1.36;
evalfr(G5,s)

 -0.1879 - 0.0017i

k = 1/abs(ans)

   5.3217
```

Note that there is a complex portion of the answer. This means I didn't read the graph to 10 decimal places. The complex part is almost zero, however, so s is close.

With this value of k,

- There should be 20% overshoot in the step response (why I picked this value of k)
- The 2% settling time should be 5.71 seconds (4 / 0.7)
- The steady-state error for a step input should be 0.2313 (better than before)

```
DC = evalfr(G5,0)

   0.6245

Kp = k * DC

   3.3235

Estep = 1/(Kp + 1)

   0.2313
```

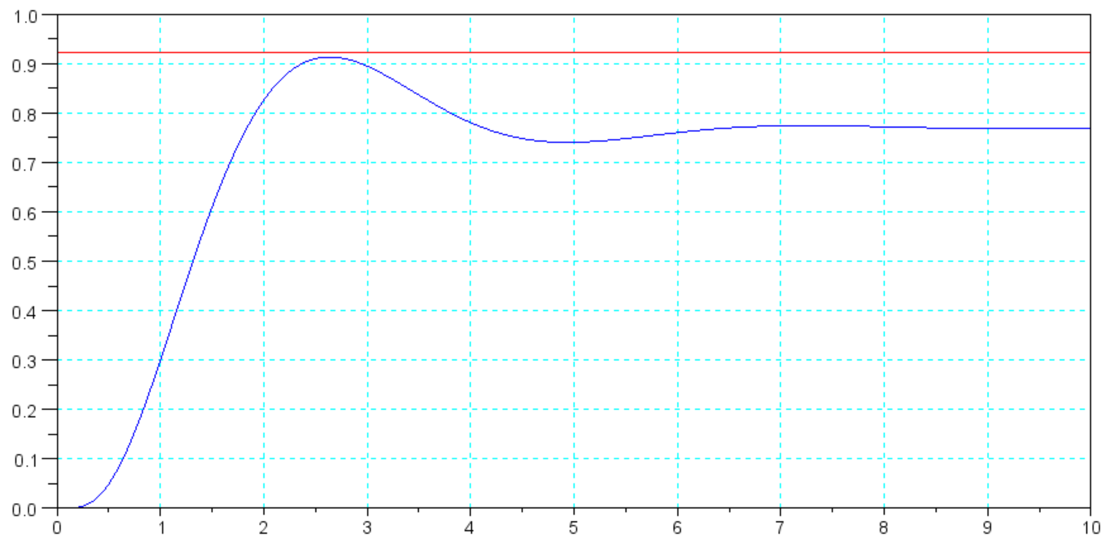
Checking the step response in Matlab

```
Gcl = minreal(G5*k / (1+G5*k));
y = step(Gcl, t);
plot(t,y);
```

The actual overshoot is 18.79% (vs. 20% expected)

```
DC = evalfr(Gcl,0)

   0.7687

OS = (max(y) - DC) / DC

   0.1879

max(y)
```

```
ans =

    0.9131

 OS = (max(y) - DC) / DC

OS =

    0.1879
```



Closed-Loop System Step Response for k = 5.3217 along with the targer (20% overshoot shown in red)

The code in heat.m changes as follows:

```
.
.
.

R = 100;

while(t < 100)

   V0 = 5.3217 * ( R - V(10) );

   dV(1)  = 10*V0   - 20.1*V(1) + 10*V(2);
   dV(2)  = 10*V(1) - 20.1*V(2) + 10*V(3);
   .
   .
   .
```