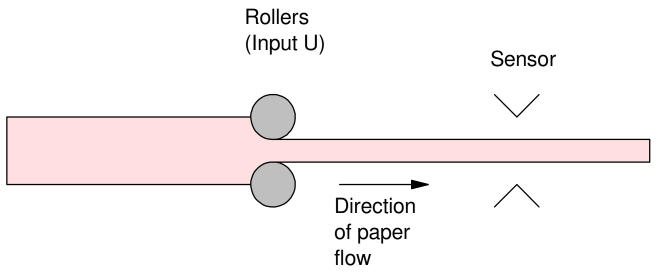
Root Locus for Systems with Delays

Source of Delays

Delays can result from many situations:

- In a paper mill, rollers are used to set the thickness of the paper being made. If the thickness sensor (beta radiation) is downstream, the changes in the roller spacing doesn't appear until that section of paper passes underneath the sensors.



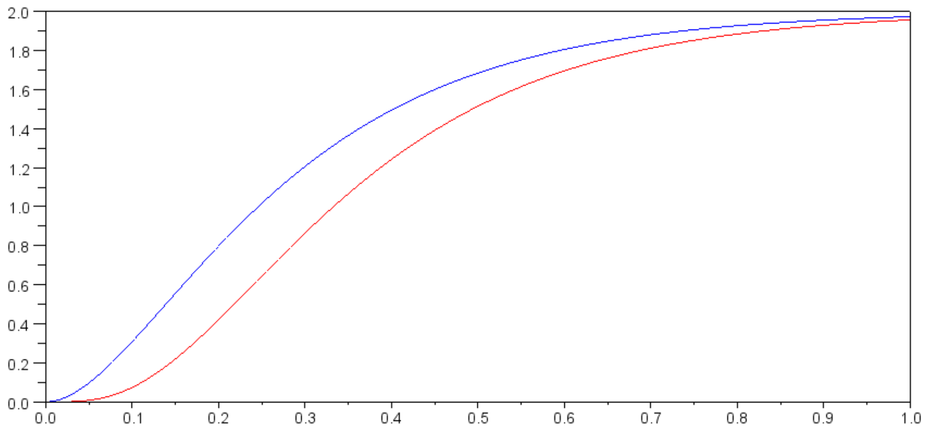
Changes in the input are not seen at the sensor for T seconds, where T = distance / paper speed.

- If you model a system using a small number of poles, the poles neglected create a delay in the output.

For example, consider the approximation

$$G(s) = \left(\frac{60{,}000}{(s+5)(s+10)(s+20)(s+30)}\right) \approx \left(\frac{100}{(s+5)(s+10)}\right)$$

The step response of the two have the same shape, but the 4th-order system has an extra delay



Step response of the 4th-order model (red) and 2nd-order approximation (blue)

To improve the 2nd-order model, a delay of T seconds could be added:

$$G(s) = \left(\frac{60{,}000}{(s+5)(s+10)(s+20)(s+30)}\right) \approx \left(\frac{100}{(s+5)(s+10)}\right) \cdot e^{-sT}$$

You can find T empirically by matching the step responses. You can also do this analytically by matching the phase at a given frequency (such as 1 rad/sec) Choosing the latter:

$$\left(\frac{60{,}000}{(s+5)(s+10)(s+20)(s+30)}\right)_{s=j1} = 1.9479\angle -21.7921^0$$

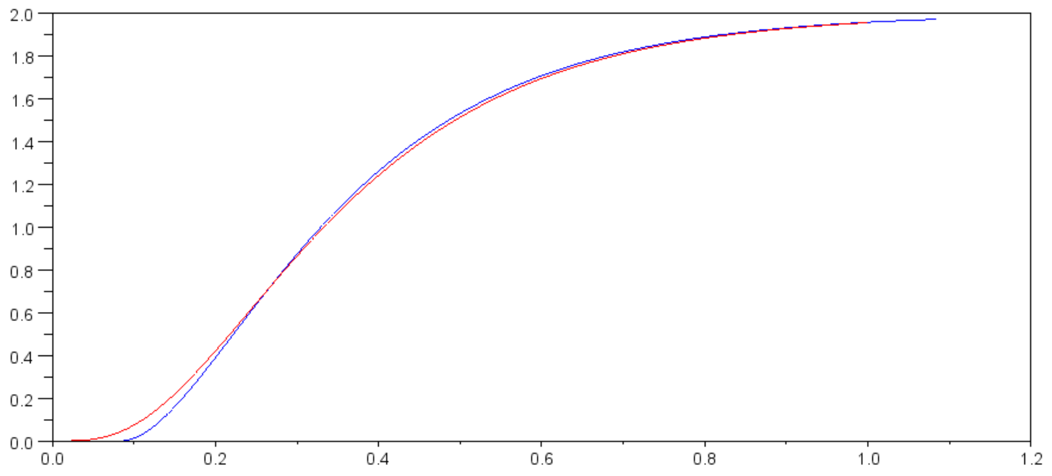
$$\left(\frac{100}{(s+5)(s+10)}\right)_{s=j1} = 1.9514\angle -17.0205^0$$

for a difference in phase of 4.7716 degrees. To match the phase at 1 rad/sec

$$\left(e^{-sT}\right)_{s=j1} = 1\angle -4.7716^0$$

$$-sT = -0.0833 \text{ radians}$$

$$T = 0.0833 \text{ seconds}$$



Step Response of 5th-Order System (red) and 3rd-Order Approximation with a Delay of 83.3ms

Better models yield better designs. If you're going to throw out a bunch of poles to simplify the model, adding a delay to compensate for the missing poles improves the accuracy of the model. The problem is that you have a term which doesn't lend itself well for root-locus techniques.

Pade Approximation:

Root locus techniques assume a system has a set of (known) poles and zeros

$$G(s) = k\frac{z(s)}{p(s)}$$

Unfortunately, delays are not in this form

$$Delay(T) = e^{-sT}$$

One way around this problem is to use the Pade approximation.

First, rewrite the delay as a numerator and denominator term:

$$e^{-sT} = \left(\frac{e^{-\frac{sT}{2}}}{e^{\frac{sT}{2}}} \right)$$

For the numerator and denominator, expand using a Taylors series

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots$$

This results in

$$e^{-sT} = \left(\frac{1+\left(\frac{-sT/2}{1!}\right)+\frac{(-sT/2)^2}{2!}+\frac{(-sT/2)^3}{3!}+\frac{(-sT/2)^4}{4!}+\frac{(-sT/2)^5}{5!}+\ldots}{1+\left(\frac{sT/2}{1!}\right)+\frac{(sT/2)^2}{2!}+\frac{(sT/2)^3}{3!}+\frac{(sT/2)^4}{4!}+\frac{(sT/2)^5}{5!}+\ldots} \right)$$

or

$$e^{-sT} = \left(\frac{1-\left(\frac{T}{2}\right)s+\left(\frac{T^2}{8}\right)s^2-\left(\frac{T^3}{48}\right)s^3+\left(\frac{T^4}{384}\right)s^4+\ldots}{1+\left(\frac{T}{2}\right)s+\left(\frac{T^2}{8}\right)s^2+\left(\frac{T^3}{48}\right)s^3+\left(\frac{T^4}{384}\right)s^4+\ldots} \right)$$

The more terms you add, the better the approximation. This is also the function *Pade(T, N)* in Matlab.

Example: Find 'k' so that the following system has 20% overshoot for its step response: (A DC servo motor with a 1/2 second delay)

$$Y = \left(\left(\frac{100}{s(s+5)(s+10)} \right) \cdot e^{-0.5s} \right) U$$

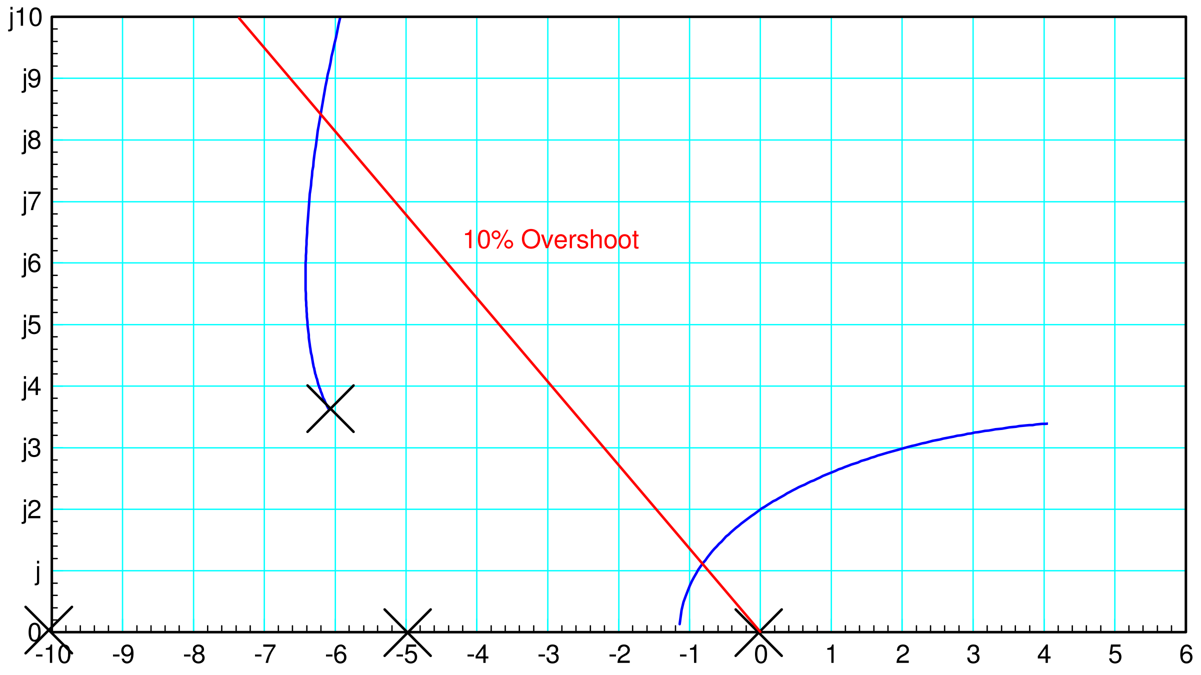
Solution #1: Use a 2nd-order Pade approximation:

```
D = tf(num,den)

s^2 - 12 s + 48
---------------
s^2 + 12 s + 48
```

$$e^{-0.5s} \approx \left(\frac{s^2-12s+48}{s^2+12s+48} \right) = \left(\frac{(s-6+j3.464)(s-6-j3.464)}{(s+6+j3.464)(s+6-j3.464)} \right)$$

In Matlab:

```
G = zpk([],[0,-5,-10],100);
[num,den] = pade(0.5, 2);
Delay = tf(num, den);
k = logspace(-2,2,1000)';
R = rlocus(G*Delay,k);
```



Root locus for G(s) with a 0.5 second delay

Note that there are poles at -6 +/- j3.464 and zeros at +6 +/- j3.464. These model the delay.

There are two solutions for a damping ratio of 0.5910 (for 10% overshoot):

s = -0.6646 + j1.3293	k = 0.4484
s = -4.5751 + j9.1502	k = 4,3299

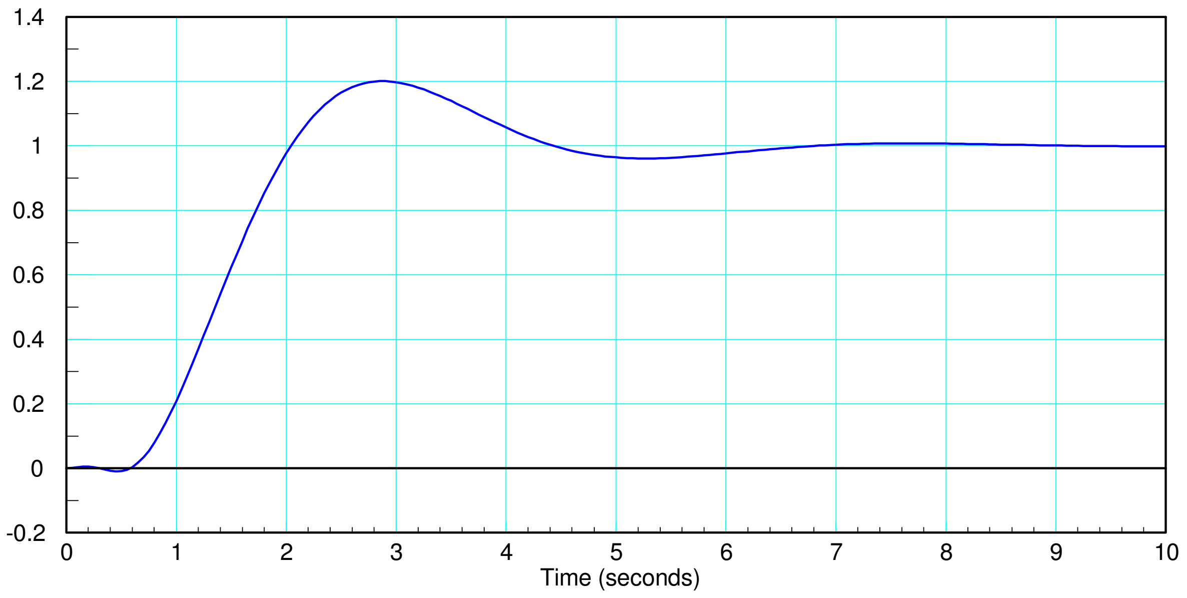
The former is the dominant pole (the one we care about). Hence,

$$\boxed{K(s) = 0.4484}$$

You can check the step response in Matlab using the Pade approximation:

```
G = zpk([],[0,-5,-10],100);
[num,den] = pade(0.5, 2);
Delay = tf(num,den)

     s^2 - 12 s + 48
```

```
      ---------------
      s^2 + 12 s + 48

k = 0.4484;

Gcl = minreal(G*Delay*k / (1 + G*Delay*k));
t = [0:0.01:10]';
y = step(Gcl,t);
plot(t,y);
```



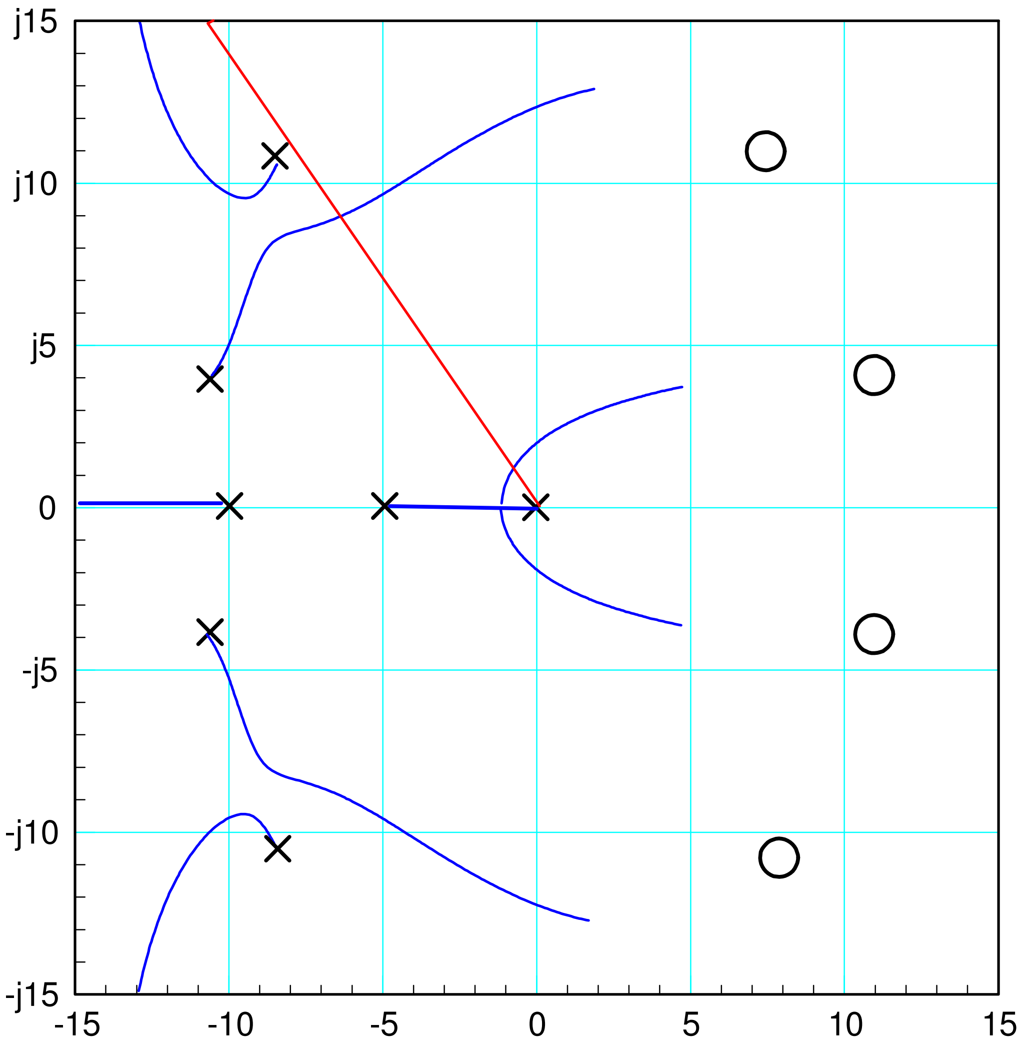
Step Response for the Closed-Loop System with a 1/2 Second Delay

Note that the Pade approximation results in the output weaving up and down for 1/2 second. This isn't exactly what a delay does, but it's the best you can do with only two poles and zeros in the approximation.

Problem: Repeat for a 4th-order Pade Approximation

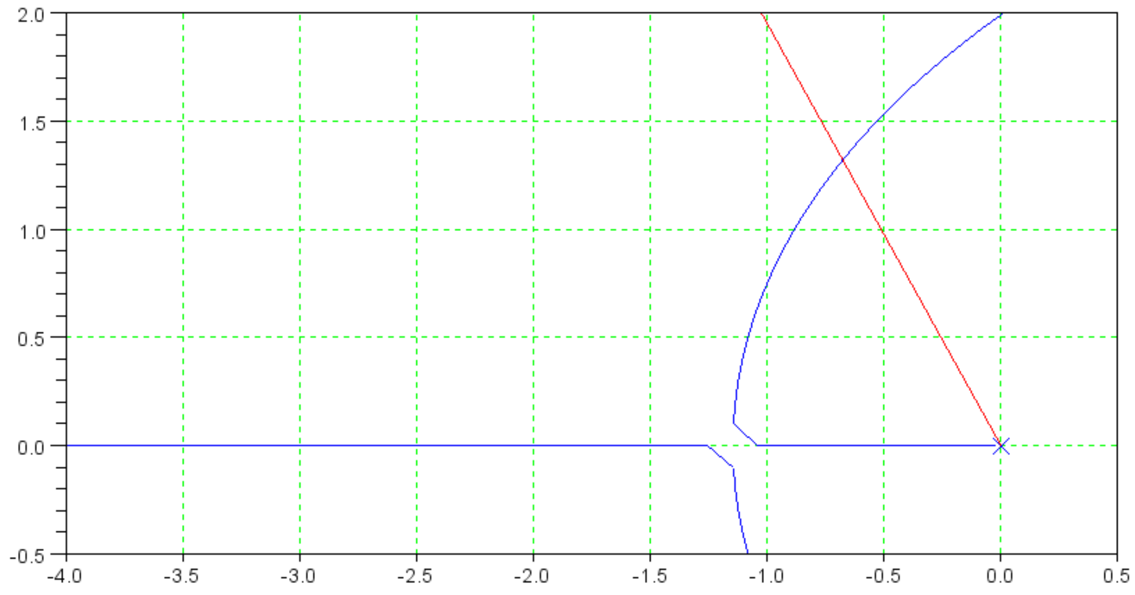
$$e^{-0.5s} \approx \left(\frac{(s-8.42 \pm j10.63)(s-11.58 \pm j3.47)}{(s+8.42 \pm j10.63)(s+11.58 \pm j3.47)} \right)$$

```
[num,den] = pade(0.5, 4);
Delay = tf(num,den)
k = logspace(-2,2,1000)';
G = zpk([],[0,-5,-10],100)
rlocus(G*Delay,k);
```



Root Locus of G(s) * delay using a 4th-order Pade approximation for the delay.
The four zeros and their mirror-image poles are from the Pade approximation.

Zooming in on the dominant pole:



Zoomed in view of the root locus for G(s) * delay

4th order Pade Model:

s = -0.6666 + j1.3333 k = 0.4566

Option #2:

This option doesn't really have a name but it's based upon root locus techniques. The idea is, for a damping ratio of 0.4559, you're searching along the line

$$s = \alpha(-1 + j2)$$

until the angle of G(s) is 180 degrees. You don't really need to use a Pade approximation to do this. Simply find the point where

$$angle\left(\frac{100 \cdot e^{-0.5s}}{s(s+5)(s+10)}\right)_{s=\alpha(-1+j2)} = 180^0$$

This method doesn't have a name and might not be taught anywhere else other than NDSU.

- It isn't really root locus design, since you're not drawing the root locus
- It sort of is root locus design, since you're finding the point on the root locus which intersects the desired damping line. That's the only point you care about anyway, so you don't need (and won't use) the rest of the root loucs.
- It's a lot easier and more accurate since you using e^{-sT} to model a delay, not an approximation. (Typing in four poles and four zeros for the 4th-order model was a bit of a pain. Typing in e^{-sT} was easy.)

In Matlab, iterate until the angle of G * delay is zero. Taking your initial guess as

$s = -0.5 + j$

then iterating by scaling s each step results in

```
-->s = 0.5 * (-1 + j*2);
-->evalfr(G,s) * exp(-s*T)

  - 2.5039068 - 0.7297865i

-->s = s * 1.1;                    10% larger
-->evalfr(G,s) * exp(-s*T)

  - 2.4062416 - 0.4851018i        good: complex portion is getting smaller.

-->s = s * 1.1;                   add another 10%
-->evalfr(G,s) * exp(-s*T)

  - 2.303687 - 0.2437231i         better:  complex portion is getting smaller

-->s = s * 1.1;                   add another 10%
-->evalfr(G,s) * exp(-s*T)

 - 2.1923504 - 0.0045661i          close:  complex portion is almost zero
```

(time passes)

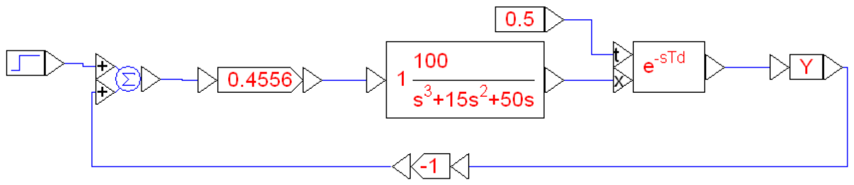
```
-->s = 0.666717 * (-1 +j*2)

  - 0.666717 + 1.333434i

-->evalfr(G,s) * exp(-s*T)

  - 2.1901017 + 0.0000002i              close enough
```

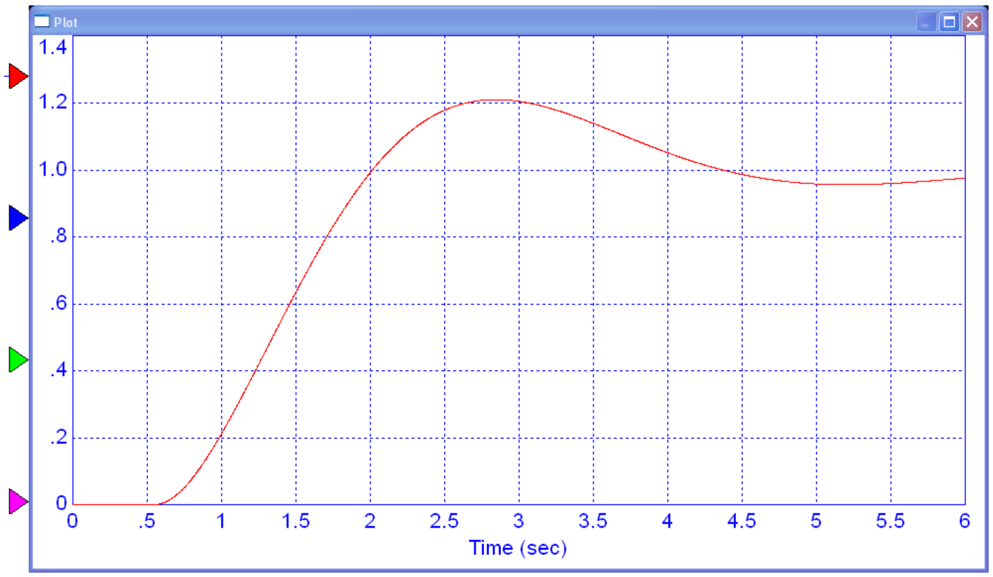
Anyway, if you use this method, the result is the exact solution

Method	s	k
2nd-Order Pade	-0.6646 + j1.3293	0.4484
4th-Order Pade	-0.6666 + j1.3333	0.4566
exp(-sT) (exact)	-0.6667 + j1.3334	0.4566

- On the plus-side, you don't need to use an approximation for the delay.
- On the minus side, you can't draw the root locus since the poles and zeros are not specified.

The real test of a design is if it works. Using VisSim with an actual 1/2 second delay, we do get 20% overshoot:





Closed-Loop Step Response with a 1/2 Second Delay