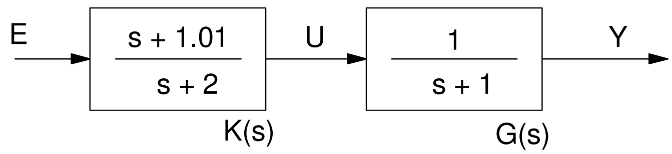
Unstable Systems and Multi-Loop Feedback

Pole-Zero Cancellation:

When you add a zero to cancel a pole, you don't actually make the pole go away. Instead, you make the initial condition small meaning the input, U, has no effect on that pole. However, nothing is perfect. When you try to cancel a pole, you will inevitably miss slightly.



For example, suppose you try to cancel a pole at s = -1 with a zero at the same location - but miss slightly. The step response of the net system will be

$$Y = \left(\frac{1}{s+1}\right)\left(\frac{s+1.01}{s+2}\right)\left(\frac{1}{s}\right)$$

Using partial fraction expansion

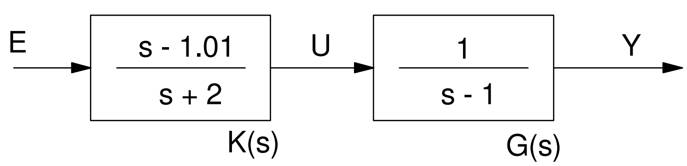
$$Y = \left(\frac{0.505}{s}\right) + \left(\frac{0.01}{s+1}\right) + \left(\frac{0.495}{s+2}\right)$$

$$y(t) = 0.505 + 0.01e^{-t} + 0.495e^{-2t} \qquad t > 0$$

$$y(t) \approx 0.505 + 0.495e^{-2t} \qquad t > 0$$

The zero makes the initial condition on this pole small. Since the pole is stable, that small term only gets smaller with time. Likewise, a good approximation for this response would be to ignore this term - implying that the pole was canceled and is no longer there.

Suppose, instead, that the pole you're trying to cancel is unstable



Now the step response is

$$Y = \left(\frac{1}{s-1}\right)\left(\frac{s-1.01}{s+2}\right)\left(\frac{1}{s}\right)$$

or using partial fractions

$$Y = \left(\frac{0.505}{s}\right) + \left(\frac{-0.00333}{s+1}\right) + \left(\frac{0.50167}{s+2}\right)$$

$$y(t) = 0.505 - 0.00333e^{t} + 0.050167e^{-2t} \qquad t > 0$$

Since the middle term blows up, it cannot be ignored. The output will blow up in spite of the pole-zero cancellation. This leads to the following rule:

You cannot cancel unstable poles

If you do, the closed-loop system will always be unstable.

Design Problem:

Design a compensator for the following system:

$$G(s) = \left(\frac{10}{(s-1)(s+1)(s+5)}\right)$$

that results in

- No error for a step input, and
- 20% overshoot for a step input

Verify your design with VisSim.

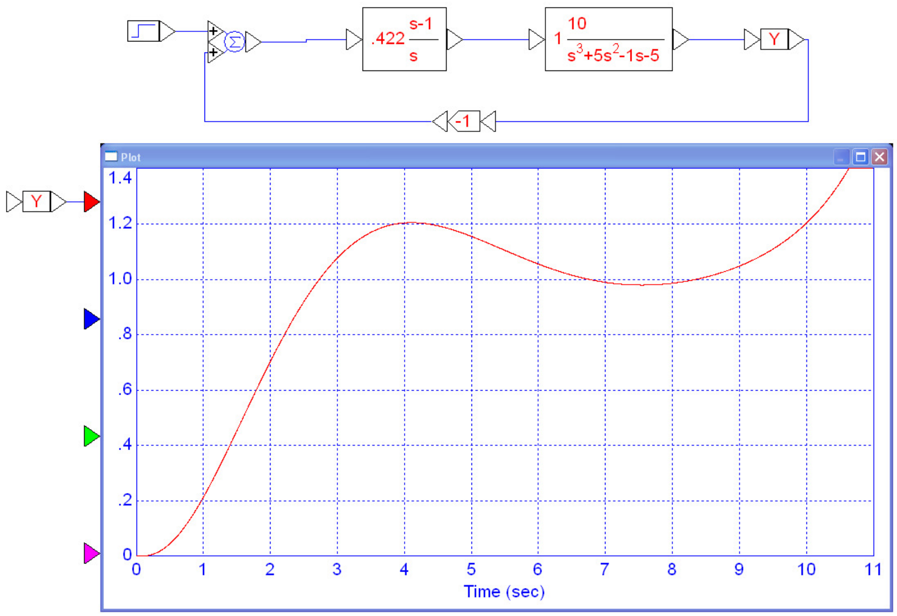
Method #1 (which won't work) .

- Cancel the pole at s = +1 since it's causing problems.
- Add a pole at s = 0 to make the system type-1.
- Add a gain of 0.4220 to place the closed-loop poles at s = -0.4031 + j0.8062

$$K(s) = 0.4220\left(\frac{s-1}{s}\right)$$

Checking the result in VisSim, the result looks OK for 6 seconds. At about 8 seconds, the system starts to blow up as e^{t}. What's happening is

- The initial condition on the pole at s = +1 is almost zero. For the first 6 seconds, that term has a negligible effect on the output.
- Numerical methods are not perfect. The zero is not exactly on top of the pole at s = +1 due to these numerical inaccuracies.
- This small initial condition grows exponentially until it takes over and sends the system off to infinity at about 10 seconds.

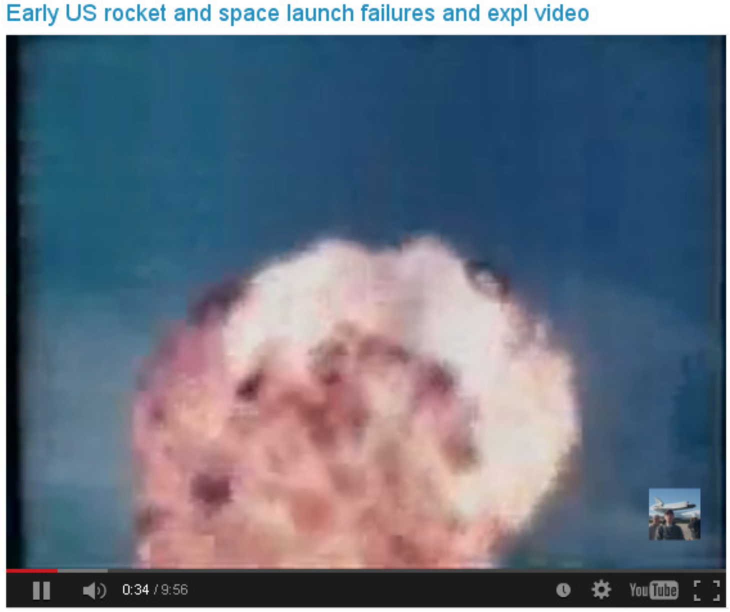


Trying to cancel an unstable pole always results in an unstable closed-loop system.

- At 8 seconds, the exponential grows large enough to show up at the output. It keep growing and growing.

If you run the simulation longer the system blows up. This is essentially what went wrong in the 1950's with the U.S. space program. Controlling a rocket is akin to an unstable system where you balance the rocket on a ball of flame. In theory, you can cancel the unstable pole and get a stable system. In practice, this is what you get.

Early US rocket and space launch failures and expl video



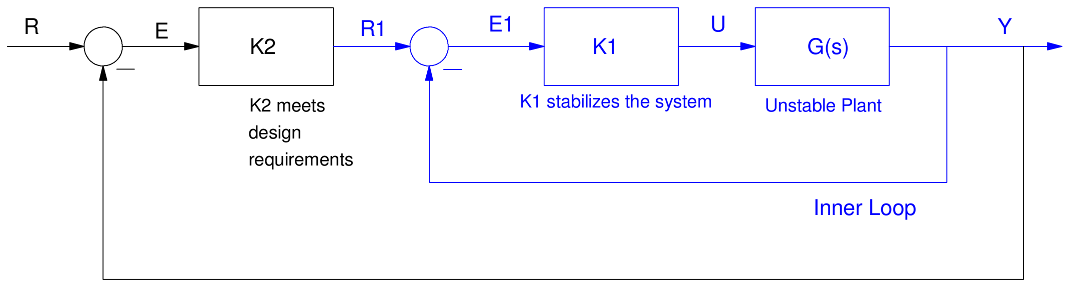
Result of trying to cancel an unstable pole in practice . (source: http://www.indyarocks.com/video/585236/Early-US-rocket-and-space-launch-failures-and-expl-video)

Method #2: Multi-Loop Feedback.

If a system is stable, you can cancel the stable poles and place them wherever you like and meet the design requirements.

If a system is unstable, then

- First, stabilize the system without any concern about meeting the requirements.
- Once stable, then add a second feedback look to meet the design requirements.



To meet the design requirements for an unstable system, create two feedback loops. The inner loop stabilizes the system, the outer loop meets the design requirements.

Going back to the previous design problem where

$$G(s) = \left(\frac{10}{(s-1)(s+1)(s+5)}\right)$$

that results in

- No error for a step input,
- 20% overshoot for a step input, and
- A 2% settling time of 4 seconds.

Verify your design with VisSim.

Step 1: Add a compensator, K1(s), to stabilize the system. Don't worry about any of the requirements - we'll take care of these in step 2.

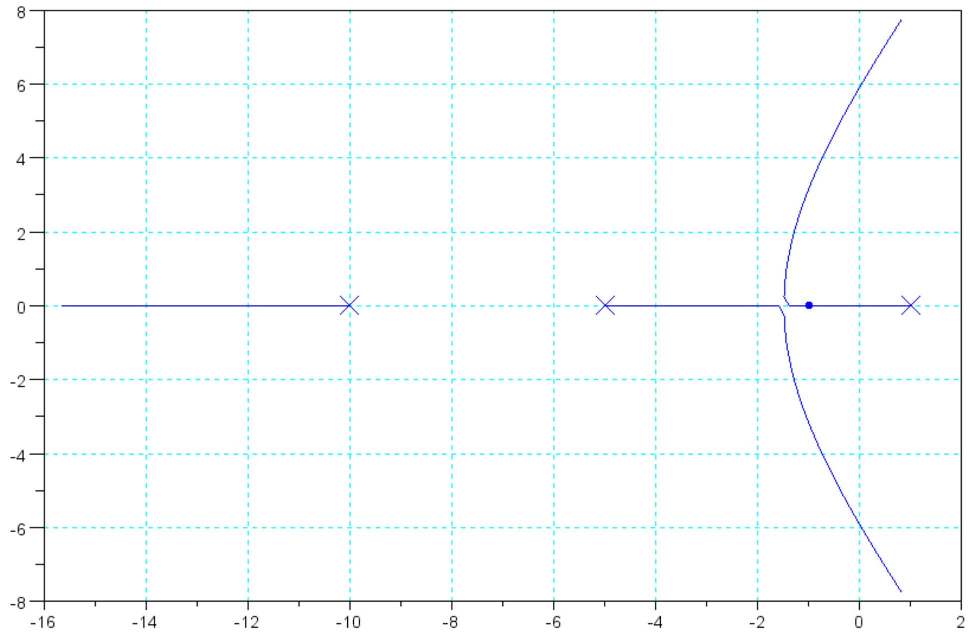
The pole at s = +1 is unstable, so you can't cancel it. However, you can cancel the pole at -1. This results in a root locus that enters the left-half plane and is stable. So

$$K_1(s) = k\left(\frac{s+1}{s+10}\right)$$

resulting in

$$GK_1 = \left(\frac{10k}{(s-1)(s+5)(s+10)}\right)$$

which has the following root locus.



Root Locus of G K1. Note that mart of the root locus is stable.

Pick any point that results in a stable system. Somewhat arbitrarily, let

s = -1

To place the closed-loop pole there

$$\left(\frac{10k}{(s-1)(s+5)(s+10)}\right)_{s=-1} = 1\angle 180^0$$

$$\left(\frac{10}{(s-1)(s+5)(s+10)}\right)_{s=-1} = 0.1389\angle 180^0$$

so

$$k = 7.200$$

and

$$K_1(s) = \left(\frac{7.2(s+1)}{s+10}\right)$$

This results in the closed-loop system being

$$G_2 = \left(\frac{GK_1}{1+GK_1}\right) = \left(\frac{72}{(s+1)(s+2)(s+11)}\right)$$

In Matlab:

```
>> G = zpk([],[1,-1,-5],10)

        10
-----------------
(s-1) (s+1) (s+5)

>> K1 = zpk(-1,-10,7.2);
>> G2 = minreal(G*K1 / (1+G*K1));
>> zpk(G2)

         72
------------------
(s+1) (s+2) (s+11)
```

Note that this doesn't meet the requirements in any way. At least it's stable though.

Step 2: Add a compensator, K2(s), so that G2(s) meets the design specifications.

$$G_2 = \left(\frac{72}{(s+1)(s+2)(s+11)}\right)$$

To meet the design requirements,

- The system should be type-1
- The closed-loop dominant pole should be at s = -1 + j2

To meet this requirement,

- Add a pole at s = 0 to make the system type-1
- Cancel the poles at -1 and -2
- Add a pole at -a so that -1 + j2 is on the root locus

$$K_2(s) = k\left(\frac{(s+1)(s+2)}{s(s+a)}\right)$$

To solve for 'a'

$$G_2K_2 = \left(\frac{72k}{s(s+11)(s+a)}\right)$$

Evaluating what we know

$$\left(\frac{72}{s(s+11)}\right)_{s=-1+j2} = 3.1574\angle -127.87^0$$

To make this 180 degrees

$$\angle(s+a) = 52.125^0$$

$$a = \frac{2}{\tan\left(52.125^0\right)} + 1$$

$$a = 2.5556$$

meaning

$$K_2(s) = k\left(\frac{(s+1)(s+2)}{s(s+2.5556)}\right)$$

and

$$G_2K_2 = \left(\frac{72k}{s(s+11)(s+2.5556)}\right)$$

To find 'k'

$$\left(\frac{72}{s(s+11)(s+2.5556)}\right)_{s=-1+j2} = 1.2461\angle 180^0$$

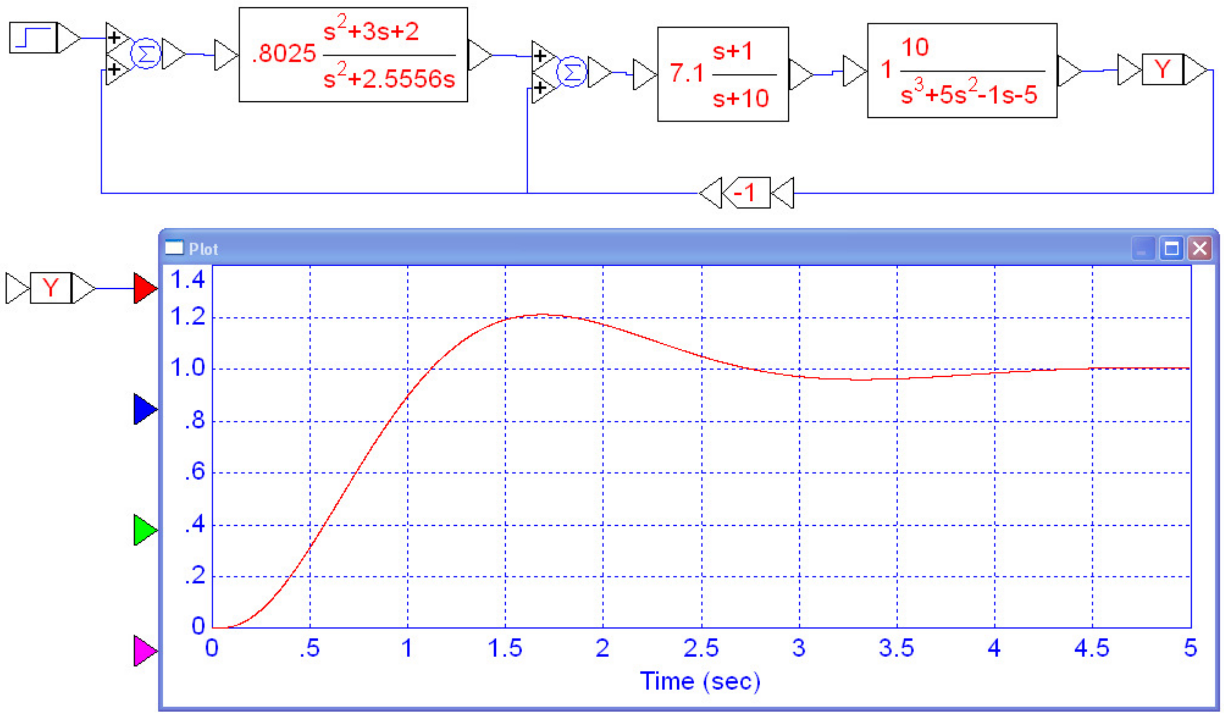
or

$$k = \frac{1}{1.2461} = 0.8025$$

and

$$K_2(s) = 0.8025\left(\frac{(s+1)(s+2)}{s(s+2.5556)}\right)$$

Step 3: Check your answer. VisSim works well here



Closed-Loop Response

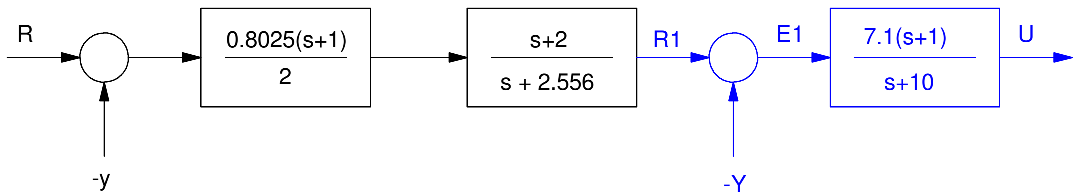
Note several things from this design

i) It doesn't really matter where you place the poles in Step 1: you're just going to cancel them in Step 2. Likewise, these poles were placed on the real axis at { -1, -2, -11 }. Real poles are easier to cancel than complex ones when using an op-amp circuit.

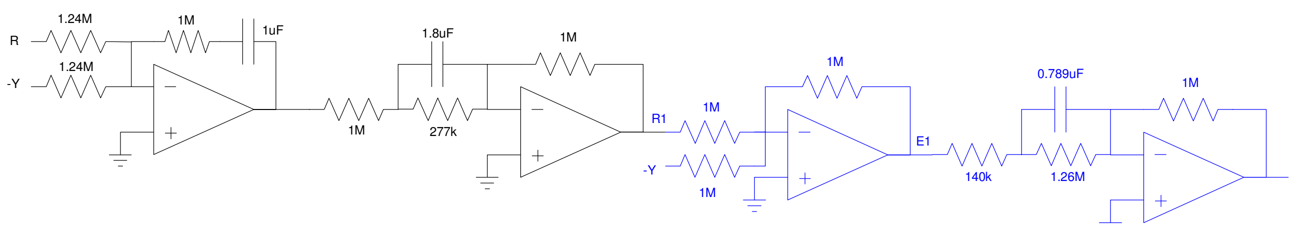
ii) The closed-loop system is stable in spite of the open-loop system being unstable. Unstable open-loop systems are OK to use - as long as your feedback control law is working.

iii) No unstable poles were canceled. The system remains stable if you run it out to 100 seconds.

The resulting compensator is then:



Compensator to stabilize G(s)



Circuit to implement the compensator: