

LECTURE 3

PID Control

Example 1 Consider a plant with transfer function $G_p(s) = \frac{1}{s^2 + 0.1s + 25} = \frac{Y(s)}{X(s)}$.

At this early stage in the course, you should already be able to discuss, at various levels, the behavior of this *dynamical system* with input $x(t)$ and output $y(t)$:

Level 1 (Coffee Room Discussion): Since ' $b^2 - 4ac$ ' is way less than zero, this system is underdamped. In other words, it will exhibit oscillatory behavior when excited by the input or in response to initial conditions. For such a second order system, the 'mantra' $s^2 + 2\zeta\omega_n s + \omega_n^2$ immediately reveals that the system undamped natural frequency is $\omega_n = 5 \text{ rad/sec}$ and the system time constant is $4\tau = 4(20) = 80 \text{ sec}$. Furthermore, since $1/\tau = \zeta\omega_n = 0.05$, the system damping ratio is $\zeta = 0.01$. Since $\omega_d = \omega_n \sqrt{1 - \zeta^2}$, the damped natural frequency is essential equal to the undamped natural frequency. Since we're *real* engineers, let's talk frequency in Hz. This system will oscillate at $f_d = \omega_d / 2\pi \cong 0.8 \text{ Hz}$. [OK. I used my smart phone's calculator for that one ☺.] This means the period of an oscillation will be $T_d = 1.25 \text{ sec}$. So, we'll see a crazy number of oscillations (~65) over the time it takes to die out. [Again, I used my calculator ☺.] It is, after all, a pretty small damping ratio. Finally, the system *static gain* is $1/25 = 0.04$.

Level 2: (Conference Room Discussion with Internet and Matlab access): I searched Google for a Laplace transform pair with denominator $s^2 + 2\zeta\omega_n s + \omega_n^2$. No luck ☹ But I did find: $\frac{b}{(s+a)^2 + b^2} \quad e^{-at} \sin(bt)$

So, yeah, I can 'complete the square' (But geez!):

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = [s^2 + 2\zeta\omega_n s + (\zeta\omega_n)^2] + [\omega_n^2 - (\zeta\omega_n)^2] = (s + \zeta\omega_n)^2 + \omega_d^2.$$

Hence,

$$G_p(s) = \frac{1}{s^2 + 0.1s + 25} = 0.2 \left(\frac{5}{s^2 + 0.1s + 25} \right) \cong 0.2 \left(\frac{5}{(s + .05)^2 + 5^2} \right)$$

And so the system impulse response is:

$$g_p(t) = 0.2e^{-0.05t} \sin(5t)$$

To get the formula for the system step response, you found

$$\frac{\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} = \frac{1}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_n \sqrt{1 - \zeta^2} t + \cos^{-1}(\zeta))$$

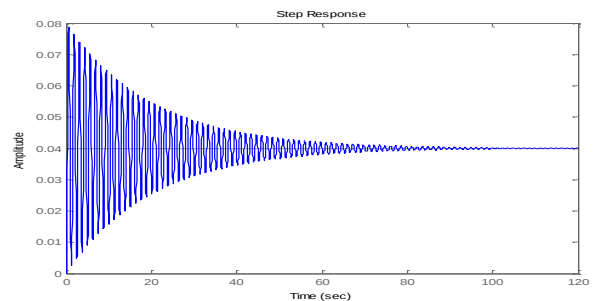
Then you tell me: Sorry man, but you didn't search hard enough:

$$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{\omega_n}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_n \sqrt{1 - \zeta^2} t)$$

So the system step response is: $y_{\text{step}}(t) \cong 0.04[1 - e^{-0.05t} \sin(5t)]$.

Using the Matlab commands: `>> Gp=tf(1,[1 .1 25]); >> step(Gp)`
Resulted in the step response at right.

Ha! Exactly like I described in the coffee room ☺.



Unity Feedback PID Control

(Back to the coffee room-which, by the way, has a white board):

$$\text{Clearly, } W(s) = \frac{Y(s)}{Y_R(s)} = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)}.$$

How, write $G_c(s) = n_c(s)/d_c(s)$ and $G_p(s) = n_p(s)/d_p(s)$.

Then the *open loop* transfer function is: $G_c(s)G_p(s) = \frac{n_c(s)n_p(s)}{d_c(s)d_p(s)} = \frac{n(s)}{d(s)} = G(s)$, and so $W(s) = \frac{n(s)}{n(s) + d(s)}$.

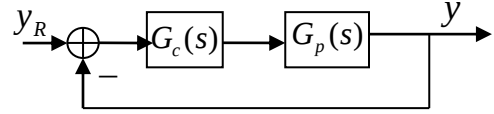


Figure 1 Unity feedback block diagram.

Transfer Function for a PID Controller: $G_c(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$.

The resulting *open loop* transfer function is: $G(s) = G_c(s)G_p(s) = \left(\frac{K_d s^2 + K_p s + K_i}{s} \right) \left(\frac{1}{s^2 + 0.1s + 25} \right) = \frac{K_d s^2 + K_p s + K_i}{s^3 + 0.1s^2 + 25s}$

So, the *closed loop* transfer function is: $W(s) = \frac{K_d s^2 + K_p s + K_i}{s^3 + (0.1 + K_d)s^2 + (25 + K_p)s + K_i}$.

How does PID control in a unity feedback control system alter the system dynamics?

- (i) The inclusion of integral control results in a closed loop system with *static gain* $W(0) = 1$. However, it also changes the order of the system from 2 to 3.
- (ii) If we want to retain a 2nd order system, we must use PD control: $G_c(s) = K_d s + K_p$. In this case, the closed loop transfer function becomes $W(s) = \frac{K_d s + K_p}{s^2 + (0.1 + K_d)s + (25 + K_p)}$, with $W(0) = \frac{K_p}{25 + K_p}$. To control the system dynamics, (assuming we still want an underdamped system) write

$$s^2 + (0.1 + K_d)s + (25 + K_p) = s^2 + 2\zeta\omega_n s + \omega_n^2.$$

This gives: $\omega_n = \sqrt{25 + K_p}$, $\tau = 2/2\zeta\omega_n = 2/(0.1 + K_d)$, and $\zeta = (0.1 + K_d)/2\sqrt{25 + K_p}$.

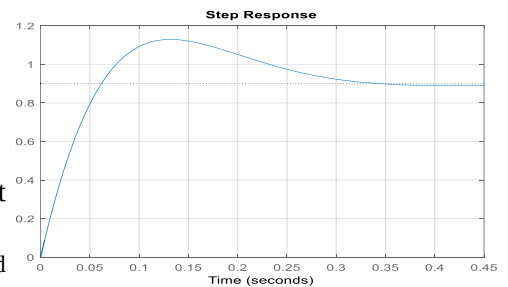
Things to note:

- The response time is inversely proportional to K_d .
- The damping is proportional to K_d , and inversely proportional to $\sqrt{K_p}$.
- The static gain is proportional to K_p , but can never be made to equal 1.0.

For example, suppose that we want static gain $g_s = 0.9 = K_p/(25 + K_p)$.

This requires $K_p = 225$. Suppose that we want ‘optimal’ damping $\zeta = 0.707$. This requires $K_d = 22.25$. The resulting time constant is then $\tau \approx 2/K_d = 0.09$ sec. The closed loop unit step response at right was obtained using the Matlab commands:

```
>> Gc=tf([22.25 225],1); >> G=Gc*Gp; >> W=feedback(G,1); >> step(W) >> grid
```



Comment: The static gain and settling time are as expected. But the amount of overshoot is much greater than what one would expect for $\zeta = 0.707$. This excess overshoot is due to the *zero* in

$$W(s) = \frac{K_d s + K_p}{s^2 + (0.1 + K_d)s + (25 + K_p)}.$$

Our design only focused on the *poles*. Clearly, one should **not** ignore the zeros. The O.D.E. is

$$\ddot{y} + (0.1 + K_d)\dot{y} + (25 + K_p)y = K_d\dot{x} + K_p x.$$

We see that the response is influenced not only by the input, but also by its *derivative*. The derivative of a unit step input is a unit *impulse*. It is this impulse that causes the response to overshoot the way it does in the figure.

Introduction to the Root Locus Method As noted above, incorporating any integral control will result in *static gain* $W(0) = 1$. The ‘cost’ is that the system will then become a 3rd order system. Suppose that we now add integral control to the PD controller we designed. The controller transfer function is then $G_c(s) = \frac{22.25s^2 + 225s + K_i}{s}$. Hence, the open loop transfer function becomes

$$G(s) = G_c(s)G_p(s) = \left(\frac{22.25s^2 + 225s + K_i}{s} \right) \left(\frac{1}{s^2 + 0.1s + 25} \right)$$

The closed loop characteristic polynomial is, therefore,

$$p(s) = s^3 + (0.1 + 22.25)s^2 + (25 + 225)s + K_i = s^3 + 22.35s^2 + 250s + K_i.$$

If we had a cubic formula similar to the quadratic formula, then we could directly evaluate the influence of K_i on the three roots of this polynomial. Unfortunately, such a formula does not exist. Hence, we will use the ‘root locus’ method. To this end, begin with setting

$$p(s) = s^3 + (0.1 + 22.25)s^2 + (25 + 225)s + K_i = s^3 + 22.35s^2 + 250s + K_i = 0. \quad (1)$$

Clearly, the values of s that solve (1) are the roots. Dividing both sides of (1) by $s^3 + 22.35s^2 + 250s$ gives

$$1 + \frac{K_i}{s^3 + 22.35s^2 + 250s} = 0. \quad (2)$$

Still, the values of s that solve (2) are the same roots. Define the fictitious open loop transfer function

$$Q(s) = \frac{1}{s^3 + 22.35s^2 + 250s}. \quad (3)$$

Then (2) becomes:

$$1 + K_i Q(s) = 0. \quad (4)$$

Now compare the form (4) to the denominator of $W(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)}$. They have exactly the same general form.

Recalling that $G_c(s)G_p(s)$ is the open loop transfer function associated with Figure 1, is the reason for defining the fictitious open loop system (3). Using the Matlab command `rlocus(Q)` will give us a plot of the roots of the corresponding fictitious closed loop system that has roots associated with (1).

In the root locus for (1) at right, we used the data cursor to see where the roots are for $K_i = 500$.

COMMENT on these roots.

>> mean(x) = 5.5000 std(x) = 3.0277

