ECE3084-L18 PID Control

Monday, April 17, 2017 2:20 PM

Talking about "tracking" implies steady state in this context.

Our P controller was robust to system changes, but did not track perfectly.

Is this always the case?

Example:

$$G_p(s) = \frac{1}{s(s+1)} \qquad G_c(s) = K_p \qquad r(t) = u(t)$$

$$H(s) = \frac{\frac{K_p}{s(s+1)}}{1 + \frac{K_p}{s(s+1)}} = \frac{K_p}{s^2 + s + K_p}$$

$$y_{ss} = \lim_{t\to\infty} y(t) \overset{\text{FVT}}{=} \lim_{s\to 0} sY(s) = \lim_{s\to 0} s\,H(s)X(s) = \overbrace{\lim_{s\to 0} H(s)}^{\text{DC Freq response } H(j0)} = \frac{K_p}{K_p} = 1$$

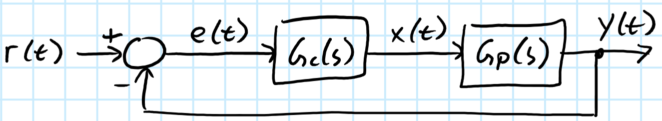
No tracking error in this case! Why?

The marginally stable $\frac{1}{s}$ pole of the open loop system integrates out any steady state error.

What if $G_p(s)$ doesn't have a pole at $s=0$?

Put one in the controller:

<u>PI Control</u>: $G_c(s) = K_p + \frac{K_i}{s}$



Controller output is $X(s) = G_c(s)E(s) = K_p E(s) + \frac{K_i}{s}E(s)$

$$x(t) = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau$$

PI = Proportional Integral

Problem with P control − Must have nonzero error in order to generate a force to counteract an external force (e.g. gravity).

The integrator allows the controller to output a force even when the error is zero.

Example (From before):

$$G_p(s) = \frac{4}{s+2} \qquad G_c(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s}$$

$$H(s) = \frac{\left(\frac{4}{s+2}\right)\left(\frac{K_p s + K_i}{s}\right)}{1 + \left(\frac{4}{s+2}\right)\left(\frac{K_p s + K_i}{s}\right)} = \frac{4\,(K_p s + K_i)}{s^2 + 2s + 4K_p s + 4K_i}$$

$$H(s) = \frac{4\,(K_p s + K_i)}{s^2 + (2+4K_p)s + 4K_i}$$

for $r(t) = u(t)$: $\quad y_{ss} = \lim_{s\to 0} s\,\frac{H(s)}{s} = \frac{4K_i}{4K_i} = \underline{1}$ Perfect Tracking

How do we pick K_p & K_i?

Put the poles where you want them (i.e. move the poles):

If we want $\zeta = 1$ (critically damped) and $\omega_n = 4$:

$$2 + 4K_p = 2\zeta\omega_n \qquad 4K_i = \omega_n^2$$

$$2 + 4K_p = 8 \qquad 4K_i = 16$$

$$K_p = \frac{6}{4} = \frac{3}{2} \qquad K_i = 4$$

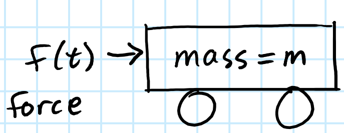
$$H(s) = \frac{4\left(\frac{3}{2}s + 4\right)}{s^2 + 8 + 16} = \frac{6s + 16}{(s+4)^2}$$

A low pass and a band pass filter added together.

Can also design K_p & K_i to cancel out the $(s+2)$ pole and replace it with another one. But this is risky — you likely won't get perfect cancellation in the real world and the undesired pole could sneak through to the output.

What if $G_p(s)$ already has a pole at $s=0$?

Example: Wheeled cart: $\quad y(t)$ = position $\qquad$ From Physics: $f = ma = m\ddot{y}$



$$\underbrace{m\ddot{y}}_{\text{total force on the cart}} = \underbrace{-d\dot{y}}_{\text{Friction (air resistance at "low" speed)}} + \underbrace{f(t)}_{\text{Applied force}}$$

$$(ms^2 + ds)\,Y(s) = F(s)$$

$$G_p(s) = \frac{1}{ms^2 + ds} = \frac{\frac{1}{m}}{s\left(s + \frac{d}{m}\right)}$$

Is this BIBO? → No. step input gives ramp output.

Don't need I control. Is P enough?

Let $G_p(s) = \frac{4}{s(s+2)}$ $\quad G_c(s) = K$

$$H(s) = \frac{\frac{4K}{s(s+2)}}{1 + \frac{4K}{s(s+2)}} = \frac{4K}{s^2 + 2s + 4K}$$

$$y_{ss} = H(j0) = \frac{4K}{4K} = 1$$ Tracks the step function!

P control is enough to stabilize the system, but not enough to put the poles wherever we want them (only one "knob" for two poles).

<u>PD Control:</u> Add a derivative term. Allows us to affect the damping using the controller.

$$G_c(s) = K_p + K_d s$$

$$X(s) = G_c(s)E(s) = (K_p + K_d s)E(s)$$

$$x(t) = K_p e(t) + K_d \frac{de(t)}{dt}$$

Could be accel. error if y(t) is velocity

But $e(t) = r(t) - y(t)$, so $\frac{de(t)}{dt} = \frac{dr(t)}{dt} - \frac{dy(t)}{dt} =$ "Velocity error"

Can also think of the integral $\int_0^t e(\tau)d\tau$ as cumulative error.

For the cart example:

$$H(s) = \frac{\frac{4}{s(s+2)}(K_p + K_d s)}{1 + \frac{4}{s(s+2)}(K_p + K_d s)} = \frac{4(K_p + K_d s)}{s^2 + 2s + 4K_d s + 4K_p}$$

$$H(s) = \frac{4(K_d s + K_p)}{s^2 + (2 + 4K_d)s + 4K_p}$$

Tracks perfectly and allows us to place the poles where we want them.

If we want $\zeta = 1$ and $\omega_n = 4$,

$$2 + 4K_d = 2\zeta\omega_n \qquad 4K_p = \omega_n^2$$

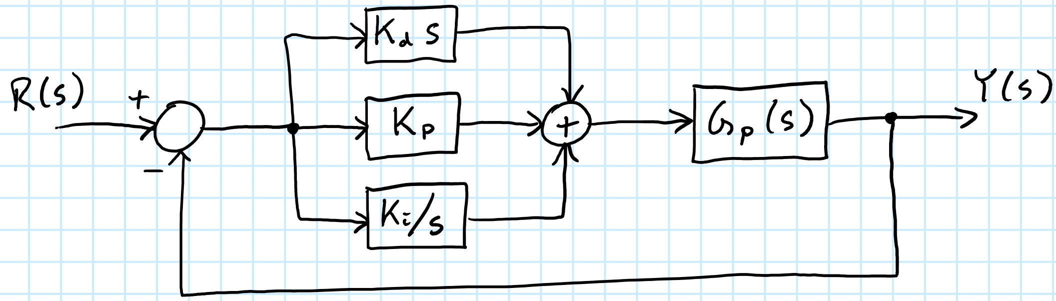
$$2 + 4K_d = 8 \qquad 4K_p = 16$$

$$K_d = \frac{3}{2} \qquad K_p = 4$$

Note: Taking derivatives can be dicey when there's noise, which tends to get amplified by the derivative. Typically, the input to the D part of the controller is first filtered to remove noise.

PID Control — Use all three terms

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$



More parameters $\Rightarrow$ More options, but also more difficult to set them well. Allows us to place 3 poles.

Most commercial controllers are PID

In practice, the gains (K_p, K_i, & K_d) are set by trial and error.

(Math can give a decent starting point, but we rarely know $G_p(s)$ exactly.)

1. Set them all to zero
2. $K_p \uparrow$ until oscillation, then back off a bit.
3. $K_i \uparrow$ until oscillation, then back off a bit.
4. Tweak K_d to reduce overshoot.
5. Repeat from step 2 until satisfied (or frustrated...).

Example: $G_p(s) = \frac{1}{s^2+9}$

Could use PD control to place both poles at -4:

$$H(s) = \frac{K_d s + K_p}{s^2 + K_d s + (9 + K_p)} \qquad (s+4)(s+4) = s^2 + 8s + 16$$

$$K_d = 8 \qquad 9 + K_p = 16 \Rightarrow K_p = 7$$

For $r(t) = u(t)$:

$$y_{ss} = H(j0) = \left.\frac{8s+7}{s^2+8s+16}\right|_{s=0} = \frac{7}{16} = 0.4375$$

Really bad at tracking a step even though we got to place the two poles where we wanted (stabilizing the system).

Use PID to get better tracking. But, need to decide where to place the third pole.

Could choose $s=-10$ to make it "fast" relative to the two poles at $s=-4$. That way, the double pole at $s=-4$ will largely dominate the system response.

Denominator: $(s+4)(s+4)(s+10) = (s^2+8s+16)(s+10)$

$$= s^3 + 18s^2 + 96s + 160$$

$$H(s) = \frac{\frac{K_d s^2 + K_p s + K_i}{s}\left(\frac{1}{s^2+9}\right)}{1 + \frac{K_d s^2 + K_p s + K_i}{s}\left(\frac{1}{s^2+9}\right)}$$

$$H(s) = \frac{K_d s^2 + K_p s + K_i}{s^3 + K_d s^2 + (9+K_p)s + K_i}$$

Match up terms

$K_d = 18 \qquad K_p = 87 \qquad K_i = 160$

For $r(t) = u(t)$

$$Y(s) = \frac{H(s)}{s} = \frac{18s^2 + 87s + 160}{s(s+10)(s+4)^2}$$

$$y_{ss} = H(j0) = \frac{160}{160} = 1$$

$$y(t) = \left(1 + C_1 e^{-10t} + C_2 e^{-4t} + C_3 t e^{-4t}\right) u(t)$$

This has to be 1 for y_{ss} to be 1.

What about tracking sinusoids? (Remember, "tracking" implies steady state.)

$r(t) = \cos(\omega_0 t) \qquad R(s) = \dfrac{s}{s^2+\omega_0^2}$

Two approaches:

1. $Y(s) = R(s)H(s) = \dfrac{A}{s+j\omega_0} + \dfrac{A^*}{s-j\omega_0} +$ other stuff

Transient since system is BIBO

$$y_{ss}(t) = 2|A|\cos(\omega_0 t + \angle A)$$

Magnitude may get scaled, delay may be introduced.

2. Treat system like a filter $H(j\omega)$ — Think frequency domain.

$$y_{ss}(t) = |H(j\omega_0)| \cos\left(\omega_0 t + \angle H(j\omega_0)\right)$$

Generally easier. Plug $s=j\omega_0$ in to $H(s)$