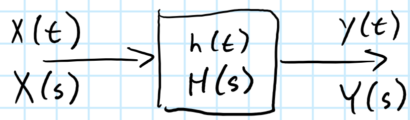
ECE3084-L17 Feedback / Intro to Control

Tuesday, April 11, 2017 4:09 AM

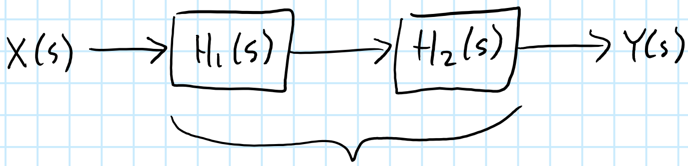
System block diagrams

$H(s) = \frac{Y(s)}{X(s)}$ Transfer function



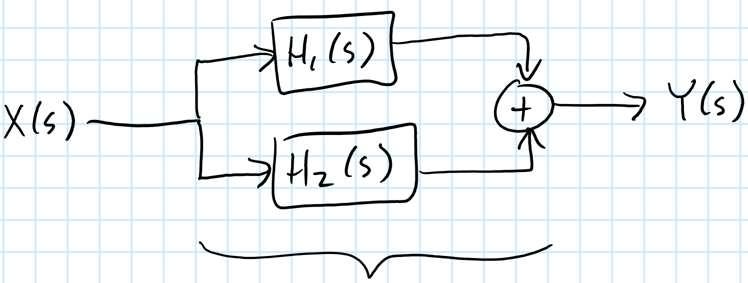
Can combine systems to make new ones:

Cascade:



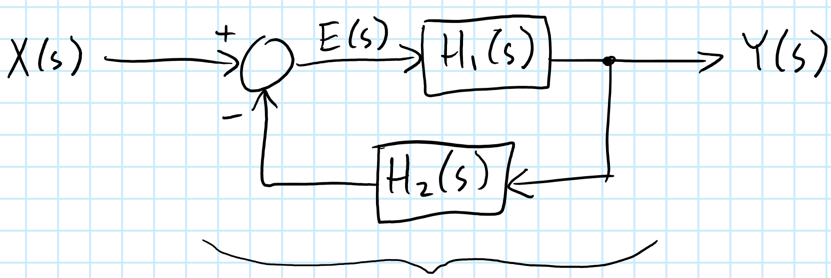
$$H(s) = H_1(s)\,H_2(s)$$

Parallel



$$H(s) = H_1(s) + H_2(s)$$

Feedback



$H(s) = ?$ Not as easy...

$$E(s) = X(s) - H_2(s)\,Y(s)$$

$$Y(s) = H_1(s)\,E(s) \implies E(s) = \frac{Y(s)}{H_1(s)}$$

Plug this in to eliminate $E(s)$.

$$\frac{Y(s)}{H_1(s)} = X(s) - H_2(s)\,Y(s)$$

$$\left(\frac{1}{H_1(s)} + H_2(s)\right) Y(s) = X(s)$$

$$\left(\frac{1 + H_1(s)H_2(s)}{H_1(s)}\right) Y(s) = X(s)$$

$$H(s) = \frac{Y(s)}{X(s)} = \frac{H_1(s)}{1 + H_1(s)\,H_2(s)}$$

Control

Terminology:

Plant – The system we're trying to control (we're usually stuck with it).

Input – Signal provided to the plant to get a desired response.

Output – The response from the plant.

$$x(t), X(s) \longrightarrow \boxed{G_p(s)}_{\text{Plant}} \longrightarrow y(t), Y(s)$$

Examples: Cruise control, temperature control, robotic control

Tracking control – How well does the output track or follow a reference signal, $r(t)$?

We specify: $r(t) = u(t)$ (set a speed, temperature, position, etc)

We want: $y(t) = r(t) = u(t)$

What does $x(t)$ have to be to produce this response?

$$Y(s) = G_p(s)\, X(s)$$ Could solve for $X(s)$

$$X(s) = \frac{Y(s)}{G_p(s)} = \frac{1}{s\, G_p(s)}$$

Example: $G_p(s) = \frac{k}{s+p}$

$$X(s) = \frac{1}{s\left(\frac{k}{s+p}\right)} = \frac{s+p}{sk} = \frac{1}{k} + \frac{p}{sk}$$

$$x(t) = \frac{1}{k}\delta(t) + \frac{p}{k}u(t)$$ Do you see any problems with this?

Have to "hit" the system with an impulse to get perfect tracking.

Not Feasible! (And likely to break something or cause non-linear effects in the attempt.)

Redefine our tracking goal: We want $y_{ss} = \lim_{t\to\infty} y(t) = 1$

i.e. The output will eventually get there, but maybe not right away.

What's a simple $x(t)$ that will do this?

$$x(t) = \frac{p}{k}\,u(t)$$

Just discard the δ fn from above.

Or, notice that the DC gain is $G_p(0) = \frac{k}{p}$

$$Y(s) = X(s)\,G_p(s)$$

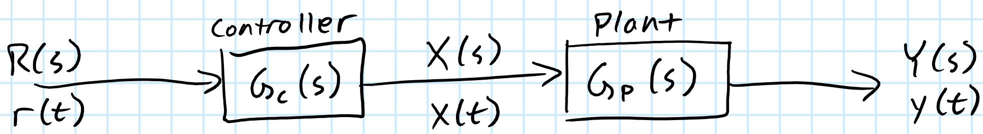
$$= \frac{p}{sk}\left(\frac{k}{s+p}\right) = \frac{p}{s(s+p)} = \frac{1}{s} - \frac{1}{s+p}$$

$$y(t) = (1 - e^{-pt})\,u(t)$$

$$y_{ss} = \lim_{t\to\infty} y(t) = \underline{1}$$

Perfect steady state tracking

Note that $r(t) = u(t)$ is in $x(t) = \frac{p}{k}u(t)$. Think of $\frac{p}{k}$ as a second system (the controller, $G_c(s) = \frac{p}{k}$) acting on the reference signal to generate the input, $x(t)$.



$$Y(s) = R(s)\,G_c(s)\,G_p(s)$$

What if we want it to converge faster or slower (i.e. change the pole location).

Let $G_c(s) = \dfrac{B(s+p)}{(s+\sigma)}$ $\qquad$ p = old pole, σ = new pole

$$Y(s) = \left(\frac{1}{s}\right)\left(\frac{B(s+p)}{(s+\sigma)}\right)\left(\frac{k}{s+p}\right) = \frac{kB}{s(s+\sigma)}$$

Using FVT: $\quad y_{ss} = \lim_{s\to 0} s\,Y(s) = \left.\dfrac{kB}{s+\sigma}\right|_{s=0} = \dfrac{kB}{\sigma}$

So, set $B = \frac{\sigma}{k}$ to get $y_{ss} = \underline{1}$

Gives $G_c(s) = \dfrac{\sigma(s+p)}{k(s+\sigma)}$

$$Y(s) = \frac{\sigma}{s(s+\sigma)} = \frac{1}{s} - \frac{1}{s+\sigma}$$

$$y(t) = (1 - e^{-\sigma t})\,u(t)$$

This is called "open loop control" $\longrightarrow$ The controller only looks at the reference signal to generate the input $x(t)$. It can't see what's happening on the output.

What happens if the real plant doesn't match our model (or if it changes over time)?

Example: Model $G_p(s) = \frac{4}{s+2}$

Designed $G_c(s) = \frac{2.5(s+2)}{s+10}$ to move the pole from $s=-2$ to $s=-10$.

If $r(t)=u(t)$ and model is accurate, $y(t) = (1-e^{-10t})u(t)$.

But if real $G_p(s) = \frac{4}{s+3}$ instead of $\frac{4}{s+2}$:

$$Y(s) = \frac{1}{s}\left(\frac{2.5(s+2)}{s+10}\right)\left(\frac{4}{s+3}\right) = \frac{10(s+2)}{s(s+3)(s+10)}$$

$$y_{ss} = \lim_{s\to 0} sY(s) = \left.\frac{10(s+2)}{(s+3)(s+10)}\right|_{s=0} = \frac{20}{30} = 0.667$$ Doesn't track!

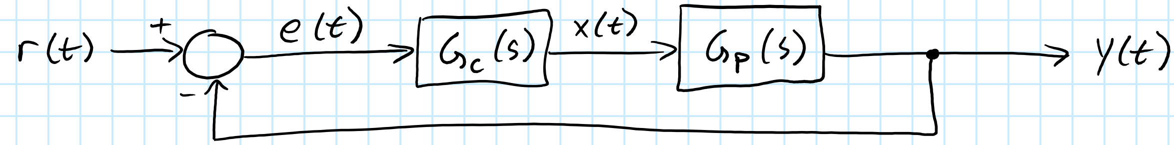
$$y(t) = \frac{2}{3}u(t) + C_2 e^{-3t}u(t) + C_3 e^{-10t}u(t)$$

Slow transient behavior shows up.

Open loop control only works if our model is accurate and the plant never changes.

Inaccuracies in the model also creates issues with inverting the transfer function to get $X(s) = \frac{Y(s)}{G_p(s)} = G_p^{-1}(s)Y(s)$. The zeros won't perfectly line up with the poles, and will thus fail to completely cancel them out.

More robust: Feedback control. Close the loop!



(can insert a sensor model here)

$e(t)$ = error (How far away the output is from where we want it.)

$$H(s) = \frac{G_c(s)\, G_p(s)}{1 + G_c(s)\, G_p(s)}$$

$Y = G_c G_p E = G_c G_p (R - Y)$

$E = R - Y$

$(1 + G_c G_p) Y = G_c G_p R$

Proportional control: $G_c(s) = K_p$ ← Just a constant. Makes $x(t) \propto e(t)$.

For $G_p(s) = \frac{k}{s+p}$

$$H(s) = \frac{\frac{k K_p}{s+p}}{1 + \frac{k K_p}{s+p}} = \frac{k K_p}{s + p + k K_p}$$

Pole has moved from $s = -p$ to $s = -(p + k K_p)$. The system is faster (could be good or bad).

$$y_{ss} = \lim_{s \to 0} sY(s) = \frac{k K_p}{p + k K_p} \rightarrow 1 \text{ for large } K_p$$

But, there are often physical limits on how large we can make K_p.

Example from before: $G_p(s) = \frac{4}{s+2}$

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

$$y_{ss} = \lim_{s \to 0} sY(s) = H(0) = \frac{4 K_p}{2 + 4K_p} = \frac{2K_p}{1 + 2K_p}$$

For $K_p = 50$, $y_{ss} = \frac{100}{101} = 0.99$

What if $G_p(s) = \frac{4}{s+3}$ (i.e. model inaccurate)

$$H(s) = \frac{4K_p}{s + 3 + 4K_p}$$

$$y_{ss} = \frac{4K_p}{3 + 4K_p} = \frac{200}{203} = 0.985 \text{ for } K_p = 50$$

Still good tracking even though the pole moved by 50%.

Closed loop is more robust to changes in the plant.

But for this system, we don't have perfect tracking.