

# Notes 19 largely plagiarized by %khc

## 1 Feedback

In the canonical feedback setup in Figure 1, we have  $G(s)$  in the forward path and  $H(s)$  in the feedback path. To find

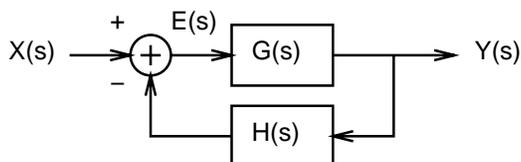


Figure 1: The canonical feedback setup.

the closed loop transfer function relating input  $X(s)$  to output  $Y(s)$ , we first write equations in terms of  $E(s)$  and then eliminate that variable.

$$\begin{aligned}
 E(s) &= X(s) - H(s)Y(s) \\
 Y(s) &= G(s)E(s) \\
 &= G(s)[X(s) - H(s)Y(s)] \\
 Y(s)[1 + G(s)H(s)] &= G(s)X(s) \\
 \frac{Y(s)}{X(s)} &= \frac{G(s)}{1 + G(s)H(s)}
 \end{aligned}$$

We now have a formula for attacking these sorts of systems.

## 2 Series and Feedback Compensation

As in Figure 2,  $G_p(s)$  represents the plant, the thing you're trying to control.<sup>1</sup> The plant is usually given to you, so you have no control over its dynamics. The only thing that you can do is design the compensator  $G_c(s)$ . The compensator can be placed in either the forward path or the feedback path. What are the relevant transfer functions?

<sup>1</sup>This is not the plant that grows outside. It originally described a chemical plant or another such industrial process.

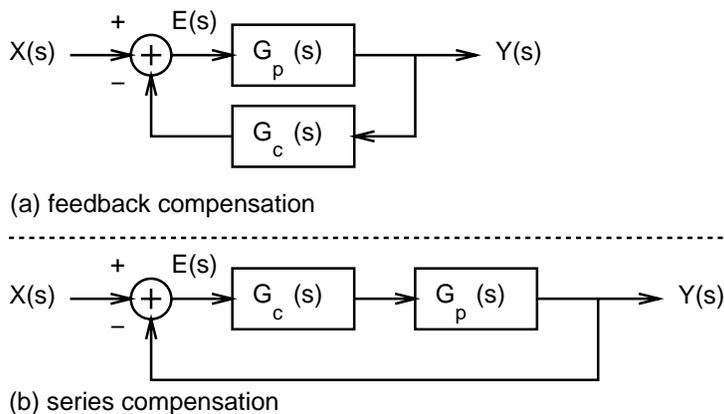


Figure 2: Two different compensation schemes.

For series compensation, using the notation of the previous section, we have  $G(s) = G_c(s)G_p(s)$  and  $H(s) = 1$ . Applying our standard formula, we get a transfer function of

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

At low frequencies, if  $G_c(s)G_p(s) \gg 1$ , this transfer function becomes

$$\begin{aligned} \frac{Y(s)}{X(s)} &\approx \frac{G_c(s)G_p(s)}{G_c(s)G_p(s)} \\ &\approx 1 \end{aligned}$$

At high frequencies,  $G_c(s)G_p(s)$  will usually be much less than 1 (in the real world<sup>TM</sup> we never get very much gain at high frequencies), so

$$\begin{aligned} \frac{Y(s)}{X(s)} &\approx \frac{G_c(s)G_p(s)}{1} \\ &\approx 0 \end{aligned}$$

For feedback compensation, we have  $G(s) = G_p(s)$  and  $H(s) = G_c(s)$ . Once again, applying our standard formula, we get a transfer function of

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

At low frequencies, if  $G_c(s)G_p(s) \gg 1$ , we get

$$\begin{aligned} \frac{Y(s)}{X(s)} &\approx \frac{G_p(s)}{G_c(s)G_p(s)} \\ &\approx \frac{1}{G_c(s)} \end{aligned}$$

At high frequencies, when  $G_c(s)G_p(s) \ll 1$ , we have

$$\frac{Y(s)}{X(s)} \approx G_p(s)$$

Why would we bother to choose one setup over another? Well, in general we're looking for a particular behavior at low frequencies. Assuming our overall system is stable, in feedback compensation we can shape this low frequency behavior with an appropriate choice of  $G_c(s)$ .

In series compensation, we're usually looking to have the output of our system  $Y(s)$  track a given input  $X(s)$ , so at low frequencies we want the transfer function to have unity gain in that range. We usually don't want the system to respond to very high frequencies, since noise is usually high frequency stuff. So that's why a transfer function which goes to 0 at high frequencies might be preferable.

### 3 Disturbance Rejection

Consider the system in Figure 3(a). Note that there are two inputs into the system:  $X(s)$  and  $D(s)$ . We can think of  $D(s)$  as some sort of disturbance injected into the system.

If we consider  $D(s) = 0$ , the transfer function relating  $X(s)$  to  $Y(s)$  is:

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

If we redraw the system as in Figure 3(b) and consider  $X(s) = 0$ , the transfer function relating  $D(s)$  to  $Y(s)$  is:

$$\frac{Y(s)}{D(s)} = \frac{G_p(s)}{1 + G_c(s)G_p(s)}$$

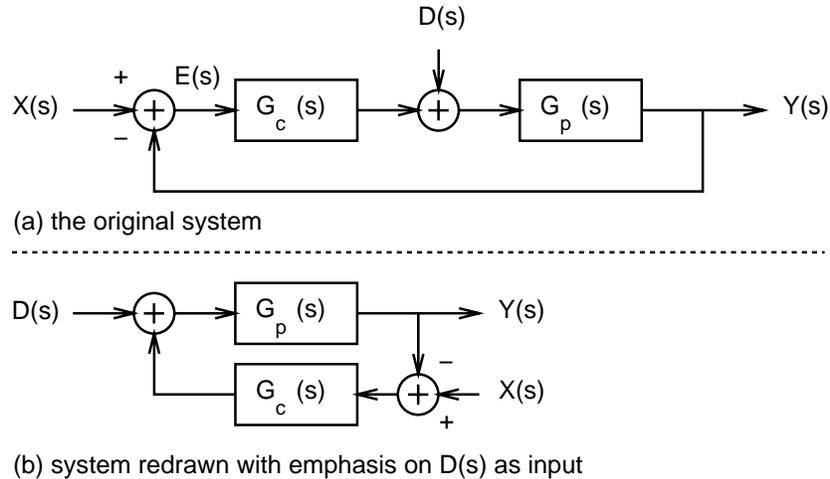


Figure 3: A disturbance injected into the plant.

Using superposition, we get:

$$Y(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)}X(s) + \frac{G_p(s)}{1 + G_c(s)G_p(s)}D(s)$$

What does this tell us? Well, if  $G_c(s)G_p(s) \gg 1$ , the output  $Y(s)$  will tend to track  $X(s)$ , while the dependence on  $D(s)$  will be attenuated by  $G_c(s)$ . In other words, in this configuration, disturbances will tend to be rejected. This is a feature.

## 4 Culture: State Space

In the state space approach to systems, we reduce a physical system to three vectors and four matrices.

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

The vectors  $u$  and  $y$  are the inputs to and the outputs of the system under study, while the vector  $x$  represents the states of the system. The states are the minimal amount of information needed to reproduce any subsequent system state, given the input  $u$  as a function of time and some initial state  $x_0$ .

The first equation relates the changes in state to the previous state and the current input, while the second gives a formula for the outputs that is dependent on both the state and the input.

The systems that we have been studying here can all be rewritten in state space form, with a single input  $u$  and a single output  $y$ . If you are really bored, you can ask me how (or if i am really bored in office hours, i probably will write it on the board at some point in time and promptly erase it).

## 5 Stability Analysis

The denominator of the transfer function in either series or feedback compensation is the same:  $1 + G(s)H(s)$ . Most of the single-input/single-output analysis concerns itself with figuring out where the roots of this equation go as a function of  $s$ , since if this is zero, the denominator of the transfer function is zero, and we have a pole at that particular value of  $s$ . Note that we usually wish to keep the poles in the open left half plane (OLHP) for stability purposes.

In general,  $1 + G(s)H(s)$  is a complex number. Solving the equation  $G(s)H(s) = -1$  [rewrite the entire equation in polar and equate magnitudes and phases] gives us two conditions on  $G(s)H(s)$ :

- $|G(s)H(s)| = 1$

- $\angle[G(s)H(s)] = \pm\pi$

Meeting these two conditions on magnitude and phase is equivalent to looking for the poles of  $1 + G(s)H(s)$ .

In the book we have one way of looking for these magnitude and phase conditions by plotting  $|G(s)H(s)|$  against  $\angle[G(s)H(s)]$ . Another way is to look at the bode plot of  $G(s)H(s)$ . Look for the crossover frequency  $\omega_{gm}$  on the phase plot where  $\angle[G(s)H(s)] = \pm\pi$ . Then look for  $\omega_{gm}$  on the magnitude plot. If  $|G(s)H(s)|$  at  $\omega_{gm}$  is less than 0dB, then you have positive gain margin [if you go over 0dB, then the gain at  $\omega_{gm}$  could drop such that the magnitude crosses through 0dB at  $\omega_{gm}$  and your system would self-destruct]. The larger the gain margin, the more likely your system is to be stable, but you could be sacrificing some performance instead.

Play the same game to find the phase margin. Look for the crossover frequency  $\omega_{pm}$  where  $|G(s)H(s)| = 0dB$ . Then look for  $\omega_{pm}$  on the phase plot. The amount of phase shift that you have to go through to get to the closest odd multiple of  $\pi$  is your phase margin. One way you can get a change in phase is to experience a time delay [ $x(t - T) \leftrightarrow X(s)e^{-sT}$ ]. See page 730 in the textbook for an example.

Why bother doing this anyway? Well, the gain and phase margin tell you how far away you are from watching everything blow up in your face. If the systems that you design have sufficient margin, then you won't be awoken late at night by irate phone calls from customer support. Getting uninterrupted sleep is definitely a feature.

Another way to attack stability is just to figure out where the closed loop poles are. If they end up in the right half plane or on the  $j\omega$  axis, your system will not be stable. To find out if those poles are in the right half plane or not, you can try to directly factor the denominator of the transfer function or you can use a computer [try the matlab `roots`]. For systems will order higher than two, the latter will probably be more useful to consider.

## 6 A Few Ad-Hoc Methods for Designing Controllers

For causal stable systems, the poles that are closer to the  $j\omega$  axis are the ones that dominate the time responses.

**Exercise** Why? Consider writing out the transfer function and taking the partial fraction expansion.

Say that i have a system with some number of poles and zeros, one pole of which is pretty close to the  $j\omega$  axis, which is slowing my step response down. i want to speed things up. Theoretically, if i take a zero and somehow manage to drop it exactly on top of that annoying pole, i can get rid of the term in the partial fraction expansion that corresponds to that slow poles. However, i probably won't be able to do that in real life<sup>TM</sup>, because i will get parameter variation [alas, beloved theory].

Let's consider parameter variation in more depth. Say that i have a closed loop transfer function  $\frac{1}{(s^2 + As + B)(s - (p + \delta))}$  and i somehow manage to add a zero at  $s = p + \epsilon$  to the overall closed loop transfer function. My new transfer function is  $\frac{s - (p + \epsilon)}{(s^2 + As + B)(s - (p + \delta))}$ . If i find the time response of this new system, i'm going to get an exponential  $e^{-(p + \delta)t} u(t)$  term. What's its coefficient though? By partial fraction expansion tricks, i'm going to end up with  $\delta - \epsilon$  multiplied by some other random stuff. If  $\epsilon$  and  $\delta$  are small, than i've effectively nuked that part of my time response. Cool!

Of course, it may not be terribly wise to nuke a pole in the right half plane this way.

**Exercise** Why? Consider what happens when my zero is off by  $\epsilon$ , as above.

Adding integrators will reduce steady state errors. For example, consider a plant [the thing you're trying to control] with transfer function  $G_p(s) = \frac{s+1}{s+2}$ . Let's place a proportional controller with transfer function  $G_c(s) = K$  in series compensation with the plant. Then, by application of the handy formula in Section 1 above, the closed loop transfer function is:

$$\frac{Y(s)}{X(s)} = \frac{K(s+1)}{(1+K)s + (2+K)}$$

As  $t \rightarrow \infty$ , for a unit step input, we have a final value of  $\frac{K}{2+K}$  by application of the final value theorem. This means that we have a steady state error of  $1 - \frac{K}{2+K}$  for a unit step input. As  $K$  gets large, this error gets smaller, but having a large gain in our controller means that we will have to throw in quite a bit of power into the system and have some pretty large control actions.

If we add an integrator to our controller, our new controller is  $G_c(s) = \frac{K}{s}$ . The new overall transfer function is then:

$$\frac{Y(s)}{X(s)} = \frac{K(s+1)}{s^2 + (2+K)s + K}$$

The steady state error then becomes 0.

**Exercise** For what values of  $K$  will the two systems above be stable? Consider putting the poles into the open left half plane.

## 7 Culture: Some Common Controllers

In series compensation, one popular family of controllers is commonly used. The controller takes as input the error between the output variable and the reference value, and generates a control action related to that error:

proportional	$G_c(s) = K_p$
proportional-integral (PI)	$G_c(s) = K_p + \frac{K_i}{s}$
proportional-derivative (PD)	$G_c(s) = K_p + K_d s$
proportional-integral-derivative (PID)	$G_c(s) = K_p + \frac{K_i}{s} + K_d s$

So the P controller generates a control action that is linearly proportional to the error, the PI controller generates a control action that is linearly proportional to the error and the integral of the error, and so on.

Proportional controllers sometimes suffer from the problem that they have steady state error. Adding an integrator will reduce steady state error; if there is a large error going into the controller, the integrator will cause the controller output to be large, and this large input into the plant will hopefully reduce the plant output. Adding a differentiator will allow the system to react more quickly to changes in the error, since a differentiator will cause a large controller output in response to a large change in the error signal.