ME 4232: Fluid Power Control Laboratory

University of Minnesota
Prof. Perry Y. Li

Lab.21: Adaptive control

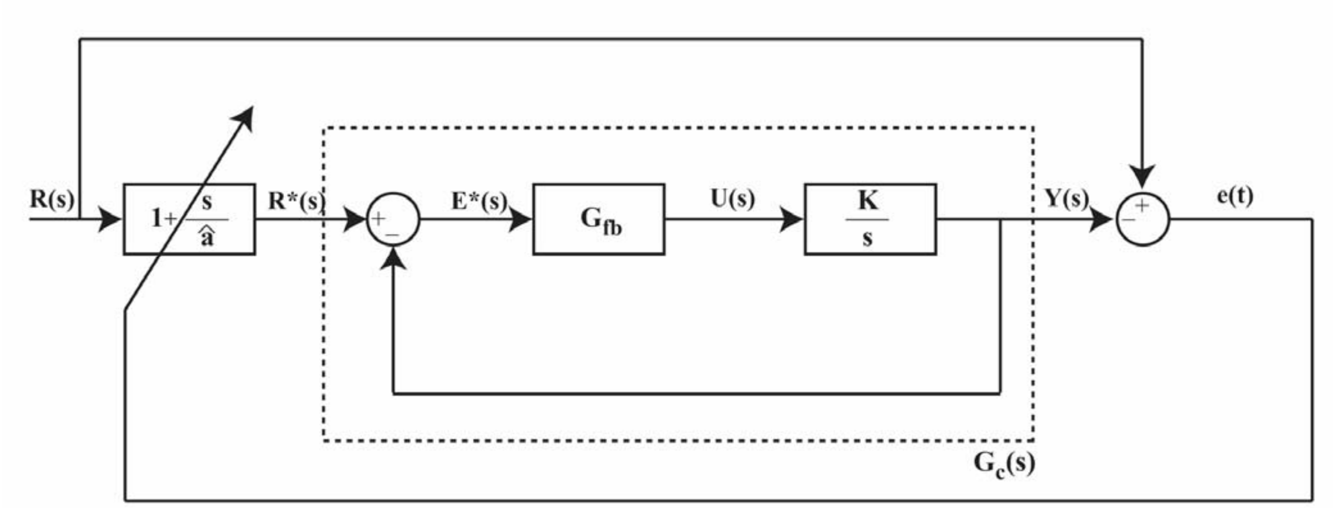
Objective

We saw in Lab 19 (feed-forward control), that for feed-forward controllers to work well, it is important to have a good estimate of the break frequency of the closed loop system $G_c(s)$, of which the feed-forward control $G_{ff}(s)$ is supposed to be the multiplicative inverse. One possibility is to identify the $G_c(s)$ well before proceeding to the design of the feed-forward controller. It is also possible to improve on the $G_{ff}(s)$ by first measuring how well it is performing and then figuring out how parameters in $G_{ff}(s)$ should be modified to offset the deviation in the desired behavior.

Another possibility, which is pursued in this lab, is to enable the controller to determine for itself what is the best parameter for $G_{ff}(s)$. In other words, $G_{ff}(s)$ will configure itself automatically to adapt to the plant being controlled.

Adaptive feed-forward control

Adaptive feed-forward control scheme is shown in the figure below. Let $G_{fb}(s) = K_p$ be simply a proportional controller.



Then,

$$G_c(s) = \frac{Y(s)}{R_2(s)} = \frac{a}{s+a}$$

Where $a = KK_p$. Of course, if K is well known, then your estimate of a will be precise. Notice that,

$$\dot{y}(t) = -ay(t) + a \cdot r_2(t)$$

As in lab 19, if your estimate of a is â, then your feed-forward controller would be given by:

$$G_{ff}(s) = 1 + \hat{b} \cdot s$$

Where, $\hat{b} = \frac{1}{\hat{a}}$. Or, written in the time domain,

$$r_2(t) = r(t) + \hat{b} \cdot \dot{r}(t)$$

If $\hat{b} = \frac{1}{a}$ (i.e., a is well known), then the controller would cause y(t) → r(t) with no steady state error as we have seen in Lab 19 when we tested the feed-forward controller for various cases.

The adaptive feed-forward controller uses the same control structure, except that it determines $\hat{b}$ on its own using an adaptation rule:

$$\frac{d}{dt}\hat{b} = \gamma \dot{r}(t)e(t)$$

Where $\gamma > 0$ is a positive learning gain, and *e(t) = r(t) – y(t)* is the tracking error.
To see that this adaptation law makes sense, let $\tilde{b} = \hat{b} - \frac{1}{a}$ be the parameter error, and combine the above equations

$$\dot{y}(t) = -a \cdot y(t) + a \cdot r(t) + a\hat{b}\dot{r}(t)$$

But, $a\hat{b} = a(\tilde{b} + {}^{1}/_{a}) = a\tilde{b} + 1$, and *e(t) = y(t) – r(t)*, so

$$\dot{y}(t) = -a \cdot y(t) + a \cdot r(t) + (a\tilde{b} + 1)\dot{r}(t)$$
$$\dot{e}(t) = -a \cdot e(t) - a\tilde{b}\dot{r}(t)$$

This differential equation tells us that if $\tilde{b} = 0$, then e(t) → 0. Suppose that $\tilde{b} > 0$, i.e. your estimate of $\hat{b} > {}^{1}/_{a}$ (or your estimate of a is too low), and also that $\dot{r} > 0$, then since $a > 0$, this will cause $\dot{e}$ to decrease, and *e(t)* will also likely be negative. On the other hand, for the same $\tilde{b}$, but with $\dot{r} < 0$, both $\dot{e}(t)$ and $e(t)$ will likely be positive. Therefore when $\dot{r}(t) > 0$, $e(t) < 0$ is an indication that $\hat{b}$ is too large, and consequently, $\hat{b}$ should be decreased. Similar logic can be deduced for $e(t) > 0$ and for $\dot{r} < 0$, so that if $\dot{r}(t)$ and $e(t)$ are of the same sign, the $\hat{b}$ is too small; and if they are of different signs, then $\hat{b}$ is too large. Notice that these logic rules are consistent with the adaptation law. A rigorous analysis is however beyond the scope of our present course.

Procedure

1. Design $G_c(s)$ with a reasonable break frequency, say 10rad/s.

2. Simulate a feed-forward controller designed based on your model, and test its performance using the signal $r(t) = \alpha \sin(\omega_1 t) + \beta \cos(\omega_2 t)$ where ω_1 and ω_2 are some known frequencies. Once you got the sinusoid to work, you may also try using a triangular signal if you have time.

3. Test the performance when you deliberately change K of your plant (thus $G_c(s)$).

4. Reconfigure your feed-forward controller so that the parameter $\hat{b}$ is adapted online (should not be much bigger than 1). Use the integrator block to implement this part.

5. Test the adaptive controller in simulation. You can initialize the integrator to some initial estimate $\hat{b}(t = 0) = {}^{1}/_{\hat{a}}$. Observe the estimate $\hat{b}$ and compare it to what you think it should be.

6. Test the adaptive controller in implementation. Observe the estimate $\hat{b}(t)$. How does your initial estimate of $\hat{b}$ and adaptation gain affect the performance?

7. Tune your controller, K_p and γ so that you get the “best” performance.

Report

There is **no report** for this lab (However, attendance will contribute towards the overall participation grade).