Steady-State Errors

Sources of Steady-State Errors

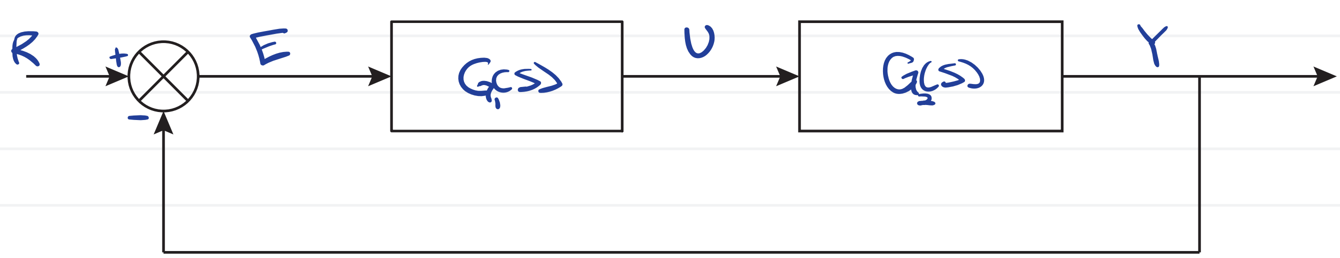
Steady-state errors arise from three primary sources:
(1) nonlinearities, like backlash in gears -- we won't explore this one;
(2) disturbances, like those from the environment -- see Nise 7.5; &
(3) input (command) type and the plant dynamics.
We will focus our attention on (3). (2) is similar.

Steady-State Errors for Unity Feedback Systems

It is uncommon for a feedback system to be truly "unity." However, as shown in Nise Section 7.6, nonunity feedback systems can be re-written and evaluated in terms of unity feedback counterparts. For this reason, we will focus on unity feedback systems.

First we recall the final value theorem. Let f(t) be a function of time that has a "final value" $f(\infty) = \lim_{t \to \infty} f(t)$. Then, from the Laplace transform of f(t), F(s), the final value is $f(\infty) = \lim_{s \to 0} s F(s)$.

Let's consider a unity feedback system.



Recall that we call e(t) or (its Laplace transform) E(s) the error. We want to know the steady-state error $e(\infty) = \lim_{t \to \infty} e(t)$. From the final value theorem, $e(\infty) = \lim_{s \to 0} s E(s)$. Now all we need is to express E(s) in more convenient terms. For the analysis that follows, we combine the controller and plant: $G(s) = G_1(s)G_2(s)$.

Given a command R(s) and forward-path t.f. G(s), we could take the inverse Laplace transform of $E(s)$ to find $e(t)$ and take the limit. However, it's much easier to use the final value theorem:

We can only explore further given a specific command $R(s)$. Three different commands are typically considered canonical. The first is now developed in detail, and the results of the other two are given below. First, consider a unit step command.

Let's give K_p a name: the position constant. If K_p is large, the steady-state error is small. If K_p is infinitely large, the steady-state error is zero. If K_p is small, the steady-state error is a finite constant.

The form of $G(s)$ has implications for K_p. $G(s)$ has a factor $1/s^n$ where n is some nonnegative integer. Since we are concerned about what happens to $G(s)$ when we take its limit as $s \to 0$, this factor is of particular importance. If $n > 0$, $K_p = \lim_{s \to 0} G(s) = \inf$. Notice that we are able to know n from the forward-path t.f. $G(s)$ because it is a property of the system to which the input is applied. We call the t.f. $1/s$ an integrator, which is the inverse of the t.f. s, the differentiator.

We needn't solve for $E(s)$ explicitly, then. All we need to know is the command $R(s)$ and the number of integrators in the forward-path t.f. $G(s)$ (we call this the system type). The steady-state error for other other commands and system type can be derived in the same manner. The results for a few common inputs are shown. (Nise)

TABLE 7.2 Relationships between input, system type, static error constants, and steady-state errors

		Type 0		Type 1		Type 2	
Input	**Steady-state error formula**	**Static error constant**	**Error**	**Static error constant**	**Error**	**Static error constant**	**Error**
Step, $u(t)$	$\frac{1}{1+K_p}$	$K_p = \text{Constant}$	$\frac{1}{1+K_p}$	$K_p = \infty$	0	$K_p = \infty$	0
Ramp, $tu(t)$	$\frac{1}{K_v}$	$K_v = 0$	∞	$K_v = \text{Constant}$	$\frac{1}{K_v}$	$K_v = \infty$	0
Parabola, $\frac{1}{2}t^2u(t)$	$\frac{1}{K_a}$	$K_a = 0$	∞	$K_a = 0$	∞	$K_a = \text{Constant}$	$\frac{1}{K_a}$

The constants $K_v = \lim_{s \to 0} s\,G(s)$ and $K_a = \lim_{s \to 0} s^2\,G(s)$ are called the velocity and acceleration constants, respectively. Collectively, K_p, K_v, and K_a are called the static error constants.

Let's do the following example from Nise.

1. For the unity feedback system shown in Figure P7.1, where

WileyPLUS
WPCS
Control Solutions

$$G(s) = \frac{450(s+8)(s+12)(s+15)}{s(s+38)(s^2+2s+28)}$$

find the steady-state errors for the following test inputs: $25u(t)$, $37tu(t)$, $47t^2u(t)$. [Section: 7.2]

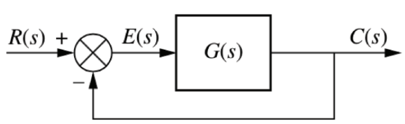


FIGURE P7.1