steady.error Steady-state error for unity feedback systems

It is uncommon for a feedback system to be truly "unity." However nonunity feedback systems can be re-written and evaluated in terms of unity feedback counterparts.[1] For this reason, we will focus on unity feedback systems.

1. For more details, see Nise. (Norman S. Nise. **Control Systems Engineering**. Sixth. John Wiley & Sons, Inc., 2011, Section 7.6)

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 $f(\infty) = \lim_{s\to 0} sF(s)$.

final value theorem

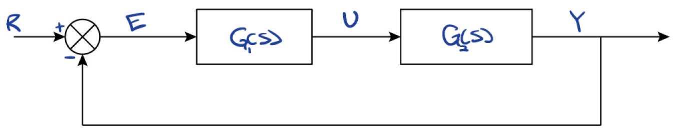

Figure error.1: unity feedback block diagram with controller $G_1(s)$ and plant $G_2(s)$.

Let's consider the unity feedback system of Figure error.1 with command R, controller transfer function G_1, plant transfer function G_2, and error E. Recall that we call $e(t)$ or (its Laplace transform) $E(s)$ the error. We want to know the steady-state error, which, from the final value theorem, is

$$e(\infty) = \lim_{s\to 0} sE(s). \tag{1}$$

Now all we need is to express $E(s)$ in more convenient terms. For the analysis that follows, we combine the controller and plant: $G(s) = G_1(s)G_2(s)$. From the block diagram, we can develop the transfer function from the command R to the error E.

Equation 2 error transfer function

Given a specific command R and forward-path transfer function G, we could take inverse Laplace transform of E(s) to find e(t) and take the limit. However, it is much easier to use the final value theorem:

This last expression is the best we can do without a specific command R. 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, which has Laplace transform $R(s) = 1/s$.

where we let $K_p = \lim_{s \to 0} G(s)$. We call K_p 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.

position 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 $n > 0$, $K_p = \lim_{s\to 0} G(s) = \infty$. We call the transfer function $1/s$ an **integrator**, which is the inverse of the transfer function s, the **differentiator**.

integrator

differentiator

We needn't solve for E explicitly, then. All we need to know is the command R and the number of integrators n in the forward-path transfer function $G(s)$ (we call this the **system type**).

system type

The steady-state error for other commands and system type can be derived in the same manner. The results for the canonical inputs are shown in Table error.1.

Example steady.error-1

re: steady-state error

Let a system have forward-path transfer

Table error.1: the static error constants and steady-state error for canonical commands $r(t)$ and systems of Types 0, 1, 2, and n (the general case). Note that the faster the command changes, the more integrators are required for finite or zero steady-state error.

	Type n		Type 0		Type 1		Type 2	
$r(t)$	**error const.**	$e(\infty)$	**error const.**	$e(\infty)$	**error const.**	$e(\infty)$	**error const.**	$e(\infty)$
$u_s(t)$	$K_p = \lim\limits_{s\to 0} G(s)$	$\dfrac{1}{1+K_p}$	K_p	$\dfrac{1}{1+K_p}$	∞	0	∞	0
$tu_s(t)$	$K_v = \lim\limits_{s\to 0} sG(s)$	$\dfrac{1}{K_v}$						
$\dfrac{1}{2}t^2u_s(t)$	$K_a = \lim\limits_{s\to 0} s^2G(s)$	$\dfrac{1}{K_a}$						

function

$$G(s) = \frac{10(s+3)(s+4)}{s(s+1)(s^2+2s+5)}.$$

For commands $r_1(t) = 2u_s(t)$, $r_2(t) = 6tu_s(t)$, and $r_3(t) = 7t^2u_s(t)$, what are the steady-state errors?