ELEC3004 Open Lecture 19/05/2015

Kerri Wait, ELEC3004 Tutor, BEng(Hons) MIEEE

19/05/2015

1 Block Diagram Simplification

Let us consider a system, represented by:

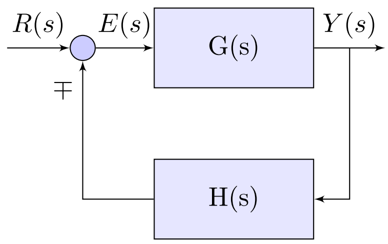

Figure 1: Closed loop form.

Let us consider another system, represented by:

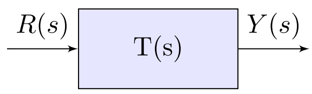

Figure 2: Open loop form.

Where:

$$T(s) = \frac{G(s)}{1 \pm G(s)H(s)}$$

These two representations both describe the behaviour of the system in its entirety. Pay careful attention to the minus plus ($\mp$) in the system block diagram (Figure 1), and note that it corresponds to the plus minus ($\pm$) in $T(s)$. H(s) is representative of the feedback in the system, and is typically negative feedback.

The open loop form is especially useful for determining the output of a system given an input, as the math can often be simplified. There are many resources that discuss simplification of system block diagrams with complex feedback structures, including an excellent wiki-book and most Control Systems textbooks.

2 Steady State Error

Consider a generic system such as that given below:

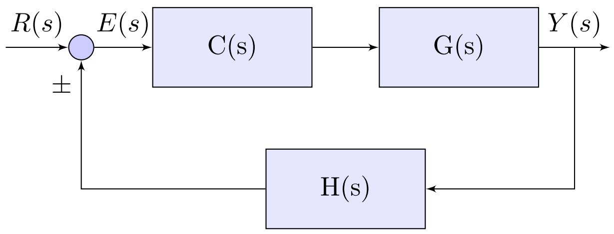

Figure 3: Generic system model.

This system can be consolidated as follows:

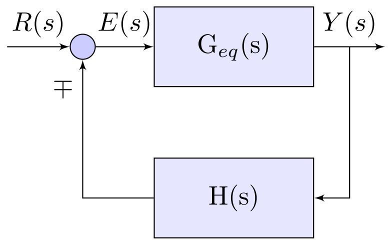

Figure 4: Consolidated system model.

Where $G_{eq}(s) = C(s)G(s)$.

Now let us consider the error:

$$E(s) = R(s) - Y(s) \tag{1}$$

But we know that:

$$Y(s) = E(s)G_{eq}(s) \tag{2}$$

If we substitute 2 into 1:

$$E(s) = R(s) - E(s)G_{eq}(s) \tag{3}$$

$$E(s)\left[1 + G_{eq}(s)\right] = R(s) \tag{4}$$

$$E(s) = \frac{R(s)}{1 + G_{eq}(s)} \tag{5}$$

We are interested in the value of $e(t)$ at infinity, which, thanks to final value theorem, can be expressed as:

$$e(\infty) = \lim_{e \to \infty} e(t) = \lim_{s \to 0} sE(s) \tag{6}$$

Substituting 5 into 6 yields:

$$\boxed{e(\infty) = \lim_{s \to 0} \frac{sR(s)}{1 + G_{eq}(s)}} \tag{7}$$

This is a useful form to express the steady state error in, as it allows us to vary both the input $R(s)$ and the system $G_{eq}(s)$ and observe the impact that this has on the steady state error value. Let us now consider the step, ramp, and parabolic inputs, and their s-domain representations (Table 1). Problem Set 4 Question 3 involves substituting each of these input signals into 7 and observing the result for various system types.

Input	Name	Time Domain	S Domain
Step	$R_{step}(s)$	$u(t)$	$\frac{1}{s}$
Ramp	$R_{ramp}(s)$	$tu(t)$	$\frac{1}{s^2}$
Parabola	$R_{parabola}(s)$	$\frac{1}{2}t^2u(t)$	$\frac{1}{s^3}$

Table 1: Input signals in the time and s domains.

3 System Type

Consider the generic system

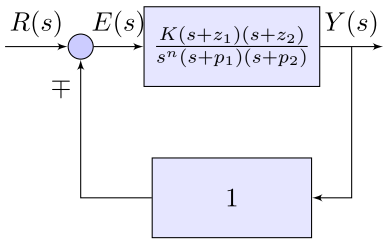

Figure 5: Closed loop form.

The system type is defined as **the number of pure integrators in the forward path**. For example, the generic transfer function

$$\frac{K(s+z_1)(s+z_2)}{s^n(s+p_1)(s+p_2)} \tag{8}$$

is representative of a *Type* n system. Note that K is a gain term, z_i is the ith zero, and p_j is the jth pole. Examples of Type 0, I, and II systems are given in Table 2.

System Type	Example Transfer Function
0	$\frac{500(s+2)(s+5)(s+6)}{(s+8)(s+10)(s+12)}$
I	$\frac{500(s+2)(s+5)(s+6)}{s(s+8)(s+10)(s+12)}$
II	$\frac{500(s+2)(s+5)(s+6)}{s^2(s+8)(s+10)(s+12)}$

Table 2: Examples of system types.

Note that the system type is a useful parameter, especially when combined with the steady state error table you will populate in Question 3. This framework allows for rapid evaluation of the steady state transient response of a system for a given input without the need for solving system equations or any further analysis.

4 MATLAB's control system toolbox

These commands should come in handy for Problem Set 4 Question 5:

Command	Typical Syntax
`tf`	`Gs = tf(numerator, denominator)`
	*numerator and denominator are polynomials
`feedback`	`Ts = feedback(Gs, Hs)`
`step`	`step(system)`
`poly`	`numerator = poly(roots)`

Table 3: Useful MATLAB commands.

For example, the following code will plot the step response of the open loop form of the unity feedback systems with $n = 0, 1, 2$ where:

$$G_n(s) = \frac{500(s+2)(s+5)}{s^n(s+8)(s+10)(s+12)}$$

Note that as this system is expressed using the roots of the function, rather than a polynomial, it is necessary to use the `poly` command to convert the numerator and denominator to polynomials prior to calling `tf`. The results of this code are shown in Figure 6.

Listing 1: MATLAB Code for Step Responses

```
clear all; close all; clc

% Define the numerator and denominator(s)
numg  = 500*poly([-2 -5 ]);
deng0 = poly([-8 -10 -12]);
deng1 = poly([0 -8 -10 -12]);
deng2 = poly([0 0 -8 -10 -12]);

% Create Gns
g0s = tf(numg, deng0);
g1s = tf(numg, deng1);
g2s = tf(numg, deng2);

% Create Ts
t0s = feedback(g0s, 1);
t1s = feedback(g1s, 1);
t2s = feedback(g2s, 1);

% Step responses!
step(t0s, t1s, t2s), legend('Type 0', 'Type I', 'Type II'), grid on
```

Sanity check: Does the steady state error value for the Type 0 system match the value in your table from Question 3? Hint: $K_p = \lim_{s\to 0} G(s)$, $e_{step}(\infty) = \frac{1}{1+K_p}$.

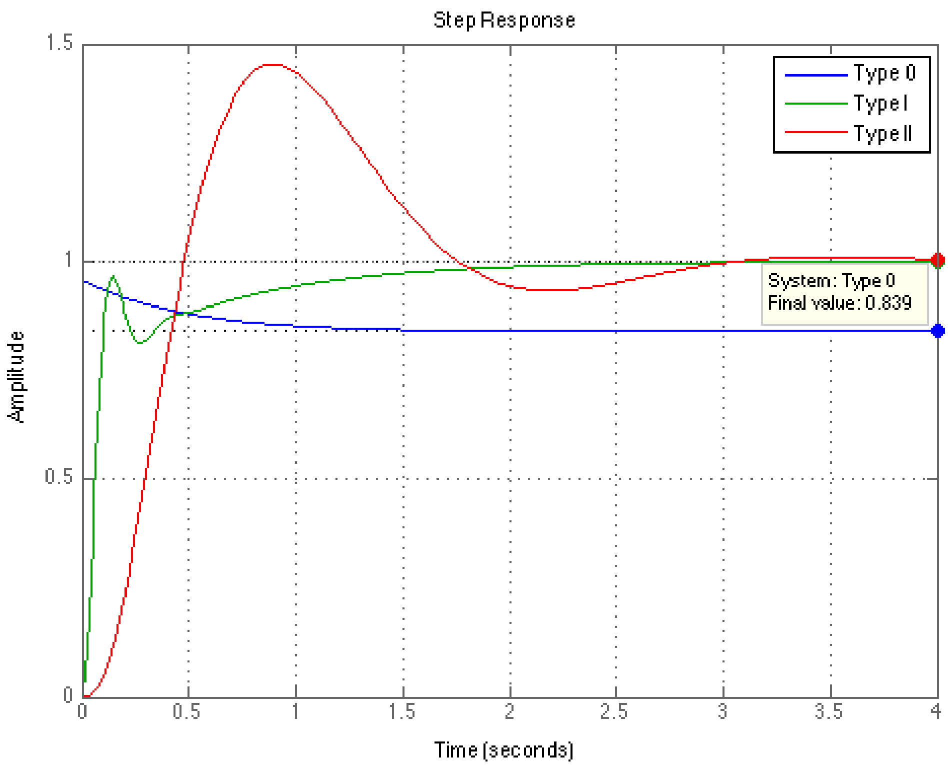

Figure 6: Step responses generated by MATLAB example code.