

Appendix W

Sensitivity of Time Response to Parameter Change

We have considered the effects of errors on the steady-state gain of a dynamic system and have shown how feedback control can reduce these errors. Since many control specifications are in terms of the step response, the sensitivity of the time response to parameter changes is sometimes very useful to explore. For example, by looking at the sensitivity plot we can tell whether increasing a particular parameter will increase or decrease the overshoot of the response.¹ The analysis that follows is also a good exercise in small-signal linearization.

To consider the sensitivity of the output $y(t, \theta)$ of a system having a parameter of interest, θ , we compute the effect of a perturbation in the parameter, $\delta\theta$, on the nominal response by using the Taylor's series expansion

$$y(t, \theta + \delta\theta) = y(t, \theta) + \frac{\partial y}{\partial \theta} \delta\theta + \dots \quad (\text{W.1})$$

The first-order approximation of the parameter perturbation effect is the term

$$\delta y(t) = \frac{\partial y}{\partial \theta} \delta\theta. \quad (\text{W.2})$$

This function can be generated from the system itself as shown by Perkins et al, 1991. We assume that the response depends linearly on the parameter and therefore that the overall transfer function $\mathcal{T}(s, \theta)$ is composed of component transfer functions that can be defined to bring out the dependence on the parameter explicitly. A block diagram of the transfer function in terms of the components $T_{ij}(s)$ can be expressed as shown in Fig. W.1, where we have labeled the parameter as θ and its input signal as Z . In terms of this block diagram, the equations relating Y and Z to the reference input can be written immediately:

$$Y = T_{11}R + T_{21}\theta Z \quad (\text{W.3})$$

and

$$Z = T_{12}R + T_{22}\theta Z. \quad (\text{W.4})$$

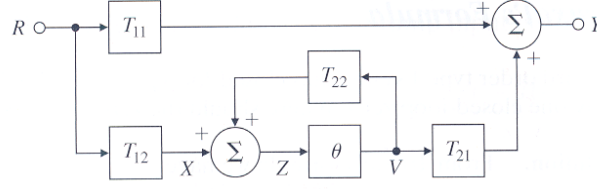
The perturbed equations are

$$Y + \delta Y = T_{11}R + T_{21}(\theta + \delta\theta)(Z + \delta Z) \quad (\text{W.5})$$

and

$$Z + \delta Z = T_{12}R + T_{22}(\theta + \delta\theta)(Z + \delta Z). \quad (\text{W.6})$$

¹As we shall see in the next chapter, the development of the MATLAB root locus interface `rltool` gives the designer a computer aid to this result.

Figure W.1: Block diagram showing the dependence of output y on parameter θ

Multiplying these out and ignoring the small term $\delta\theta\delta Z$, the expressions for the perturbations in Y and Z are given by

$$\delta Y = T_{21}(Z\delta\theta + \theta\delta Z) \quad (\text{W.7})$$

and

$$\delta Z = T_{22}(Z\delta\theta + \theta\delta Z). \quad (\text{W.8})$$

The solutions to these equations can be best presented as a block diagram, shown in Fig. W.2(a). The output of this figure is $\delta Y = \frac{\partial y}{\partial \theta} \delta \theta$, and we notice that the input Z is multiplied by a gain of $\delta \theta$. Therefore, if we drop the block $\delta \theta$, the output will be simply $\frac{\partial y}{\partial \theta}$, as shown in Fig. W.2(b). Finally, to compute the sensitivity as the variation to a percent change in the parameter, which is $\frac{\partial y}{\partial \ln \theta} = \frac{\frac{\partial y(\epsilon, \theta)}{\partial \theta}}{\frac{\partial \ln \theta}{\partial \theta}} = \theta \frac{\partial y}{\partial \theta}$, we need only shift the input Z from the output side of the θ block to its input as shown in Fig. W.2(c). We are now in a position to give the final block diagram of the system as it is to be implemented, shown in Fig. W.3.

In this figure, it is clear that to compute the sensitivity of the output to a parameter, one needs to simulate two copies of the system. The input to the first system is the reference input of interest, and the input to the second system is at the input to the parameter of interest of the variable Z , taken from the input to the parameter in the original system. The transfer function from the reference input to the output sensitivity is readily computed to be

$$\frac{T_{12}\theta T_{21}}{(1 - \theta T_{22})^2}. \quad (\text{W.9})$$

Theorem W.1 *Response sensitivity* From this function it is clear that to keep the sensitivity of the output signal to a parameter change low, it is important to have feedback with high gain around the parameter in question.

Example W.1 *Time-Domain Sensitivity* Compute the sensitivity of the output of the speed-control example described by the plant

$$G(s) = \frac{A}{\tau s + 1}$$

with respect to the proportional control gain, K_{cl} . Take the nominal values to be $K_{cl} = 9$, $\tau = 0.01$ sec, and $A = 1$ rad/volt-sec.

SOLUTION The required block diagram for the computation is given in Fig. W.4, based on Fig. W.3. In MATLAB, we will construct the several transfer functions with $T_{ij} = \frac{n_{ij}}{d_{ij}}$ and will implement Eq. (W.9). For comparison, we compute the nominal response from Fig. W.1 and add 10% of the sensitivity to the nominal response. The instructions to do the computation in MATLAB are

```
% script to compute sensitivity for Fig. W.4
% First input the data for the component transfer functions T_ij
% and the nominal parameter, Kcl for this problem
```

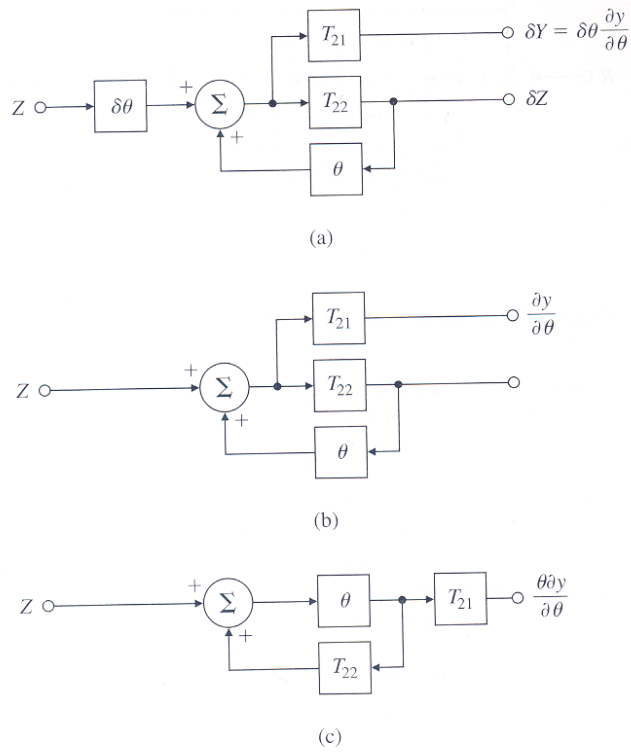


Figure W.2: Block diagrams showing the generation of (a) δY and δZ , (b) $\frac{\partial y}{\partial \theta}$, and (c) $\theta \frac{\partial y}{\partial \theta}$

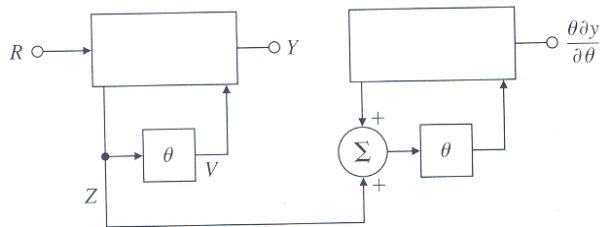


Figure W.3: Block diagram showing the computation of $\theta \frac{\partial y}{\partial \theta}$ from the original transfer function

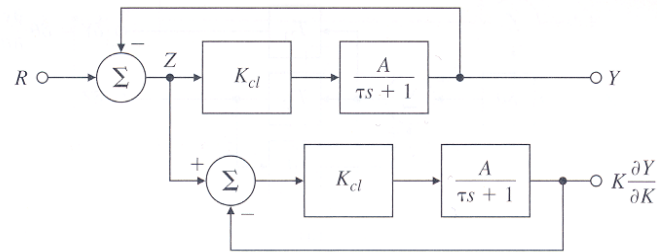


Figure W.4: Block diagram showing the computation of the sensitivity of the output of the speed-control example

```

Kcl = 9; tau = .01;
n11 = 0; d11 = 1;
n12 = 1; d12 = 1;
n22 = [0 -1]; d22 = [tau 1];
n21 = 1; d21 = [tau 1];
% Now compute the numerator and denominator polynomials of the transfer
% functions using the convolution function conv to multiply the polynomials
% and put them into system transfer function forms with the MATLAB function tf.
% The overall transfer function is
% Y/R = n11/d11 + (n12*n21*d22)/(d12*d21*[d22-Kcl*n22]) = sysy
% The transfer function from the reference input to the sensitivity is
% Kcl*(dy/dKcl)/R = sysdy
% Now define the numerators and denominators of several intermediate
% intermediate transfer functions
n1 = Kcl*conv(n21,n12);
d1 = conv(d21,d12);
n2 = d22;
d2 = [d22-Kcl*n22];
ny = conv(n1,n2);
dy = conv(d1,d2);
% Now put these together to form two intermediate transfer functions
sysy1 = tf(ny,dy);
sysy2 = tf(n11,d11);
% Now construct the final transfer functions
% The overall transfer function Y/R
sysy = sysy1+sysy2;
% The sensitivity transfer function
ndy = conv(ny,n2);
ddy = conv(dy,d2);
sysdy = tf(ndy,ddy);
% Now use these to compute the step responses and
% plot the output, the sensitivity and a perturbed response
[y,t]= step(sysy);
[yd,t]= step(sysdy);
plot(t,[y yd y+.1*yd]);

```

These instructions are constructed to compute the sensitivity for any system, given the several transfer functions. The script input is for the specific example. Plots of the output, its sensitivity, and the result of a 10% change in the parameter value are given in Fig. W.5.

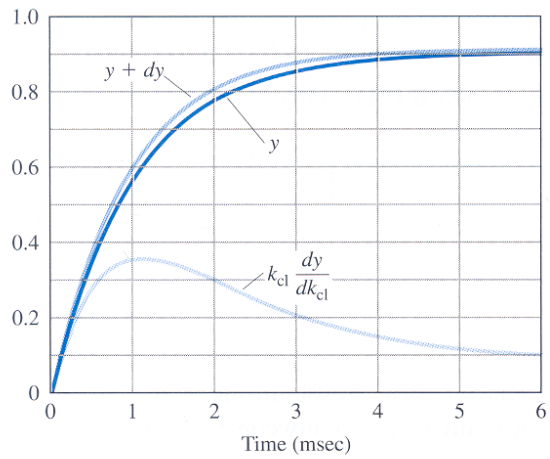


Figure W.5: Plots of the output, the sensitivity, and the result of a 10% change in the parameter value for the speed-control example