2.11 Robustness

Robustness of a control system is related to its *sensitivity* to *unmodeled dynamics*, i.e. part of the behavior of an actual control system which is *not* included in the mathematical model of the control system (governing differential equations, transfer function, etc.). Since we have to deal with actual control systems in which it is impossible (or difficult) to mathematically model *all* physical processes, we are always left with the question: will the control system based upon a mathematical model really work? In Chapter 1, we saw that *disturbances (or noise)* such as road roughness, tyre condition, wind velocity, etc., cannot be mathematically modeled when discussing a control system for the car driver example. If a control system meets its performance and stability objectives in the presence of all kinds of expected *noises* (whose mathematical models are *uncertain*), then the control system is said to be *robust*. Hence, robustness is a desirable property that dictates whether a control system is immune to uncertainties in its mathematical model. More specifically, robustness can be subdivided into *stability robustness* and *performance robustness*, depending upon whether we are looking at the robustness of the *stability* of a system (determined by the location of the system's poles), or that of its *performance objectives* (such as peak overshoot, settling time, etc.).

We intuitively felt in Chapter 1 that a closed-loop system is *more* robust than an open-loop system. We are now in a position to mathematically compare the *sensitivities* of open and closed-loop systems, representatives of which are shown in Figure 2.39. For simplicity, it is assumed that for both open and closed loop systems, the controller transfer function is a constant, given by K, while the plant transfer function is $G(s)$.The transfer function of the open-loop control-system, $F_o(s)$, is given by

$$F_o(s) = Y(s)/Y_d(s) = KG(s) \tag{2.154}$$

while that of the closed-loop system ($F_c(s)$) is given by Eq. (2.124)

$$F_c(s) = Y(s)/Y_d(s) = KG(s)/[1 + KG(s)] \tag{2.155}$$

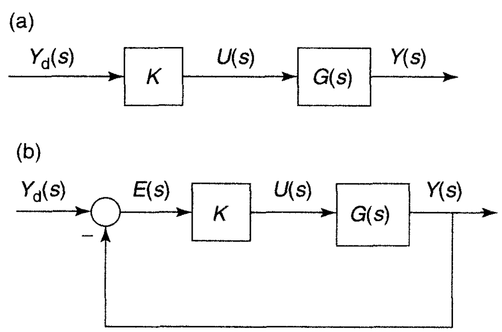

Figure 2.39 Single-input, single-output open-loop (a) and closed-loop (b) control systems with controller transfer function K and plant transfer function $G(s)$

Since the controller transfer function, K, is derived based on some mathematical model of $G(s)$, we can determine each control system's robustness by calculating the *sensitivity* of the overall transfer function, $Y(s)/Y_d(s)$, to variation in K. Mathematically, the sensitivity of either open-loop, or closed-loop system to variation in controller gain, K, can be expressed as

$$S(s) = [Y_d(s)/Y(s)]\partial[Y(s)/Y_d(s)]/\partial K \tag{2.156}$$

where $\partial[Y(s)/Y_d(s)]/\partial K$ denotes the change in the transfer function due to a change in K. Then the sensitivities of the open and closed-loop systems to variation in K are given by

$$S_o(s) = [1/F_o(s)]\partial F_o(s)/\partial K = 1/K \tag{2.157}$$

and

$$S_c(s) = [1/F_c(s)]\partial F_c(s)/\partial K = 1/[K(1 + KG(s))] \tag{2.158}$$

The ratio of the open-loop sensitivity to the closed-loop sensitivity $S_o(s)/S_c(s)$ is thus

$$S_o(s)/S_c(s) = [1 + KG(s)] \tag{2.159}$$

which is nothing else but our well known acquaintance, the *return difference function* (or the closed-loop characteristic polynomial)! The *magnitude* of the return difference, $|1 + KG(s)|$, is *greater than* 1, which confirms that an open-loop system is *more sensitive* (or *less robust*) when compared to the closed-loop system to variations in controller gain. The *greater* the value of the return difference, the *larger* is the robustness of a closed-loop system (Eq. (2.158)). Hence, one can measure a closed-loop system's robustness by determining the return difference function in the frequency domain, $s = i\omega$, for a range of frequencies, ω.

We need not confine ourselves to closed-loop systems with a constant controller transfer function when talking about robustness; let us consider a cascade closed-loop system (Figure 2.32) with a controller transfer function, $H(s)$. Either the Bode plot of $G(i\omega)H(i\omega)$, or the Nyquist plot of $G(s)H(s)$ can be utilized to convey information about a system's *stability robustness* (i.e. *how robust* is the *stability* of the system with respect to *variations* in the system's model). The Nyquist plot is more intuitive for analyzing stability robustness. From Nyquist stability theorem, we know that the closed-loop system's stability is determined from the encirclements by the locus of $G(s)H(s)$ of the point -1 in the $G(s)H(s)$ plane. Therefore, a measure of stability robustness can be *how far away* the locus of $G(s)H(s)$ is to the point -1, which indicates how far the system is from being unstable. The farther away the $G(s)H(s)$ locus is from -1, the greater is its stability robustness, which is also called the *margin of stability*. The *closest distance* of $G(s)H(s)$ from -1 can be defined in the complex $G(s)H(s)$ plane by two quantities, called the *gain margin* and the *phase margin*, which are illustrated in Figure 2.40 depicting a typical Nyquist diagram for a stable closed-loop system. A circle of unit radius is overlaid on the Nyquist plot. The closest distance of the Nyquist plot of $G(s)H(s)$ to -1 is indicated by two points A and B. Point A denotes the intersection of $G(s)H(s)$ with the negative real axis *nearest to the point -1*, while point B denotes the intersection of $G(s)H(s)$ with the *unit circle*. Point A is situated at a distance α from

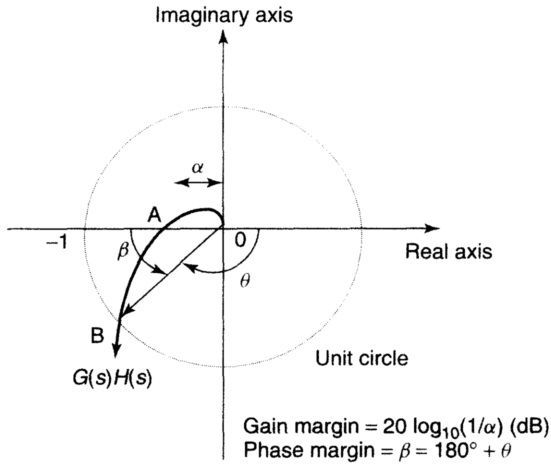

Figure 2.40 The Nyquist plot of $G(s)H(s)$ showing the gain and phase margins

the origin, while point B is located on the unit circle at an *angle θ from the positive real axis*. It is clear that the gain of $G(s)H(s)$ at point B is unity, while its phase at the same point is θ.

The *gain margin* is defined as the *factor* by which the gain of $G(s)H(s)$ can be *increased* before the locus of $G(s)H(s)$ hits the point -1. From Figure 2.40, it is evident that the gain margin is equal to $1/\alpha$, or in dB it is given by

$$\text{Gain Margin in dB} = 20\ \log_{10}(1/\alpha) \tag{2.160}$$

A *negative* gain margin indicates that the system is *unstable*. The *phase margin* is defined as the difference between the phase of $G(s)H(s)$ at point B, θ, and the phase of the *negative real axis*, -180° (on which the point -1 is located). Thus, phase margin is given by the angle β in Figure 2.40:

$$\text{Phase Margin} = \beta = \theta - (-180^\circ) = \theta + 180^\circ \tag{2.161}$$

The gain margin indicates how far away the *gain* of $G(s)H(s)$ is from 1 (i.e. the gain of the point -1) when its phase is -180° (point A). Similarly, the phase margin indicates how far away the *phase* of $G(s)H(s)$ is from -180° (i.e. the phase of the point -1) when the gain of $G(s)H(s)$ is 1 (point B).

Since the Bode plot is a plot of gain and phase in frequency domain (i.e. when $s = i\omega$), we can use the Bode plot of $G(s)H(s)$ to determine the gain and phase margins. In the Bode gain plot, the unit circle of the Nyquist plot (Figure 2.40) translates into the line for or *zero* dB gain (i.e. *unit* magnitude, $|G(i\omega)H(i\omega)|$), while the negative real axis of the Nyquist plot transforms into the line for -180° phase in the Bode phase plot. Therefore, the gain margin is simply the gain of $G(i\omega)H(i\omega)$ when its phase crosses the -180° phase line, and the phase margin is the difference between the phase of $G(i\omega)H(i\omega)$ and -180° when the gain of $G(i\omega)H(i\omega)$ crosses the 0 dB line. The frequency for which the phase of $G(i\omega)H(i\omega)$ crosses the -180° line is called the *phase crossover frequency*, ω_P, while the frequency at which the gain of $G(i\omega)H(i\omega)$ crosses the 0 dB line is called the *gain crossover frequency*, ω_G.

Example 2.22

Consider the closed-loop system of Figure 2.32, with $G(s) = (2s^2 + 5s + 1)/(s^2 + 2s + 3)$ and $H(s) = 1/s^2$. Figure 2.41 shows the Bode plot of $G(i\omega)H(i\omega)$ which is obtained by the following MATLAB CST command:

```
>>num=[2 5 1]; den=conv([1 2 3],[1 0 0]); G=tf(num,den); bode(G) <enter>
```

Figure 2.41 shows that the phase crosses the $-180°$ line at the phase crossover frequency, $\omega_P = 3.606$ rad/s. The gain present at this frequency is -14.32 dB. Hence, the gain margin is 14.32 dB, indicating that the gain of $G(s)H(s)$ can be increased by 14.32 dB before its Nyquist locus hits the point -1. In Figure 2.41, we can see that the 0 dB gain line is crossed at gain crossover frequency, $\omega_G = 1.691$ rad/s, for which the corresponding phase angle is $-148.47°$. Therefore, the phase margin is $-148.47° + 180° = 31.53°$.

The numerical values of gain margin, phase margin, gain crossover frequency, and phase crossover frequency, can be directly obtained using the MATLAB CST command *margin(sys)*, or *margin(mag,phase,w)*, where *mag,phase,w* are the magnitude, phase, and frequency vectors obtained using the command $[mag,phase,w] = bode(sys,w)$. For a system having a frequency response, $G(i\omega)H(i\omega)$, which is changing rapidly with frequency (such as in the present

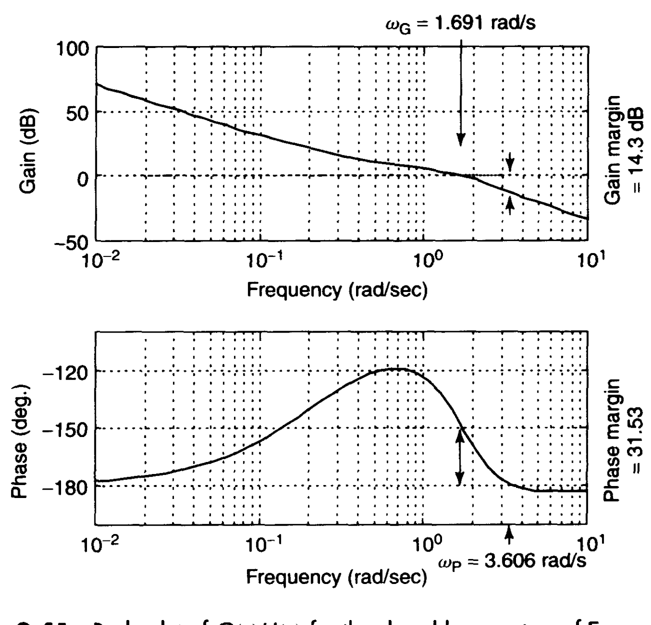

Figure 2.41 Bode plot of $G(s)H(s)$ for the closed-loop system of Example 2.22

example), the command *margin(sys)* may yield inaccurate results. Therefore, it is advisable to use the command *margin(mag,phase,w)*, with the magnitude and phase generated at a large number of frequencies in the range where the frequency response is changing rapidly. This is done by using the command $w = logspace(a, b, n)$, which generates n equally space frequencies between 10^a and 10^b rad/s, and stores them in vector w. Then the command $[mag,phase,w] = bode(sys,w)$ will give the desired magnitude and phase vectors from which the gain and phase margins can be calculated. This procedure is illustrated for the present example by the following commands:

```
>>num=[2 5 1]; den=conv([1 2 3],[1 0 0]); w=logspace(-2,1,1000);
  G=tf(num,den);

>>[mag,phase,w]=bode(G); <enter>

>>margin(mag,phase,w) <enter>
```

A Bode plot results, with computed gain and phase margins and the corresponding crossover frequencies indicated at the top of the figure (Figure 2.42).

Another way of obtaining gain and phase margins is from the *Nichols* plot, which is a plot of the gain of the open-loop frequency response, $G(i\omega)H(i\omega)$, against its phase. The gain and phase margins can be directly read from the resulting plot in which the point -1 of the Nyquist plot corresponds to the point 0 dB, $-180°$.

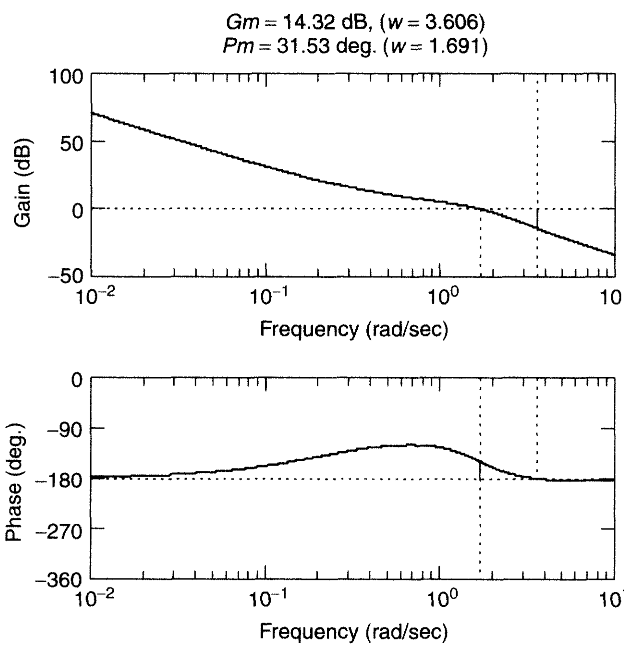

Figure 2.42 Gain and phase margins and crossover frequencies for the system of Example 2.22 obtained using the MATLAB Control System Toolbox command *margin*

In addition to the open-loop gain and phase, the Nichols plot shows *contours* of constant *closed-loop magnitude and phase*, $|Y(i\omega)/Y_d(i\omega)|$, and $\phi(\omega)$, given by

$$\begin{aligned} Y(i\omega)/Y_d(i\omega) &= G(i\omega)H(i\omega)/[1 + G(i\omega)H(i\omega)] \\ &= |Y(i\omega)/Y_d(i\omega)|\, e^{i\phi(\omega)} \end{aligned} \tag{2.162}$$

It can be shown that the closed-loop gain and phase are related to the open-loop gain and phase $|G(i\omega)H(i\omega)|$ and θ, respectively, by the following equations:

$$\begin{aligned} |Y(i\omega)/Y_d(i\omega)| &= 1/[1 + 1/|G(i\omega)H(i\omega|^2 \\ &\quad + 2\cos(\theta)/|G(i\omega)H(i\omega|]^{1/2} \end{aligned} \tag{2.163}$$

$$\phi(\omega) = -\tan^{-1}\{\sin(\theta)/[\cos(\theta) + |G(i\omega)H(i\omega|]\} \tag{2.164}$$

where $G(i\omega)H(i\omega) = |G(i\omega)H(i\omega|e^{i\theta}$. Seeing the formidable nature of Eqs. (2.163) and (2.164), plotting the contours of constant closed-loop gain and phase by hand appears impossible. However, MATLAB again comes to our rescue by providing the Control System Toolbox (CST) command *nichols(sys)* for plotting the open-loop gain and phase, and the command *ngrid* for plotting the contours of the closed-loop gain and phase. Here *sys* denotes the LTI object of the open-loop transfer function, $G(s)H(s)$. Figure 2.43 shows the Nichols plot of the system in Example 2.22 obtained using the following MATLAB commands:

```
>>num=[2 5 1]; den=conv([1 0 0],[1 2 3]); G=tf(num,den); ngrid('new');
  nichols(G) <enter>
```

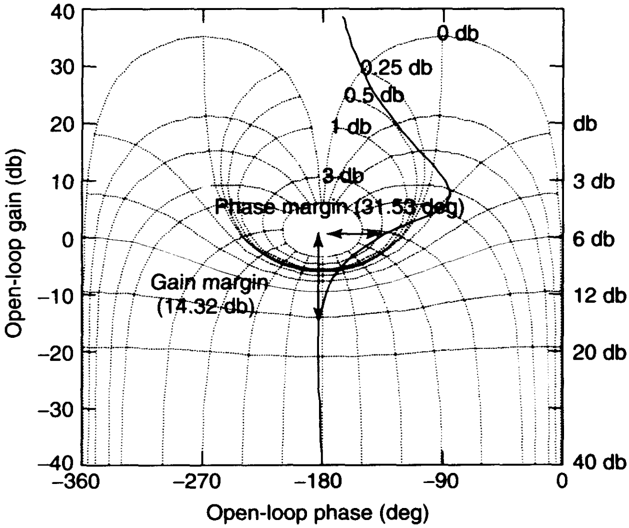

Figure 2.43 Nichols plot for the system of Example 2.22