Handbook of Networked and Embedded Control Systems, Birkhauser, Boston, 2005.

Control of Single-Input Single-Output Systems*

Dimitrios Hristu-Varsakelis[1] and William S. Levine[2]

[1] Department of Applied Informatics,
University of Macedonia, Thessaloniki, 54006, Greece dcv@uom.gr
[2] Electrical and Computer Engineering,
University of Maryland, College Park, MD 20742, U.S.A. wsl@umd.edu

1 Introduction

There is an extensive body of theory and practice devoted to the design of feedback controls for linear time-invariant systems. This chapter contains a brief introduction to the subject with emphasis on the design of digital controllers for continuous-time systems. Before we begin it is important to appreciate the limitations of linearity and of feedback. There are situations where it is best not to use feedback in the control of a system. Typically, this is true for systems that do not undergo much perturbation and for which sensors are either unavailable or too inaccurate. There are also limits to what feedback can accomplish. One of the most important examples is the nonlinearity that is present in virtually all systems due to the saturation of the actuator. Saturation will limit the range of useful feedback gains even when instability does not. It is important to keep this in mind when designing controllers for real systems, which are only linear within a limited range of input amplitudes.

The method used to design a controller depends critically on the information available to the designer. We will describe three distinct situations:

1. The system to be controlled is available for experiment but the designer cannot obtain a mathematical model of the system.
2. The designer has an experimentally determined frequency response of the system but does not have other modeling information.
3. The designer has a mathematical model of the system to be controlled.

The second case arises when the underlying physics of the system is poorly understood or when a reasonable mathematical model would be much too complicated to be useful. For example, a typical feedback amplifier might contain 20 or more energy storage elements. A mathematical model for this amplifier would be at least 20th order.

*This work was supported in part by NSF Grant EIA-008001.

It will be easiest to understand the different design methods if we begin with the third case, where there is an accurate mathematical model of the plant (the system to be controlled). When such a model is available, the feedback control of a single-input single-output (SISO) system begins with the following picture. The plant shown in Fig. 1 typically operates in continuous

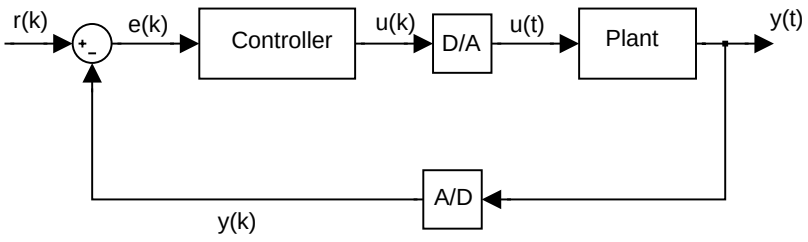

Fig. 1. A sampled-data feedback control system

time. It can be described by its transfer function:

$$Y(s) = G_c(s)U(s),$$

where $U(s)$, $Y(s)$ are the Laplace transforms of the input and output signals respectively, and

$$G_c(s) = \frac{\sum_0^{n-1} b_i s^i}{\sum_0^n a_i s^i} = \frac{\prod_1^{n-1}(s - z_i)}{\prod_1^n (s - p_i)}. \tag{1}$$

The coefficients a_i, b_i are real; the roots z_i, p_i of the numerator and denominator are the zeros and poles (respectively) of the transfer function. We assume that these parameters are given and that they are constant.

Note that (1) limits the class of systems to those that can be adequately approximated by such a transfer function. For a discussion of controller design when $G_c(s)$ includes a pure delay, described by e^{-sT}, see "Control Issues in Systems with Loop Delays" by Mirkin and Palmor in this handbook. The output in Fig. 1 is fed directly back to the summer (comparator). For simplicity and clarity we restrict our discussion to unity feedback systems, as in Fig. 1. It is fairly easy to account for dynamics or filtering associated with the sensor if necessary.

The controller (in cascade with the plant) is to be designed so that the closed-loop system meets a given set of specifications. The controller is assumed to be linear (in a sense to be made precise shortly). Modern controllers are often implemented in a digital computer. This requires the use of analog-to-digital (A/D) and digital-to-analog (D/A) converters in order to interface with the continuous-time plant. This makes the plant, as seen by the

controller, a sampled-data system with input $u(k)$ and output $y(k)$. See the chapter, in this handbook, entitled "Basics of Sampling and Quantization" by Santina and Stubberud for a discussion of the effects of time discretization and D/A and A/D conversion.

2 Description of Sampled-Data Systems

The D/A block shown in Fig. 1 converts the discrete-time signal $u(k)$ produced by the controller to a continuous-time piecewise constant signal via a "zero-order hold" (ZOH). Let $u(k)$ be the discrete-time input signal, arriving at the D/A block at multiples of the sampling period T. In the time domain, the ZOH can be modeled as a sum of shifted unit step functions:[3]

$$u(t) = \sum_{0}^{\infty} u(k)[1(t - kT) - 1(t - (k + 1)T)].$$

The Laplace transform of the last expression yields

$$U(s) = \underbrace{\sum_{0}^{\infty} u(k)e^{-kTs}}_{U(z)} \left(\frac{1}{s} - \frac{e^{-Ts}}{s}\right).$$

If we think of $u(k)$ as a continuous-time impulse train, $u(k)\delta(t - kT)$, then the ZOH has a transfer function

$$G_{ZOH}(s) = \frac{1}{s}(1 - e^{-sT}).$$

From the point of view of the (discrete-time) controller, the transfer function of the sampled-data system is given by the z-transform of the ZOH/plant system

$$G(s) = \frac{1 - e^{-sT}}{s} G_c(s),$$

which is (by $z = e^{sT}$)

$$G(z) = \frac{z - 1}{z} \mathcal{Z}\{G_c(s)/s\},$$

where $\mathcal{Z}\{G_c(s)/s\}$ is computed by first calculating the inverse Laplace transform of $G_c(s)/s$ to obtain a continuous-time signal, $\hat{g}(t)$, then sampling this signal, and finally computing the Z-transform of this discrete-time signal.

If we let $C(z)$ denote the transfer function of the controller, then the closed-loop transfer function is

[3]The unit step function $1(t)$ equals zero for $t < 0$, one for $t \geq 0$.

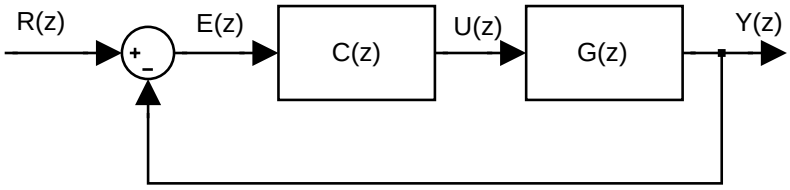

Fig. 2. A sampled-data feedback control system

$$\frac{Y(z)}{U(z)} = G_{cl}(z) = \frac{G(z)C(z)}{1 + G(z)C(z)}.$$

This is illustrated in Fig. 2.

An important point to remember about sampled-data systems is that the real system evolves in continuous time, including the time between the sampling instants. This inter-sample behavior must be accounted for in most applications.

3 Control Specifications

The desired performance of the closed-loop system in Fig. 2 is usually described by means of a collection of specifications. They can be organized into four groups:

- Stability
- Steady-state error
- Transient response
- Robustness

These will be discussed in order below.

3.1 Stability

A system is *bounded-input bounded-output (BIBO) stable* if any bounded input results in a bounded output. A system is *internally stable* if its state decays to zero when the input is identically zero. If we limit ourselves to linear time-invariant (LTI) systems, then all questions of stability can be settled easily by examining the poles of the closed-loop system. In particular, the closed-loop system is both BIBO and internally stable if and only if all of its poles[4] are inside the unit circle. Mathematically, if the poles of the closed-loop system are denoted by p_i, $i = 1, 2, ..., n$ then the system is BIBO and internally stable if $|p_i| < 1$ for all i.

[4]This must include any poles that are cancelled by zeros.

3.2 Steady-state error

In many situations the main objective of the closed-loop system is to track a desired input signal closely. For example, a paper-making or metal-rolling machine is expected to produce paper or metal of a specified thickness. Brief, transient errors when the process starts, while undesirable, can often be ignored. On the other hand, persistent tracking errors are a serious problem. Typically, the specification will be that the steady-state error in response to a unit step input must be exactly zero. It is surprisingly easy to meet this requirement in most cases.

The difference between input and output is $e(k)$, or in the z-domain,

$$E(z) = \frac{R(z)}{1 + G(z)C(z)}.$$

We can examine the steady-state error by using the "final value theorem"

$$e(\infty) \stackrel{\triangle}{=} \lim_{k\to\infty} e(k) = \lim_{z\to 1}(1 - z^{-1})E(z).$$

If the input is a unit step ($U_s(z) = z/(z-1)$), then the last equation yields

$$e(\infty) = \lim_{z\to 1} \frac{1}{1 + G(z)C(z)}. \tag{2}$$

Equation (2) indicates that the steady-state error will be zero provided that

$$\lim_{z\to 1} G(z)C(z) = \infty,$$

which will be true if $G(z)C(z)$ has one or more poles at $z = 1$.

More elaborate steady-state specifications exist, but the details can easily be derived using this example as a model or by consulting the books by Dorf and Bishop [5] or Franklin et al. [6].

3.3 Transient response

The transient response of the closed-loop system is important in many applications. A good example is the stability and control augmentation systems (SCASs) now common in piloted aircraft and some automobiles. These are systems that form an inner (usually multi-input multi-output (MIMO)) control loop that improves the handling qualities of the vehicle. The pilot or driver is the key component in an outer control loop that provides command inputs to the SCAS. The transient characteristics of the vehicle are crucial to the pilot's and driver's handling of the vehicle and to the passenger's perception of the ride. If you doubt this, imagine riding in or driving a car with a large rise time or large percent overshoot (defined below).

The transient response of an LTI system depends on the input as well as on the initial conditions. The standard specifications assume a unit step as the test input, and the system starts from rest, with zero initial conditions. The resulting step response is then characterized by several of its properties, most notably its rise time, settling time, and percent overshoot. These are displayed in Fig. 3 and defined below.

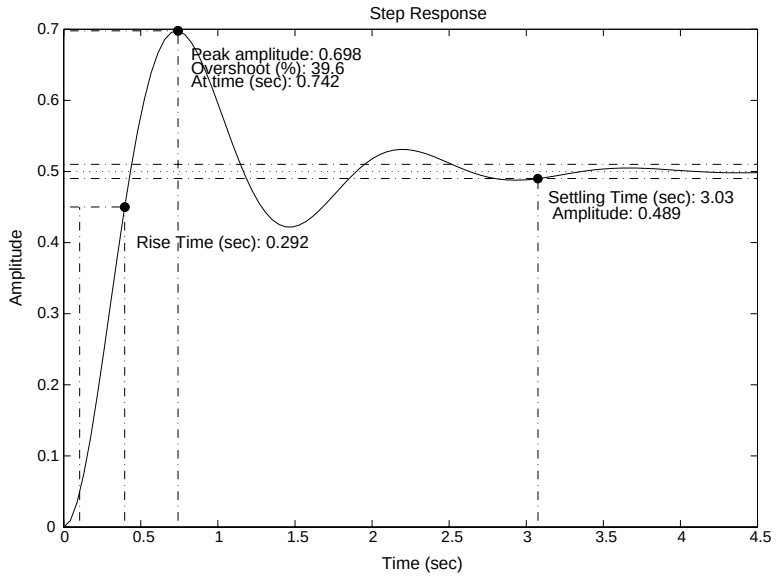

Fig. 3. The step response of an LTI system and its properties

- Rise time: Usually defined to be the time required for the step response to go from 10% of its final value to 90% of its final value.
- Settling time: Usually defined to be the time at which the step response last crosses the lines at $\pm 2\%$ of its final value.
- Percent overshoot: Usually defined to be the ratio (peak amplitude minus final value)/(final value) expressed as a percentage.

In each case there are variant definitions. For example, sometimes $\pm 1\%$ or $\pm 5\%$ is used instead of $\pm 2\%$ in the definition of settling time. The final value is the steady-state value of the step response, 0.5 in Fig. 3.

3.4 Robustness

Because a system either is or is not stable, a nominally stable system may become unstable as a result of arbitrarily small differences between the nominal plant $G(z)$ used for design and the real plant. Such differences might be

due to inaccuracies in parameter values, variations in operating conditions, or the deliberate omission of aspects of the nominal plant. For example, the flexure modes of the body and wings of an aircraft are usually omitted from the nominal plant model used for controller design. This underscores the importance of knowing how "close" to instability the closed-loop system is. The "distance to instability" is commonly quantified for SISO LTI systems in two ways. One is the *gain margin*, namely the gain factor K that must be applied to the forward path (replacing $G(z)C(z)$ by $KG(z)C(z)$ in Fig. 2) in order for the system to become unstable. The other, known as the *phase margin*, is the maximum amount of delay (or phase shift) $e^{-j\phi_M}$ that can be introduced in the forward path before the onset of instability.

Robustness, as a specification and property of a controlled system, has received much attention in the research literature in recent years. This has led to robustness tests for MIMO systems as well as a variety of tools for designing robust control systems. See [8, 15] for more details.

4 Analysis and Design Tools

4.1 The root locus

Consider making the controller in Fig. 2 simply a gain, i.e., $C(z) = K$.

As K varies from 0 to ∞, the poles of $G_{cl}(z) = \frac{KG(z)}{1+KG(z)}$ trace a set of curves (called the "root locus") in the complex plane. When $K = 0$ the poles of the "closed-loop system" are identical to the poles of the open-loop system, $G(z)$. Thus, each locus starts at one of the poles of $G(z)$. As $K \to \infty$ it is possible to prove that the closed-loop poles go to the open-loop zeros, including both the finite and infinite zeros, of $G(z)$. Given a specific value for K, it is easy to compute the resulting closed-loop pole locations. Today, one can easily compute the entire root locus; for example, the MATLAB command `rlocus` was used to produce Fig. 4. The root locus plot is obviously useful to the designer who plans on using a controller $C(z) = K$. He or she simply chooses a desirable set of pole locations, consistent with the loci, and determines the corresponding value of K. MATLAB has a command, `rlocfind`, that facilitates this. Alternatively, one can use the sisotool graphical user interface (GUI) in MATLAB to perform the same task. The choice of pole location is aided by the use of a grid that displays contours of constant natural frequency and damping ratio. We will have more to say regarding the choice of pole locations and the use of the root locus plot in Section 5.1.

By combining the controller and the plant and multiplying by K (the effective plant is then $C(z)G(z)$), the root locus can be used to determine the gain margin. As will be explained later, the effect of various compensators can also be analyzed and understood by appropriate use of the root locus. Lastly, the idea of the root locus, the graphical display of the pole locations as an implicit function of a single variable in the design, can be very useful in

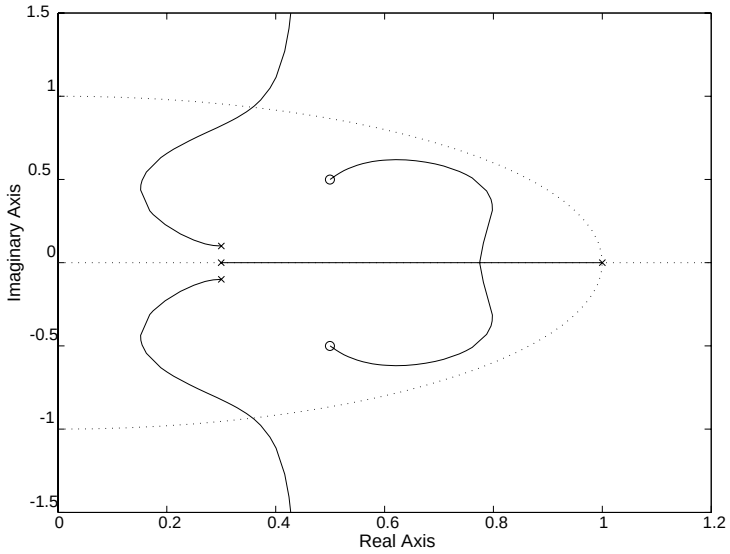

Fig. 4. The root locus when $G(z) = \frac{z^2-z+0.5}{z^4-1.9z^3+1.18z^2-0.31z+0.03}$

a variety of applications. Modern computers make it fairly easy to generate such root loci.

4.2 The Bode, Nyquist, and Nichols plots

There are at least two situations where it is preferable to use the frequency response of the plant rather than its transfer function $G(z)$ for control system design. First, when the plant is either stable or easily stabilized, it is often possible to determine $|G(e^{j\Omega T})|$ and $\angle G(e^{j\Omega T})$, where T is the time interval between samples, experimentally for a range of values of Ω. This data is sufficient for control design, completely eliminating the need for an analytical expression for $G(z)$. Second, a system with many poles and zeros can produce a very complicated and confusing root locus. The frequency response plots of such a system can make it easier for the designer to focus on the essentials of the design. This second situation is exemplified by feedback amplifier design, where a state space or transfer function model would be of high order, but the frequency response is relatively simple.

The Nyquist plot of the imaginary part of $G(e^{j\Omega T})$ versus the real part of $G(e^{j\Omega T})$ provides a definitive test for stability of the closed-loop system. It also gives the exact gain and phase margins unambiguously. However, it is not particularly easy to use for design. In contrast, both the Bode plots and Nichols chart are very useful for design but can be ambiguous with regard to stability. There are two Bode plots. The Bode magnitude plot presents

$20\log|G(e^{j\Omega T})|$ on the vertical axis versus $\log\Omega$ on the horizontal axis. The Bode phase plot shows $\angle G(e^{j\Omega T})$ on the vertical axis and uses the same horizontal axis as the magnitude plot. The Nichols chart displays $20\log|G(e^{j\Omega T})|$ on the vertical axis versus $\angle G(e^{j\Omega T})$ on the horizontal axis. An example of both plots is shown in Fig. 5. Note that the lightly dotted curves on the Nichols chart are contours of constant gain (in decibels) and phase (in degrees) of the closed-loop system. Thus, any point on the Nichols plot for $G(z)$ also identifies a value of $20\log|\frac{G(z)}{(1+G(z))}|$ and of $\angle\frac{G(z)}{(1+G(z))}$ for some value of Ω.

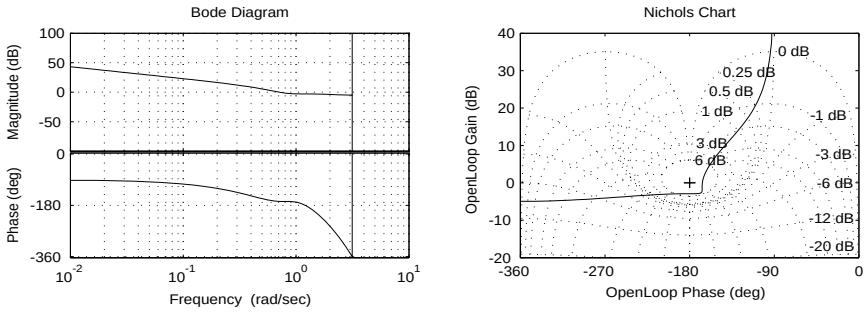

Fig. 5. The Bode plots and Nichols chart for $G(z) = \frac{z^2-z+0.5}{z^4-1.9z^3+1.18z^2-0.31z+0.03}$

The use of logarithmic scaling for the magnitude offers an important convenience: The effect of a series compensator $C(z)$ on the logarithmic magnitude is additive, as is its effect on the phase.

5 Classical Design of Control Systems

In reality, the design of a control system usually includes the choosing of sensors, actuators, computer hardware and software, A/D and D/A converters, buffers, and, possibly, other components of the system. In a modern digital controller the code implementing the controller must also be written. In addition, most control systems include a considerable amount of protection against emergencies, overloads, and other exceptional circumstances. Lastly, it is now common to include some collection and storage of maintenance information as well. Although control theory often provides useful guidance to the designer in all of the above-mentioned aspects of the design, it only provides explicit answers for the choice of $C(z)$ in Fig. 2. It is this aspect of control design that is covered here.

5.1 Analytical model-based design

The theory of control design often begins with an explicitly known plant $G(z)$ and a set of specifications for the closed-loop system. The designer is

expected to find a controller $C(z)$ such that the closed-loop system satisfies those specifications. In this case, a natural beginning is to plot the root locus for $G(z)$. If the root locus indicates that the specifications can be met by a controller $C(z) = K$, then the theoretical design is done. However, it is not a trivial matter to determine from the root locus if there is a value of K for which the specifications are met. Notice that the example specifications in Section 3 include both time domain and frequency domain requirements.

The designer typically needs to be able to visualize the closed-loop step response from knowledge of the closed-loop pole and zero locations only. This is easily done for second-order systems where there is a tight linkage between the pole locations and transient response. Many SISO controlled systems can be adequately approximated by a second-order system even though the actual system is of higher order. For example, there are many systems in which an electric motor controls an inertia. The mechanical time constants in such a system are often several orders of magnitude slower than the electrical ones and dominate the behavior. The electrical transients can be largely ignored in the controller design.

A second-order system can be put in a standard form that only depends on two parameters, the damping ratio ζ and the natural frequency ω_n. The continuous-time version is

$$G_{cl}(s) = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2}, \tag{3}$$

where $G_{cl}(s)$ denotes the closed-loop transfer function. Notice that the poles of $G_{cl}(s)$ are located at $s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = \omega_n e^{j\pi\pm\cos^{-1}\zeta}$. For stable systems with a pair of complex conjugate poles, $0 \leq \zeta < 1$. The description (3) is not used for systems with real poles. The system (3) has step response

$$y(t) = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\left(\sin(\sqrt{1-\zeta^2}\omega_n t + \tan^{-1}(\frac{\sqrt{1-\zeta^2}}{\zeta}))\right). \tag{4}$$

The constants ζ and ω_n completely determine the step response. With a little experience a designer can then picture the approximate step response in his or her mind while looking at the pole locations. For a system with additional poles and zeros the actual step response can be quite different from that in (4), but designers need insight and a way to start. An initial design that is very far from meeting the specifications can often be modified and adjusted into a good design after several iterations.

It is possible to create a second-order discrete-time system whose step response exactly matches that of (4). The first step is to choose a time interval between outputs of the discrete-time system, say T_s. Then, if the continuous-time system has a pole at p_i, the corresponding discrete-time system must have a corresponding pole at $p_{id} = e^{p_i T_s}$. The poles of the continuous-time system (3) are at $p_i = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$. Thus, the poles of the discrete-time system are at $p_{id} = e^{-\zeta\omega_n T_s} e^{\pm j\omega_n\sqrt{1-\zeta^2}T_s}$. Writing the p_{id} in polar form

as $R \cdot e^{j\theta}$ (the subscripts have been dropped because there is only one value) gives

$$R = e^{-\zeta\omega_n T_s} \tag{5}$$

$$\theta = \pm\omega_n\sqrt{1-\zeta^2}T_s. \tag{6}$$

Solving explicitly for ζ and ω_n gives

$$\zeta = \pm\frac{\ln(R)}{\sqrt{\theta^2 + (\ln(R))^2}} \tag{7}$$

$$\omega_n = \pm\sqrt{\theta^2 + (\ln(R))^2}. \tag{8}$$

This defines two curves in the z-plane, a curve of constant ζ and a curve of constant ω_n. These curves can be plotted on the root locus plot—the MATLAB command is `zgrid`. For a second-order system in the standard form (3), both the transient response characteristics and the phase margin are directly related to ζ and ω_n:

$$rise\ time = t_r \approx \frac{1.8}{\omega_n} \tag{9}$$

$$settling\ time = t_s \approx \frac{4.6}{\zeta\omega_n} \tag{10}$$

$$percent\ overshoot = P.O. = 100\frac{e^{-\pi\zeta/\sqrt{1-\zeta^2}}}{final\ value}. \tag{11}$$

The $final\ value$ is the constant steady-state value reached after the transients have died out ($final\ value = \lim_{k\to\infty} y(k)$).

Clearly, if a designer can satisfy the specifications using only $C(z) = K$, the best value of K can be chosen by plotting the root locus of $G(z)$ and looking at where the loci intersect the contours of constant ζ and ω_n. If this is not sufficient, there are several standard components one can try to include in $C(z)$ in order to alter the root locus so that its branches pass through the desired values of ζ and ω_n. The best known of these are the lead and lag compensators defined here for discrete-time systems.

Lead compensator:

$$C_{le}(z) = \frac{(\frac{z}{z_l} - 1)}{(\frac{z}{p_l} - 1)}, \quad 0 \le p_l < z_l \le 1 \tag{12}$$

Lag compensator:

$$C_{la}(z) = \frac{(\frac{z}{z_l} - 1)}{(\frac{z}{p_l} - 1)}, \quad 0 \le z_l < p_l \le 1. \tag{13}$$

It will be easiest to understand the different design methods if we begin with the third case, where there is an accurate mathematical model of the plant (the system to be controlled). When such a model is available, the feedback control of a single-input single-output (SISO) system begins with the following picture. The plant shown in Fig. 1 typically operates in continuous

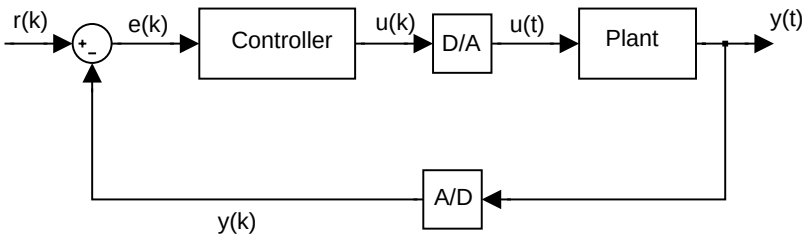

Fig. 1. A sampled-data feedback control system

time. It can be described by its transfer function:

$$Y(s) = G_c(s)U(s),$$

where $U(s)$, $Y(s)$ are the Laplace transforms of the input and output signals respectively, and

$$G_c(s) = \frac{\sum_0^{n-1} b_i s^i}{\sum_0^n a_i s^i} = \frac{\prod_1^{n-1}(s - z_i)}{\prod_1^n (s - p_i)}. \tag{1}$$

The coefficients a_i, b_i are real; the roots z_i, p_i of the numerator and denominator are the zeros and poles (respectively) of the transfer function. We assume that these parameters are given and that they are constant.

Note that (1) limits the class of systems to those that can be adequately approximated by such a transfer function. For a discussion of controller design when $G_c(s)$ includes a pure delay, described by e^{-sT}, see "Control Issues in Systems with Loop Delays" by Mirkin and Palmor in this handbook. The output in Fig. 1 is fed directly back to the summer (comparator). For simplicity and clarity we restrict our discussion to unity feedback systems, as in Fig. 1. It is fairly easy to account for dynamics or filtering associated with the sensor if necessary.

The controller (in cascade with the plant) is to be designed so that the closed-loop system meets a given set of specifications. The controller is assumed to be linear (in a sense to be made precise shortly). Modern controllers are often implemented in a digital computer. This requires the use of analog-to-digital (A/D) and digital-to-analog (D/A) converters in order to interface with the continuous-time plant. This makes the plant, as seen by the

controller, a sampled-data system with input $u(k)$ and output $y(k)$. See the chapter, in this handbook, entitled "Basics of Sampling and Quantization" by Santina and Stubberud for a discussion of the effects of time discretization and D/A and A/D conversion.

2 Description of Sampled-Data Systems

The D/A block shown in Fig. 1 converts the discrete-time signal $u(k)$ produced by the controller to a continuous-time piecewise constant signal via a "zero-order hold" (ZOH). Let $u(k)$ be the discrete-time input signal, arriving at the D/A block at multiples of the sampling period T. In the time domain, the ZOH can be modeled as a sum of shifted unit step functions:[3]

$$u(t) = \sum_{0}^{\infty} u(k)[1(t - kT) - 1(t - (k + 1)T)].$$

The Laplace transform of the last expression yields

$$U(s) = \underbrace{\sum_{0}^{\infty} u(k)e^{-kTs}}_{U(z)} \left(\frac{1}{s} - \frac{e^{-Ts}}{s}\right).$$

If we think of $u(k)$ as a continuous-time impulse train, $u(k)\delta(t - kT)$, then the ZOH has a transfer function

$$G_{ZOH}(s) = \frac{1}{s}(1 - e^{-sT}).$$

From the point of view of the (discrete-time) controller, the transfer function of the sampled-data system is given by the z-transform of the ZOH/plant system

$$G(s) = \frac{1 - e^{-sT}}{s} G_c(s),$$

which is (by $z = e^{sT}$)

$$G(z) = \frac{z - 1}{z} \mathcal{Z}\{G_c(s)/s\},$$

where $\mathcal{Z}\{G_c(s)/s\}$ is computed by first calculating the inverse Laplace transform of $G_c(s)/s$ to obtain a continuous-time signal, $\hat{g}(t)$, then sampling this signal, and finally computing the Z-transform of this discrete-time signal.

If we let $C(z)$ denote the transfer function of the controller, then the closed-loop transfer function is

[3]The unit step function $1(t)$ equals zero for $t < 0$, one for $t \geq 0$.

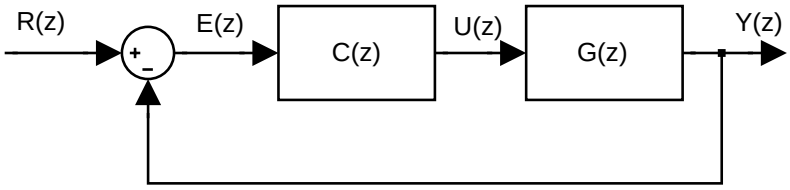

Fig. 2. A sampled-data feedback control system

$$\frac{Y(z)}{U(z)} = G_{cl}(z) = \frac{G(z)C(z)}{1 + G(z)C(z)}.$$

This is illustrated in Fig. 2.

An important point to remember about sampled-data systems is that the real system evolves in continuous time, including the time between the sampling instants. This inter-sample behavior must be accounted for in most applications.

3 Control Specifications

The desired performance of the closed-loop system in Fig. 2 is usually described by means of a collection of specifications. They can be organized into four groups:

- Stability
- Steady-state error
- Transient response
- Robustness

These will be discussed in order below.

3.1 Stability

A system is *bounded-input bounded-output (BIBO) stable* if any bounded input results in a bounded output. A system is *internally stable* if its state decays to zero when the input is identically zero. If we limit ourselves to linear time-invariant (LTI) systems, then all questions of stability can be settled easily by examining the poles of the closed-loop system. In particular, the closed-loop system is both BIBO and internally stable if and only if all of its poles[4] are inside the unit circle. Mathematically, if the poles of the closed-loop system are denoted by p_i, $i = 1, 2, ..., n$ then the system is BIBO and internally stable if $|p_i| < 1$ for all i.

[4]This must include any poles that are cancelled by zeros.

3.2 Steady-state error

In many situations the main objective of the closed-loop system is to track a desired input signal closely. For example, a paper-making or metal-rolling machine is expected to produce paper or metal of a specified thickness. Brief, transient errors when the process starts, while undesirable, can often be ignored. On the other hand, persistent tracking errors are a serious problem. Typically, the specification will be that the steady-state error in response to a unit step input must be exactly zero. It is surprisingly easy to meet this requirement in most cases.

The difference between input and output is $e(k)$, or in the z-domain,

$$E(z) = \frac{R(z)}{1 + G(z)C(z)}.$$

We can examine the steady-state error by using the "final value theorem"

$$e(\infty) \stackrel{\triangle}{=} \lim_{k\to\infty} e(k) = \lim_{z\to 1}(1 - z^{-1})E(z).$$

If the input is a unit step ($U_s(z) = z/(z-1)$), then the last equation yields

$$e(\infty) = \lim_{z\to 1} \frac{1}{1 + G(z)C(z)}. \tag{2}$$

Equation (2) indicates that the steady-state error will be zero provided that

$$\lim_{z\to 1} G(z)C(z) = \infty,$$

which will be true if $G(z)C(z)$ has one or more poles at $z = 1$.

More elaborate steady-state specifications exist, but the details can easily be derived using this example as a model or by consulting the books by Dorf and Bishop [5] or Franklin et al. [6].

3.3 Transient response

The transient response of the closed-loop system is important in many applications. A good example is the stability and control augmentation systems (SCASs) now common in piloted aircraft and some automobiles. These are systems that form an inner (usually multi-input multi-output (MIMO)) control loop that improves the handling qualities of the vehicle. The pilot or driver is the key component in an outer control loop that provides command inputs to the SCAS. The transient characteristics of the vehicle are crucial to the pilot's and driver's handling of the vehicle and to the passenger's perception of the ride. If you doubt this, imagine riding in or driving a car with a large rise time or large percent overshoot (defined below).

The transient response of an LTI system depends on the input as well as on the initial conditions. The standard specifications assume a unit step as the test input, and the system starts from rest, with zero initial conditions. The resulting step response is then characterized by several of its properties, most notably its rise time, settling time, and percent overshoot. These are displayed in Fig. 3 and defined below.

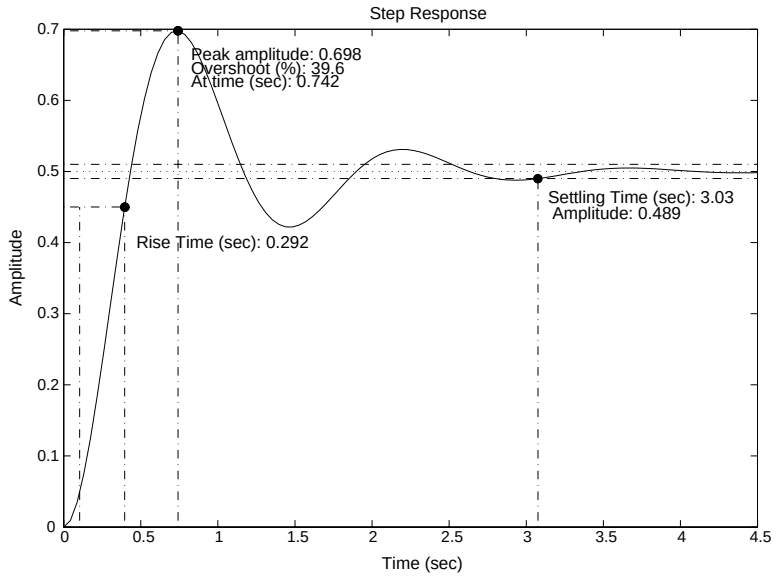

Fig. 3. The step response of an LTI system and its properties

- Rise time: Usually defined to be the time required for the step response to go from 10% of its final value to 90% of its final value.
- Settling time: Usually defined to be the time at which the step response last crosses the lines at $\pm 2\%$ of its final value.
- Percent overshoot: Usually defined to be the ratio (peak amplitude minus final value)/(final value) expressed as a percentage.

In each case there are variant definitions. For example, sometimes $\pm 1\%$ or $\pm 5\%$ is used instead of $\pm 2\%$ in the definition of settling time. The final value is the steady-state value of the step response, 0.5 in Fig. 3.

3.4 Robustness

Because a system either is or is not stable, a nominally stable system may become unstable as a result of arbitrarily small differences between the nominal plant $G(z)$ used for design and the real plant. Such differences might be

due to inaccuracies in parameter values, variations in operating conditions, or the deliberate omission of aspects of the nominal plant. For example, the flexure modes of the body and wings of an aircraft are usually omitted from the nominal plant model used for controller design. This underscores the importance of knowing how "close" to instability the closed-loop system is. The "distance to instability" is commonly quantified for SISO LTI systems in two ways. One is the *gain margin*, namely the gain factor K that must be applied to the forward path (replacing $G(z)C(z)$ by $KG(z)C(z)$ in Fig. 2) in order for the system to become unstable. The other, known as the *phase margin*, is the maximum amount of delay (or phase shift) $e^{-j\phi_M}$ that can be introduced in the forward path before the onset of instability.

Robustness, as a specification and property of a controlled system, has received much attention in the research literature in recent years. This has led to robustness tests for MIMO systems as well as a variety of tools for designing robust control systems. See [8, 15] for more details.

4 Analysis and Design Tools

4.1 The root locus

Consider making the controller in Fig. 2 simply a gain, i.e., $C(z) = K$.

As K varies from 0 to ∞, the poles of $G_{cl}(z) = \frac{KG(z)}{1+KG(z)}$ trace a set of curves (called the "root locus") in the complex plane. When $K = 0$ the poles of the "closed-loop system" are identical to the poles of the open-loop system, $G(z)$. Thus, each locus starts at one of the poles of $G(z)$. As $K \to \infty$ it is possible to prove that the closed-loop poles go to the open-loop zeros, including both the finite and infinite zeros, of $G(z)$. Given a specific value for K, it is easy to compute the resulting closed-loop pole locations. Today, one can easily compute the entire root locus; for example, the MATLAB command `rlocus` was used to produce Fig. 4. The root locus plot is obviously useful to the designer who plans on using a controller $C(z) = K$. He or she simply chooses a desirable set of pole locations, consistent with the loci, and determines the corresponding value of K. MATLAB has a command, `rlocfind`, that facilitates this. Alternatively, one can use the sisotool graphical user interface (GUI) in MATLAB to perform the same task. The choice of pole location is aided by the use of a grid that displays contours of constant natural frequency and damping ratio. We will have more to say regarding the choice of pole locations and the use of the root locus plot in Section 5.1.

By combining the controller and the plant and multiplying by K (the effective plant is then $C(z)G(z)$), the root locus can be used to determine the gain margin. As will be explained later, the effect of various compensators can also be analyzed and understood by appropriate use of the root locus. Lastly, the idea of the root locus, the graphical display of the pole locations as an implicit function of a single variable in the design, can be very useful in

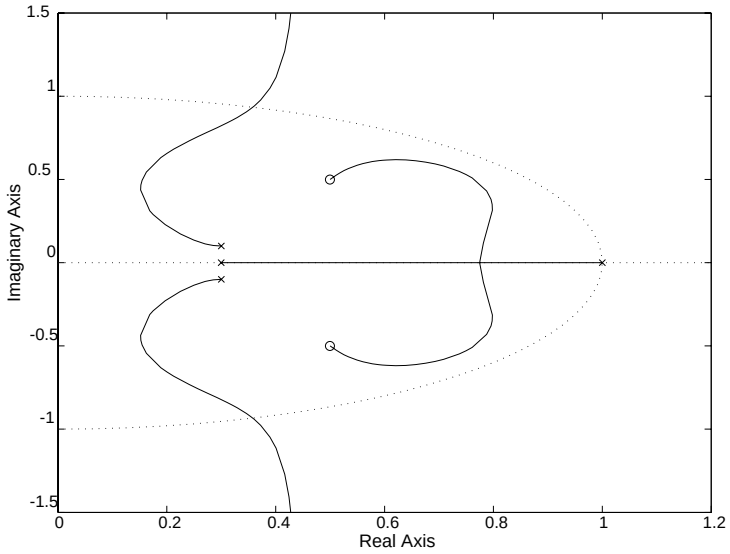

Fig. 4. The root locus when $G(z) = \frac{z^2-z+0.5}{z^4-1.9z^3+1.18z^2-0.31z+0.03}$

a variety of applications. Modern computers make it fairly easy to generate such root loci.

4.2 The Bode, Nyquist, and Nichols plots

There are at least two situations where it is preferable to use the frequency response of the plant rather than its transfer function $G(z)$ for control system design. First, when the plant is either stable or easily stabilized, it is often possible to determine $|G(e^{j\Omega T})|$ and $\angle G(e^{j\Omega T})$, where T is the time interval between samples, experimentally for a range of values of Ω. This data is sufficient for control design, completely eliminating the need for an analytical expression for $G(z)$. Second, a system with many poles and zeros can produce a very complicated and confusing root locus. The frequency response plots of such a system can make it easier for the designer to focus on the essentials of the design. This second situation is exemplified by feedback amplifier design, where a state space or transfer function model would be of high order, but the frequency response is relatively simple.

The Nyquist plot of the imaginary part of $G(e^{j\Omega T})$ versus the real part of $G(e^{j\Omega T})$ provides a definitive test for stability of the closed-loop system. It also gives the exact gain and phase margins unambiguously. However, it is not particularly easy to use for design. In contrast, both the Bode plots and Nichols chart are very useful for design but can be ambiguous with regard to stability. There are two Bode plots. The Bode magnitude plot presents

$20\log|G(e^{j\Omega T})|$ on the vertical axis versus $\log\Omega$ on the horizontal axis. The Bode phase plot shows $\angle G(e^{j\Omega T})$ on the vertical axis and uses the same horizontal axis as the magnitude plot. The Nichols chart displays $20\log|G(e^{j\Omega T})|$ on the vertical axis versus $\angle G(e^{j\Omega T})$ on the horizontal axis. An example of both plots is shown in Fig. 5. Note that the lightly dotted curves on the Nichols chart are contours of constant gain (in decibels) and phase (in degrees) of the closed-loop system. Thus, any point on the Nichols plot for $G(z)$ also identifies a value of $20\log|\frac{G(z)}{(1+G(z))}|$ and of $\angle\frac{G(z)}{(1+G(z))}$ for some value of Ω.

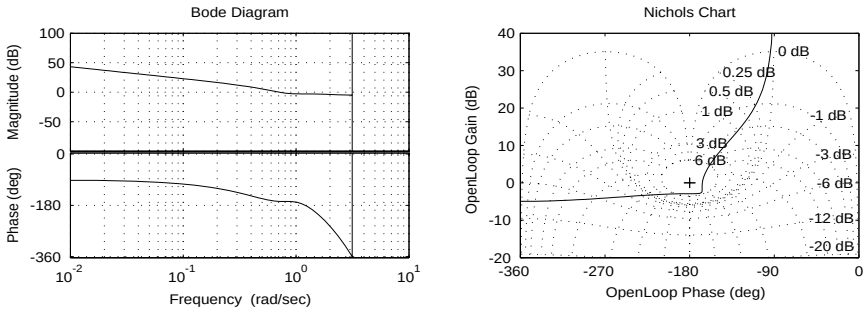

Fig. 5. The Bode plots and Nichols chart for $G(z) = \frac{z^2-z+0.5}{z^4-1.9z^3+1.18z^2-0.31z+0.03}$

The use of logarithmic scaling for the magnitude offers an important convenience: The effect of a series compensator $C(z)$ on the logarithmic magnitude is additive, as is its effect on the phase.

5 Classical Design of Control Systems

In reality, the design of a control system usually includes the choosing of sensors, actuators, computer hardware and software, A/D and D/A converters, buffers, and, possibly, other components of the system. In a modern digital controller the code implementing the controller must also be written. In addition, most control systems include a considerable amount of protection against emergencies, overloads, and other exceptional circumstances. Lastly, it is now common to include some collection and storage of maintenance information as well. Although control theory often provides useful guidance to the designer in all of the above-mentioned aspects of the design, it only provides explicit answers for the choice of $C(z)$ in Fig. 2. It is this aspect of control design that is covered here.

5.1 Analytical model-based design

The theory of control design often begins with an explicitly known plant $G(z)$ and a set of specifications for the closed-loop system. The designer is

expected to find a controller $C(z)$ such that the closed-loop system satisfies those specifications. In this case, a natural beginning is to plot the root locus for $G(z)$. If the root locus indicates that the specifications can be met by a controller $C(z) = K$, then the theoretical design is done. However, it is not a trivial matter to determine from the root locus if there is a value of K for which the specifications are met. Notice that the example specifications in Section 3 include both time domain and frequency domain requirements.

The designer typically needs to be able to visualize the closed-loop step response from knowledge of the closed-loop pole and zero locations only. This is easily done for second-order systems where there is a tight linkage between the pole locations and transient response. Many SISO controlled systems can be adequately approximated by a second-order system even though the actual system is of higher order. For example, there are many systems in which an electric motor controls an inertia. The mechanical time constants in such a system are often several orders of magnitude slower than the electrical ones and dominate the behavior. The electrical transients can be largely ignored in the controller design.

A second-order system can be put in a standard form that only depends on two parameters, the damping ratio ζ and the natural frequency ω_n. The continuous-time version is

$$G_{cl}(s) = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2}, \tag{3}$$

where $G_{cl}(s)$ denotes the closed-loop transfer function. Notice that the poles of $G_{cl}(s)$ are located at $s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = \omega_n e^{j\pi\pm\cos^{-1}\zeta}$. For stable systems with a pair of complex conjugate poles, $0 \leq \zeta < 1$. The description (3) is not used for systems with real poles. The system (3) has step response

$$y(t) = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\left(\sin(\sqrt{1-\zeta^2}\omega_n t + \tan^{-1}(\frac{\sqrt{1-\zeta^2}}{\zeta}))\right). \tag{4}$$

The constants ζ and ω_n completely determine the step response. With a little experience a designer can then picture the approximate step response in his or her mind while looking at the pole locations. For a system with additional poles and zeros the actual step response can be quite different from that in (4), but designers need insight and a way to start. An initial design that is very far from meeting the specifications can often be modified and adjusted into a good design after several iterations.

It is possible to create a second-order discrete-time system whose step response exactly matches that of (4). The first step is to choose a time interval between outputs of the discrete-time system, say T_s. Then, if the continuous-time system has a pole at p_i, the corresponding discrete-time system must have a corresponding pole at $p_{id} = e^{p_i T_s}$. The poles of the continuous-time system (3) are at $p_i = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$. Thus, the poles of the discrete-time system are at $p_{id} = e^{-\zeta\omega_n T_s} e^{\pm j\omega_n\sqrt{1-\zeta^2}T_s}$. Writing the p_{id} in polar form

Notice that the lead compensator has its zero to the right of its pole and the lag compensator has its zero to the left of its pole.

The principle behind both compensators is the same. Consider the real singularities (poles and zeros) of the open-loop system. Suppose that the rightmost real singularity is a pole. This open-loop real pole will move towards a real open-loop zero placed to its left when the loop is closed with a positive gain K. If the open-loop system has a pole near $z = 1$, it is usually possible to speed up the closed-loop transient response by adding a zero to its left. For several reasons (the most important will be explained in Section 6 on limitations of control) one should never add just a zero. Thus, one must add a real pole to the left of the added zero, thereby creating a lead compensator. This lead compensator will generally improve the transient response. The best value of the gain K can be determined using the root locus plot of the combined plant and lead compensator.

The lag compensator is used to reduce the steady-state error. This is done by adding a real pole near the point $z = +1$. Adding only a pole will badly slow the closed-loop transient response. Adding a real zero to the left of the pole at $z = 1$ will pull the closed-loop pole to the left for positive gain K, thereby improving the transient response of the closed-loop system.

Another common compensator is the notch filter. It is used when the plant has a pair of lightly damped open-loop poles. These poles can severely limit the range of useful feedback gains, K, because their closed-loop counterparts may become unstable for relatively small values of K. Adding a compensator that has a pair of complex conjugate zeros close to these poles will pull the closed-loop poles towards the zeros as K is increased. One must be careful about the placement of the zeros. If they are placed wrongly, the root locus from the undesirable poles to the added zeros will loop out into the unstable region before returning inside the unit circle. If they are properly placed, this will not happen. Again, one must also add a pair of poles, or the compensator will cause other serious problems, as explained in Section 6.1.

The use of lead and lag compensators is illustrated in the following example.

Design example

Consider a plant with $G(s) = 600/(s + 1)(s + 6)(s + 40)$. This is sampled at $T = 0.0167s$ resulting in $G(z) = 0.000386(z+3.095)(z+0.218)/(z-0.983)(z-0.905)(z - 0.513)$. The root locus for this plant is shown on the left in Fig. 6 as a solid line. Closing the loop with a gain of $K = 1$ results in the closed-loop step response shown at the right as a solid line. The rise time is 0.47, the settling time is 1.35, and the steady-state value is 0.71. There are two aspects of this design that one might want to improve. The step response is rather slow. We would like to make the rise and settling times smaller. The steady-state error in response to a unit step is rather large, 0.29. We would like to make it smaller. Note that increasing the gain from 1 to a larger value

would improve both of these aspects of the step response, but the cost would be a more oscillatory response with a larger overshoot as well as a less robust controller.

A lead compensator, $C_{lead}(z) = K(z - 0.905)/(z - 0.794)$, is added to reduce the rise and settling times without compromising either robustness or overshoot. The zero is placed directly on top of the middle pole of the original plant. The pole is placed so that the largest value of $u(k)$ in the step response of the closed-loop system is less than 4. The resulting root locus is shown as a dotted line in Fig. 6. Closing the loop with $K = 4$ results in the dotted step response shown on the right. The new rise time is 0.267, the settling time is 0.735, and the steady-state value is 0.821. Notice that the lead compensator has improved every aspect of the closed-loop step response. However, the steady-state error in response to a unit step input is still 0.179.

Finally, a lag compensator is added to further reduce the steady-state error in response to a unit step. Adding the lag element makes the complete controller $C_{leadlag}(z) = K(z - 0.905)(z - 0.985)/(z - 0.794)(z - 0.999)$. The pole of the lag compensator is placed close to $z = 1$. The zero is placed just to the right of the pole of the original plant at $z = 0.983$. With these choices, a reasonable gain pulls the added open-loop pole almost onto the added zero. This gives a small steady-state error without significantly compromising the transient response. The new root locus is shown as a dashed line in Fig. 6. The closed-loop step response using this controller is shown dashed at the right of the figure. The rise time is 0.284, the settling time is 0.668, and the steady-state value is 0.986. Note that the steady-state error is now less than 0.02 and the other aspects of the response are nearly as good as they were with only the lead compensator.

5.2 Frequency response-based design

There are two common reasons why one might base a control system design only on the frequency response plots, i.e., on plots of $|G(j\omega)|$ and $\angle G(j\omega)$ versus ω. First, there are systems for which the frequency response can be determined experimentally although an analytical expression for the transfer function is unknown. Although one could estimate a transfer function from this data, it is arguably better not to introduce additional modelling errors by doing this. Second, some systems that are very high order have relatively simple frequency responses. The best example of this is an electronic audio amplifier, which may have approximately 20 energy storage elements. Its transfer function would have denominator degree around 20. Its frequency response plots would be fairly simple, especially since its purpose is to amplify audio signals. In fact, this was the application that drove the work of Bode and Nyquist on feedback. It is also somewhat easier to design a lag compensator in the frequency domain.

One can use either the Bode plots or the Nichols chart of the open-loop system as the basis for the design. Both the gain and phase margin can be

Notice that the lead compensator has its zero to the right of its pole and the lag compensator has its zero to the left of its pole.

The principle behind both compensators is the same. Consider the real singularities (poles and zeros) of the open-loop system. Suppose that the rightmost real singularity is a pole. This open-loop real pole will move towards a real open-loop zero placed to its left when the loop is closed with a positive gain K. If the open-loop system has a pole near $z = 1$, it is usually possible to speed up the closed-loop transient response by adding a zero to its left. For several reasons (the most important will be explained in Section 6 on limitations of control) one should never add just a zero. Thus, one must add a real pole to the left of the added zero, thereby creating a lead compensator. This lead compensator will generally improve the transient response. The best value of the gain K can be determined using the root locus plot of the combined plant and lead compensator.

The lag compensator is used to reduce the steady-state error. This is done by adding a real pole near the point $z = +1$. Adding only a pole will badly slow the closed-loop transient response. Adding a real zero to the left of the pole at $z = 1$ will pull the closed-loop pole to the left for positive gain K, thereby improving the transient response of the closed-loop system.

Another common compensator is the notch filter. It is used when the plant has a pair of lightly damped open-loop poles. These poles can severely limit the range of useful feedback gains, K, because their closed-loop counterparts may become unstable for relatively small values of K. Adding a compensator that has a pair of complex conjugate zeros close to these poles will pull the closed-loop poles towards the zeros as K is increased. One must be careful about the placement of the zeros. If they are placed wrongly, the root locus from the undesirable poles to the added zeros will loop out into the unstable region before returning inside the unit circle. If they are properly placed, this will not happen. Again, one must also add a pair of poles, or the compensator will cause other serious problems, as explained in Section 6.1.

The use of lead and lag compensators is illustrated in the following example.

Design example

Consider a plant with $G(s) = 600/(s + 1)(s + 6)(s + 40)$. This is sampled at $T = 0.0167s$ resulting in $G(z) = 0.000386(z+3.095)(z+0.218)/(z-0.983)(z-0.905)(z - 0.513)$. The root locus for this plant is shown on the left in Fig. 6 as a solid line. Closing the loop with a gain of $K = 1$ results in the closed-loop step response shown at the right as a solid line. The rise time is 0.47, the settling time is 1.35, and the steady-state value is 0.71. There are two aspects of this design that one might want to improve. The step response is rather slow. We would like to make the rise and settling times smaller. The steady-state error in response to a unit step is rather large, 0.29. We would like to make it smaller. Note that increasing the gain from 1 to a larger value

would improve both of these aspects of the step response, but the cost would be a more oscillatory response with a larger overshoot as well as a less robust controller.

A lead compensator, $C_{lead}(z) = K(z - 0.905)/(z - 0.794)$, is added to reduce the rise and settling times without compromising either robustness or overshoot. The zero is placed directly on top of the middle pole of the original plant. The pole is placed so that the largest value of $u(k)$ in the step response of the closed-loop system is less than 4. The resulting root locus is shown as a dotted line in Fig. 6. Closing the loop with $K = 4$ results in the dotted step response shown on the right. The new rise time is 0.267, the settling time is 0.735, and the steady-state value is 0.821. Notice that the lead compensator has improved every aspect of the closed-loop step response. However, the steady-state error in response to a unit step input is still 0.179.

Finally, a lag compensator is added to further reduce the steady-state error in response to a unit step. Adding the lag element makes the complete controller $C_{leadlag}(z) = K(z - 0.905)(z - 0.985)/(z - 0.794)(z - 0.999)$. The pole of the lag compensator is placed close to $z = 1$. The zero is placed just to the right of the pole of the original plant at $z = 0.983$. With these choices, a reasonable gain pulls the added open-loop pole almost onto the added zero. This gives a small steady-state error without significantly compromising the transient response. The new root locus is shown as a dashed line in Fig. 6. The closed-loop step response using this controller is shown dashed at the right of the figure. The rise time is 0.284, the settling time is 0.668, and the steady-state value is 0.986. Note that the steady-state error is now less than 0.02 and the other aspects of the response are nearly as good as they were with only the lead compensator.

5.2 Frequency response-based design

There are two common reasons why one might base a control system design only on the frequency response plots, i.e., on plots of $|G(j\omega)|$ and $\angle G(j\omega)$ versus ω. First, there are systems for which the frequency response can be determined experimentally although an analytical expression for the transfer function is unknown. Although one could estimate a transfer function from this data, it is arguably better not to introduce additional modelling errors by doing this. Second, some systems that are very high order have relatively simple frequency responses. The best example of this is an electronic audio amplifier, which may have approximately 20 energy storage elements. Its transfer function would have denominator degree around 20. Its frequency response plots would be fairly simple, especially since its purpose is to amplify audio signals. In fact, this was the application that drove the work of Bode and Nyquist on feedback. It is also somewhat easier to design a lag compensator in the frequency domain.

One can use either the Bode plots or the Nichols chart of the open-loop system as the basis for the design. Both the gain and phase margin can be

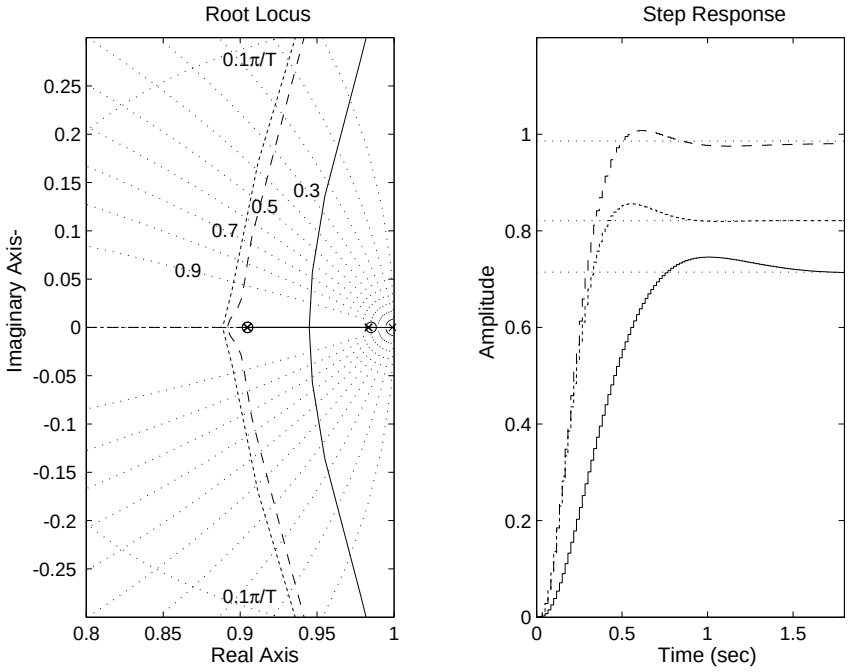

Fig. 6. The root loci and step responses for the design example

read directly from these graphs, making them the most important criteria for design in the frequency domain. There is a convenient relationship between the phase margin and the damping ratio for second-order systems such as (3). It is

$$\zeta \approx (phase\ margin)/100. \tag{14}$$

The effect of a pure gain controller, $C(s) = K$, on the Bode magnitude plot is simply a vertical shift by $20 \log |K|$. The effect on the Nichols chart is a vertical shift by the same amount. Using (14) and the gain and phase margins, the designer can choose a value of the gain K that meets the specifications as in the continuous-time case. If the specifications cannot be satisfied by a pure gain controller, then the various compensators can be tried.

The basic idea behind lead-lag compensation in the frequency domain is that the closed-loop transient response is dominated by the open-loop frequency response near the *gain* and *phase crossover* frequencies, defined to be the frequencies at which the gain crosses 0 dB and the phase crosses $-180°$. The steady-state behavior is determined by the low frequency characteristics of the open-loop frequency response. Thus, the general idea is to add a lag compensator whenever the closed-loop steady-state error is too large. The pole and zero of the lag compensator are placed at low enough frequencies so that they do not affect the open-loop frequency response near the crossover fre-

quencies. On the other hand, the lead compensator is added in the vicinity of the phase crossover frequency where it adds phase margin, thereby improving the transient response as suggested by (14).

The notch filter is used to cancel a large peak in the open-loop frequency response (a resonant peak). The reason it is called a notch filter is evident from its Bode plot. The notch filter has a "notch" in the magnitude of its transfer function. This notch is used to cancel the peak in the open-loop frequency response. The exact placement of the notch is tricky. See [6] for the details.

The design example developed previously using the root locus is repeated below in terms of frequency responses.

Design example revisited

The Bode and Nichols plots for the open-loop plant $G(z) = 0.000386(z + 3.095)(z+0.218)/(z-0.983)(z-0.905)(z-0.513)$ (the same as in Section 5.1), this time with an additional gain of 1.2, are shown in Fig. 7 as solid curves. Closing the loop with unity gain results in a gain margin of 22 dB, a phase margin of 83 degrees, a gain crossover frequency of 2.6 rad/s, and a phase crossover frequency of 14.4 rad/s. Although the gain is slightly higher than it was in our root locus-based design, the closed-loop step response is nearly the same as before, so it is not reproduced here. There is slightly more overshoot and the rise and settling times are slightly faster. We chose the higher gain to emphasize the similarity among the three frequency responses.

The same lead compensator as in the root locus design example is added to speed up the closed-loop response to a unit step. Because of the link between phase margin and damping ratio, ζ (see (14)) we know that increasing the phase margin will speed up the step response. The Bode and Nichols plots of $G(z)C_{lead}(z)$ with a gain of $K = 4$, exactly as in the root locus case, are shown dotted in Fig. 7. Note the slightly more positive phase angle in the critical region near the gain and phase crossover frequencies. The resulting gain margin is 19 dB; the phase margin is 78 degrees; the gain crossover is at 4.25 rad/s; the phase crossover is at 20.4 rad/s. We already know that the resulting closed-loop step response is considerably faster. If we did not know the root locus, we would have placed the maximum phase lead of the lead compensator close to the phase crossover of the original plant.

The same lag compensator as in Section 5.1 is added to reduce the steady-state error in response to a unit step. The Bode and Nichols plots of $G(z)C_{leadlag}(z)$ with a gain of $K = 4$, exactly as in the root locus case, are shown dashed in Fig. 7. The frequency response plots show that the lag compensator greatly increases the DC gain of the open-loop system ($C_{leadlag}(z)G(z)$) while making minimal changes to the frequency response near the critical frequencies. The resulting gain margin is 18 dB; the phase margin is 67 degrees; the gain crossover is at 4.31 rad/s; the phase crossover is at 19.7 rad/s. The lag compensator is placed so that all of its effects occur at lower frequencies than the critical ones.

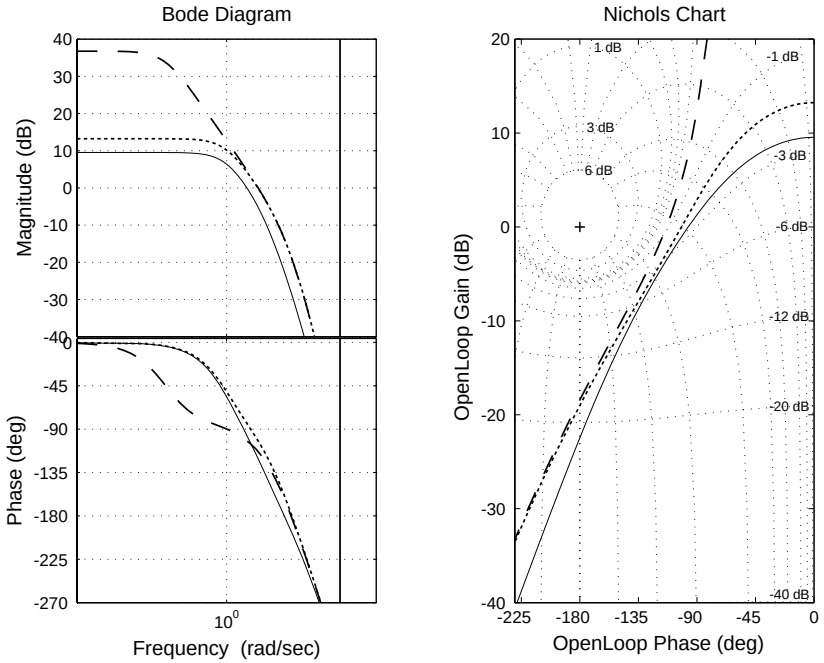

Fig. 7. The Bode and Nichols plots for the design example

5.3 PID control

There are many situations in which it would be inconvenient or impractical to measure the frequency response of the plant and in which the transfer function is either unknown or far too complicated to use for controller design. There are many good examples in the process industries, such as paper-making machines. In many of these applications the specifications are not too demanding. Again, the paper-making machine is illustrative: the transient response is not very important, but tight steady-state control of the thickness is. This is the paradigmatic use of PID control, although the method is also used for much more demanding applications, including many for which a good plant model is known.

The discrete-time (proportional + integral + derivative) (PID) controller is derived from the original continuous-time version. A realistic, as opposed to academic, version of the continuous-time PID controller is

$$C_{PID}(s) = K_P + K_I \frac{1}{s} + K_D \frac{s}{1 + sT_f}. \tag{15}$$

It is also common in practice to use $(\alpha R(s) - Y(s))$ as the input to the derivative term (coefficient K_D) instead of $(R(s) - Y(s))$. Often, α is set to zero. The continuous-time controller can be discretized in a variety of ways, each with

its advantages and disadvantages; see [1]. The most commonly used method is the backwards difference method, which is known to be well behaved. The result can be written as

$$u(k) = u_P(k) + u_I(k) + u_D(k), \tag{16}$$

where

$$u_P(k) = K_P(r(k) - y(k)) \tag{17}$$

$$u_I(k) = u_I(k-1) + K_I T(r(k) - y(k)) \tag{18}$$

$$u_D(k) = (1 + \frac{T}{T_f})^{-1} u_D(k-1) - \frac{K_D}{T_f}(\frac{1}{1 + \frac{T}{T_f}})(y(k) - y(k-1)), \tag{19}$$

where T is the sampling interval and T_f is a filtering coefficient.

One can purchase a PID controller as an essentially turnkey device. It is connected to the plant and the 3–5 parameters of the controller are then adjusted (tuned) to the particular plant. There is an extensive literature on tuning PID controllers dating back at least to Ziegler and Nichols [1]. The basic ideas are relatively simple if one sets the D-terms to zero. One straightforward tuning method is to set the D- and I-terms to zero and gradually increase the gain K_P just until the closed-loop step response becomes unstable. Reducing the gain by 50%, for example, will produce a closed-loop system with a gain margin of 6 dB. For a proportional controller this is regarded as a fairly good choice of gain. If this is sufficient to meet the specifications, there is no more to be done.

If the steady-state error is too large, an I-term must be added to the controller. Doing so adds a pole at $z = +1$, thereby eliminating the steady-state error in response to a unit step. It will also add a zero at $-K_I/K_P$ in the continuous-time case. Some modest fine-tuning of the two gains will improve the transient response without, of course, changing the steady-state error. The well-known but overly aggressive Ziegler–Nichols rules suggest decreasing K_P to 40% of the value of K_P that caused instability and then choosing $K_I = K_P/(0.8T_u)$, where T_u is the period of the oscillation that resulted when K_P was chosen to make the closed-loop system unstable.

Tuning the D-term is notoriously difficult. Its basic role is to add a zero and a pole to the controller. If chosen properly this zero and pole will act as a lead compensator and speed up the closed-loop transient response. See [1] for details.

If one has a good mathematical description of the plant, then either a root locus plot, a Bode plot, or a Nichols chart of the open-loop system can be used to choose the parameters of the PID controller (which is basically a lead-lag controller with the lag pole at $z = 1$) to achieve a desired step response.

It is now possible to buy "self-tuning" PID controllers. They are available from several manufacturers and they use a variety of tuning methods. The details are often proprietary. Generally, an operator commands the controller

to enter its tuning mode. The controller then tunes itself and switches to operate mode and stays there until it is again commanded to tune itself.

5.4 Design by emulation

In the discussion above, we have described the basics of *direct digital design*, meaning that the plant is first discretized (taking into account the effects of sampling and ZOH) and a discrete-time controller is designed.

An alternative is to initially ignore the effects of D/A and A/D conversion and design a continuous compensator $C(s)$ for the continuous time plant $G(s)$. The continuous-time compensator is then discretized to obtain $C_d(z)$. This procedure, known as *design by emulation*, may be used when a working continuous-time controller already exists or when the designer has very good intuition for continuous-time control.

The conversion of a continuous-time controller to an approximately equivalent discrete-time controller can be done in a variety of ways. Two simple and useful methods require only that s in the continuous-time controller be replaced by the appropriate formula involving z. They are:

- Backward rule: $s = \frac{(z-1)}{T_s}$
- Tustin's method: $s = \frac{2}{T_s}\frac{(z-1)}{(z+1)}$.

A third method is only slightly more complicated.

- Matched pole-zero (MPZ) method:

Recall that the poles of a continuous-time transfer function $C(s)$ are related to the poles of its z-transform $C_d(z) = \mathcal{Z}\{C(s)\}(z)$ by

$$z = e^{sT},$$

where T is the sampling period. One can then attempt to obtain a digital version of $C(s)$ by applying this relationship to its zeros as well as its poles (we stress that this represents only an approximation—the zeros are not related by $z = e^{sT}$). The resulting discrete-time transfer function $C_d(z)$ is then obtained with a minimum of calculations.

If $C(s)$ is strictly proper (the degree of its denominator is greater than that of its numerator) it is sometimes desirable to further modify the resulting $C(z)$ by multiplying it repeatedly by $(1+z^{-1})$ (adding zeros at $z=-1$) until the resulting transfer function has denominator degree equal to that of the numerator, or equal to that of the numerator minus one ("modified matched pole-zero method"). Doing so has the effect of "averaging" past and present inputs. The MPZ method requires inputs of up to $e(k+1)$ in order to produce $u(k+1)$. This may be undesirable in applications where the time to compute $u(k+1)$ is significant compared with the sampling period. The modified MPZ method does not suffer from this drawback, as it requires only "past" inputs to produce the current output.

The approximations obtained via the methods described here are typically useful at frequencies below 1/4 of the sampling rate. Furthermore, because design by emulation ignores the effect of the ZOH, the performance of the resulting controllers yields reasonable results at sampling rates that are approximately 20 times or higher than the bandwidth of the continuous-time plant. For lower sampling rates, it is important to analyze the resulting closed-loop system in discrete time to ensure adequate performance.

5.5 Advanced methods of control

One method of control design is only slightly more involved than those discussed so far and lends itself very well to digital implementation. It is known as the two-degrees-of-freedom (2DOF) method. The basic idea is to divide the controller into two nearly independent components as shown in Fig. 8. The

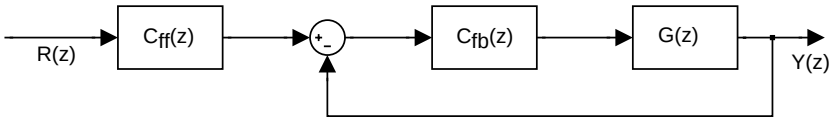

Fig. 8. A 2DOF controller

feedback component of the controller, $C_{fb}(z)$, is designed to deal primarily with disturbances while the feedforward component $C_{ff}(z)$ deals mainly with the response to a command signal $R(z)$. Although $C_{ff}(z)$ acts open loop, it can be realized very accurately on the computer. Thus, there should be minimal uncertainty associated with its behavior. The feedback portion of the controller, $C_{fb}(z)$, is designed to minimize the effects of plant uncertainty and to make the closed-loop system have a gain of one within the frequency range of possible inputs.

There is a very large literature on controller design. There are state-space methods for arbitrarily placing the poles of the closed-loop system assuming only that the open-loop system is controllable and observable [3]. Because it is not at all obvious where the closed-loop poles should be placed, there is also a large literature on optimal control. For linear systems, the linear quadratic regulator and the H_2 and H_∞ methods are particularly important [15]. There has also been much research and some applications in the field of nonlinear control. Introductions to all of these topics can be found in [9].

6 Limitations on Control Systems

It is very important for the control system designer to be aware of several limitations on the stability and performance of real control systems. These limitations are due to inaccuracies in the plant model, inevitable disturbances that

enter the system, actuator saturation, and fundamental unavoidable trade-offs in the design. The first step in appreciating these limitations is to examine the more realistic picture of a SISO control loop in Fig. 9, where $G_n(z)$ is the

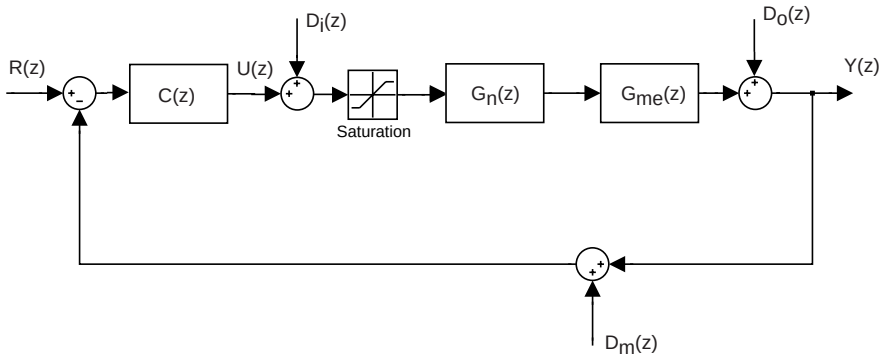

Fig. 9. A feedback control system with input, output, and measurement disturbances

nominal plant which is used in the design. The actual plant is $G_{me}(z)G_n(z)$ where $G_{me}(z) = (1+G_\Delta(z))$ denotes modelling errors. Generally, only bounds are known for the multiplicative modelling error, $G_\Delta(z)$. In particular, in a networked and embedded control system the phase of $G_\Delta(z)$ is known only to lie within limits determined by the timing accuracy of the system. The additional inputs are $D_i(z)$ representing input disturbances, $D_o(z)$ for output disturbances, and $D_m(z)$ for measurement noise. Note that we have omitted any sensor dynamics in order to focus on the most essential aspects of robustness and sensitivity.

LTI control systems are also limited by the fundamental Bode gain-phase relation. The precise theorem can be found in [14]. A simple rule of thumb based on Bode's result is that each $-20n$ dB/decade of reduction in the open-loop gain implies $\approx -90°n$ of phase shift, where n is a positive integer. This link between gain and phase is easily seen in lead and lag compensators. A lead compensator basically adds positive phase to improve the transient performance of the closed-loop system. The price paid for this positive phase is an undesirable increase in high frequency gain. A lag compensator is used to add to the DC gain of the open-loop system, thus decreasing the steady-state error of the closed-loop system. The price paid for this improvement is an undesirable negative phase shift.

6.1 Sensitivity to disturbances

The effect of the disturbances on the performance of the control system can be studied by writing the transfer functions from each disturbance to $Y(z)$

and $U(z)$. The effect of the disturbances on $U(z)$ is particularly important because of saturation. The transfer functions are written using the nominal plant so they are nominal sensitivities.

$$\begin{aligned}
\frac{Y(z)}{R(z)} &= G_{cl}(z) = \frac{G_n(z)C(z)}{1+G_n(z)C(z)} \\
\frac{Y(z)}{D_i(z)} &= S_{io}(z) = \frac{G_n(z)}{1+G_n(z)C(z)} \\
\frac{Y(z)}{D_o(z)} &= S_o(z) = \frac{1}{1+G_n(z)C(z)} \\
\frac{Y(z)}{D_m(z)} &= -G_{cl}(z) = -\frac{G_n(z)C(z)}{1+G_n(z)C(z)} \\
\frac{U(z)}{D_m(z)} &= S_{ou}(z) = \frac{C(z)}{1+G_n(z)C(z)}
\end{aligned} \tag{20}$$

Notice that $S_{ou}(z)$ is also the transfer function from $R(z)$ and $D_o(z)$ to $U(z)$, which explains why it is a bad idea to use a zero as a lead compensator without also including a pole. Such a choice would result in $C(z) = (z - z_l)$, and this would cause $S_{ou}(z)$ to amplify any high frequency components of $R(z)$, $D_o(z)$, and $D_m(z)$. This would result in actuator saturation on noise. Ultimately, the placement of the pole in a lead compensator and hence, the amount of phase lead possible is limited by the amplitude of the disturbances.

A fundamental limit on controller performance is easily derived from the transfer functions above:

$$G_{cl}(z) + S_o(z) = 1, \quad \text{for all } z \in \mathbb{C}. \tag{21}$$

Another limitation follows from the fact that $G_{cl}(z)$ is the transfer function from both $-D_m(z)$ and $R(z)$ to $Y(z)$. This makes it very desirable to keep $|C(z)|$ small at those frequencies at which $R(z)$ is zero. A typical example is in aircraft SCAS where pilot inputs and aircraft maneuvers are known to be limited to relatively low frequencies, implying that any signal at high frequency must be noise. Now consider the implications of (21) for a closed-loop system having the property that $|G_{cl}(z)|$ is small at high frequency. Such a system will pass output disturbances at those frequencies without attenuation.

6.2 Robustness

It is important that the closed-loop system remain stable despite the differences between the nominal plant used for the controller design and the real plant. There has been extensive research devoted to robust stability in recent years. There are many results available; see [9, 15]. The following is a simple example from [8].

Theorem 1. *Consider a plant with nominal transfer function $G_n(z)$ and actual transfer function $G_n(z)(1 + G_\Delta(z))$. Assume that $C(z)$ is a controller that achieves internal stability for $G_n(z)$. Assume also that $G_n(z)C(z)$ and $G_n(z)(1 + G_\Delta(z))C(z)$ both have the same number of unstable poles. Then a sufficient condition for stability of the true feedback loop obtained by applying the controller $C(z)$ to the actual plant is that*

$$|G_{cl}(z)||G_\Delta(z)| = \left| \frac{G_n(z)C(z)}{1 + G_n(z)C(z)} \right| |G_\Delta(z)| < 1. \tag{22}$$

The proof is a straightforward application of the Nyquist stability theorem. Notice that the theorem holds regardless of the uncertainties in the phase. Thus, it is valuable in ensuring that delays due to networking and computing cannot compromise the stability of the real closed-loop system.

The use of the theorem can be understood by dividing the frequency response of the nominal open-loop system and compensator, $G_n(z)C(z)$, into three regions. In the low frequency region, it is normally true that $|G_\Delta(z)|$ is small. Thus, the controller can have high gain and the nominal closed-loop system can have a magnitude near one without compromising stability robustness. At high frequencies, $|G_\Delta(z)|$ is usually large but $|G_n(z)C(z)|$ is small. Again, stability robustness is not a problem because the nominal closed-loop system has small magnitude. The critical region is the frequency range near the gain and phase crossover frequencies. In this region, the bounds on $|G_\Delta(z)|$ are very important.

6.3 Trade-offs

The following theorem [14], due originally to Bode [4], proves that there is a fundamental trade-off inherent in any attempt to reduce the sensitivity, $S_o(z)$, of a closed-loop system.

Theorem 2. *Consider a SISO LTI discrete-time open-loop system $G_n(z)C(z)$ with its corresponding stable closed-loop system $G_{cl}(z) = \frac{G_n(z)C(z)}{1+G_n(z)C(z)}$ and sensitivity $S_o(z) = \frac{1}{1+G_n(z)C(z)}$. Then*

$$\int_{-\pi}^{\pi} \ln |S_o(e^{j\omega})| d\omega = 2\pi \sum_i (\ln |p_i| - \ln |\gamma + 1|) \tag{23}$$

where the p_i are the unstable poles of the open-loop system and $\gamma = \lim_{z\to\infty} G_n(z)C(z)$.

Notice that if the open-loop system is stable and strictly proper, then the theorem implies that $\int_{-\pi}^{\pi} \ln |S_o(e^{j\omega})| d\omega = 0$. Typically, one wants to design the controller to keep the sensitivity small at low frequencies. The theorem proves that the inevitable consequence is that the controller increases the sensitivity at high frequencies.

7 Beyond This Introduction

There are many good textbooks on classical control. Two popular examples are [5] and [6]. A less typical and interesting alternative is the recent textbook [8]. All three of these books have at least one chapter devoted to the basics of digital control. Textbooks devoted to digital control are less common, but there are some available. The best known is probably [7]. Other possibilities are [2, 12, 13] An excellent book about PID control is the one by Aström and Hägglund [1]. Good references on the limitations of control are [10] and [11]. Bode's book [4] is still interesting, although the emphasis is on vacuum tube circuits.

References

1. K. J. Åström and T. Hägglund. *PID Controllers: Theory, Design and Tuning*. International Society for Measurement and Control, Seattle, WA, 2nd edition, 1995.
2. K. J. Åström and B. Wittenmark. *Computer Controlled Systems*. Prentice-Hall, Englewood Cliffs, NJ, 3rd edition, 1996.
3. P. R. Bélanger. *Control Engineering: A Modern Approach*. Saunders College Publishing, Stamford, CT, 1995.
4. H. W. Bode. *Network Analysis and Feedback Amplifier Design*. R. E. Krieger Pub. Co., Huntington, NY, 1975.
5. R. C. Dorf and R. H Bishop. *Modern Control Systems*. Prentice-Hall, Upper Saddle River, NJ, 10th edition, 2004.
6. G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamical Systems*. Prentice-Hall, Upper Saddle River, NJ, 4th edition, 2002.
7. G. F. Franklin, J. D. Powell, and M. L. Workman. *Digital Control of Dynamic Systems*. Prentice-Hall, Upper Saddle River, NJ, 3rd edition, 1997.
8. G. C. Goodwin, S. F. Graebe, and M. E. Salgado. *Control System Design*. Prentice-Hall, Upper Saddle River, NJ, 2000.
9. W. S. Levine, editor. *The Control Handbook*. CRC Press, Boca Raton, FL, 1996.
10. D. P. Looze and J. S. Freudenberg. *Frequency Domain Properties of Scalar and Multivariable Feedback Systems*. Springer-Verlag, Berlin, 1988.
11. D. P. Looze and J. S. Freudenberg. Tradeoffs and limitations in feedback systems. In W. S. Levine, editor, *The Control Handbook*, pages pp. 537–550. CRC Press, Boca Raton, FL, 1996.
12. M. S. Santina and A. R. Stubberud. *Discrete-Time Equivalents to Continuous-Time Systems*. Eolss Publishers Co. Ltd., Oxford, U.K., 2004.
13. M. S. Santina, A. R. Stubberud, and G. H. Hostetter. *Digital Control System Design*. International Thomson Publishing, Stamford, CT, 2nd edition, 1994.
14. B.-F. Wu and E. A. Jonckheere. A simplified approach to Bode's theorem for continuous-time and discrete-time systems. *IEEE Transactions on Automatic Control*, 37(11):1797–1802, Nov. 1992.
15. K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, Upper Saddle River, NJ, 1995.