Farid Golnaraghi, Benjamin C. Kuo - Automatic Control Systems-McGraw-Hill (2017)

Design of Control Systems

11-1 INTRODUCTION

We now utilize all the foundations and analyses that we have provided in the preceding chapters in the ultimate goal of design of control systems. Starting with the controlled process such as that shown by the block diagram in Fig. 11-1, control system design involves the following three steps:

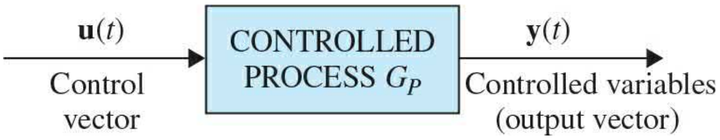


Figure 11-1 Controlled process.

1. Use design specifications to determine what the system should do and how to do.
2. Determine the controller or compensator configuration, relative to how it is connected to the controlled process.
3. Determine the parameter values of the controller to achieve the design goals.

These design tasks are explored further in the following sections.

Learning Outcomes

After successful completion of this chapter, you will be able to

1. Design simple control systems using time-domain and frequency-domain approaches.
2. Incorporate various controllers, including proportional, derivative, integral, lead, and lag, into your control system for simple processes.

3. Use MATLAB to investigate the time- and frequency-domain performance of the control systems.
4. Use the MATLAB SISO design tool to expedite the design process.

11-1-1 Design Specifications

As discussed in Chap. 7, we use design specifications to describe the expected performance of a system for a given input. These specifications are unique to individual applications and often include specifications about **relative stability**, **steady-state accuracy** (**error**), **transient-response characteristics**, and **frequency-response characteristics**. In some applications, there may be additional specifications on **sensitivity to parameter variations**, that is, **robustness**, or **disturbance rejection**.

The design of linear control systems can be carried out in either the time domain or the frequency domain. For instance, **steady-state accuracy** is often specified with respect to a step input, a ramp input, or a parabolic input, and the design to meet a certain requirement is more conveniently carried out in the time domain. Other specifications, such as **maximum overshoot, rise time**, and **settling time,** are all defined for a unit-step input and, therefore, are used specifically for time-domain design. We have learned that relative stability is also measured in terms of **gain margin**, **phase margin**, and $\boldsymbol{M_r}$. These are typical frequency-domain specifications, which should be used in conjunction with such tools as the Bode plot, polar plot, gain-phase plot, and Nichols chart.

We have shown that, for a second-order prototype system, there are simple analytical relationships between some of these time-domain and frequency-domain specifications. However, for higher-order systems, correlations between time-domain and frequency-domain specifications are difficult to establish. As pointed out earlier, the analysis and design of control systems is pretty much an exercise of selecting from several alternative methods for solving the same problem.

Thus, the choice of whether the design should be conducted in the time domain or the frequency domain depends often on the preference of the designer. We should be quick to point out, however, that in most cases, time-domain specifications such as maximum overshoot, rise time, and settling time are usually used as the final measure of system performance. To an inexperienced designer, it is difficult to comprehend the physical connection between frequency-domain specifications such as gain and phase margins and resonance peak to actual system performance. For instance, does a gain margin of 20 dB guarantee a maximum overshoot of less than 10 percent? To a designer it makes more sense to specify, for example, that the maximum overshoot should be less than 5 percent and a settling time less than 0.01 s. It is less obvious what, for example, a phase margin of 60° and an M_r of less than 1.1 may bring in system performance. The following outline will hopefully clarify and explain the choices and reasons for using time-domain versus frequency-domain specifications.

1. Historically, the design of linear control systems was developed with a wealth of graphical tools such as the Bode plot, Nyquist plot, gain-phase plot, and Nichols chart, which are all carried out in the frequency domain. The advantage of these tools is that they can all be sketched by following approximation methods without detailed plotting. Therefore, the designer can carry out designs using frequency-domain specifications such as **gain margin, phase margin**, $\boldsymbol{M_r}$, and the like. High-order systems do not generally pose any particular problem. For certain types of controllers, design procedures in the frequency domain are available to reduce the trial-and-error effort to a minimum.
2. Design in the time domain using such performance specifications as **rise time**, **delay time**, **settling time**, **maximum overshoot**, and the like is possible *analytically* only for second-order systems or for systems that can be approximated by second-order systems. General design procedures using time-domain specifications are difficult to establish for systems with an order higher than the second.

The development and availability of high-powered and user-friendly computer software, such as MATLAB, is rapidly changing the practice of control system design, which until recently had been dictated by historical development. Now with MATLAB, the designer can go through a large number of design runs using the time-domain specifications within a matter of minutes. This diminishes considerably the historical edge of the frequency-domain design, which is based on the convenience of performing graphical design manually.

Throughout this chapter, we have incorporated small MATLAB toolboxes to help your understanding of the examples, and, at the end of the chapter, we introduce the MATLAB SISO design tool that would enhance your ability to design controllers using the root-locus and frequency-domain approaches.

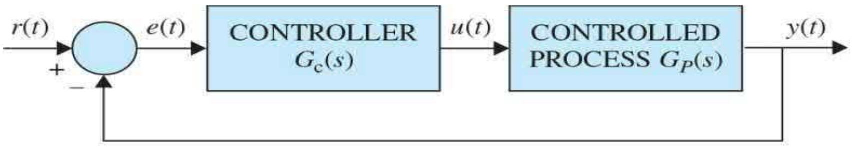
Finally, it is generally difficult (except for an experienced designer) to select a meaningful set of frequency-domain specifications that will correspond to the desired time-domain performance requirements. For example, specifying a phase margin of 60° would be meaningless unless we know that it corresponds to a certain maximum overshoot. As it turns out, to control maximum overshoot, usually one has to specify at least phase margin and M_r. Eventually, establishing an intelligent set of frequency-domain specifications becomes a trial-and-error process that precedes the actual design, which often is also a trial-and-error effort. However, frequency-domain methods are still valuable in interpreting noise rejection and sensitivity properties of the system, and, most important, they offer another perspective to the design process. Therefore, in this chapter the design techniques in the time domain and the frequency domain are treated **side by side**, so that the methods can be easily compared.

11-1-2 Controller Configurations

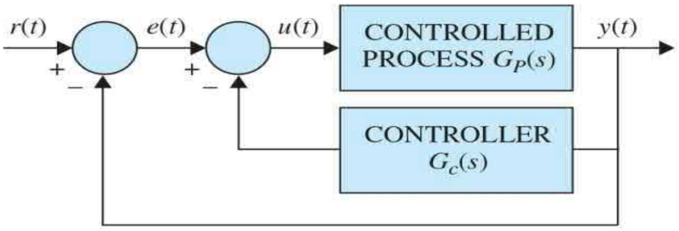
In general, the dynamics of a linear controlled process can be represented by the block diagram shown in Fig. 11-1. The design objective is to have the controlled variables, represented by the output vector **y**(*t*), behave in certain desirable ways. The problem essentially involves the determination of the control signal **u**(*t*) over the prescribed time interval so that the design objectives are all satisfied.

Most of the conventional design methods in control systems rely on the so-called **fixed-configuration design** in that the designer at the outset decides the basic configuration of the overall designed system and decides where the controller is to be positioned relative to the controlled process. The problem then involves the design of the elements of the controller. Because most control efforts involve the modification or compensation of the system-performance characteristics, the general design using fixed configuration is also called **compensation**.

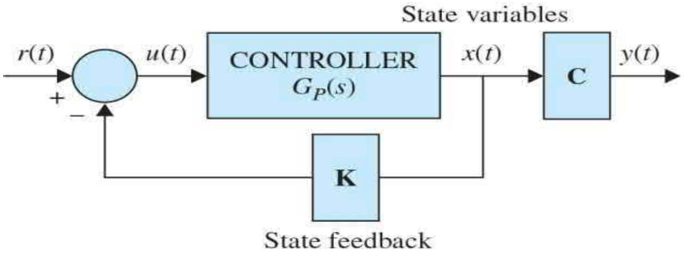
Figure 11-2 illustrates several commonly used system configurations with controller compensation. These are described briefly as follows:


r(t)
e(t)
CONTROLLER
G_c(s)
u(t)
CONTROLLED
PROCESS G_P(s)
y(t)
+
−

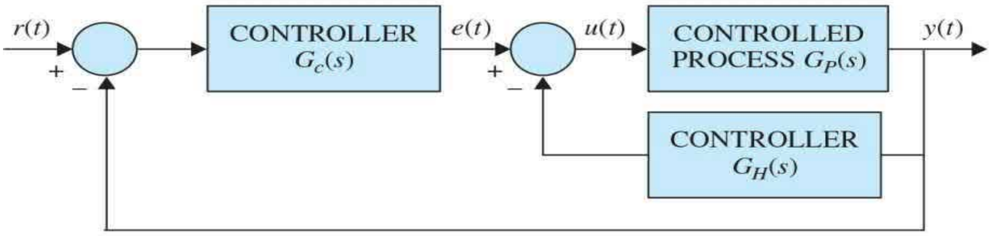
(a)


r(t)
e(t)
u(t)
CONTROLLED
PROCESS G_P(s)
y(t)
+
−
+
−
CONTROLLER
G_c(s)

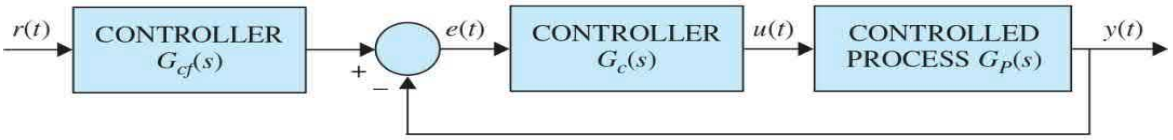
(b)


State variables
r(t)
u(t)
CONTROLLER
G_P(s)
x(t)
C
y(t)
+
−
K
State feedback

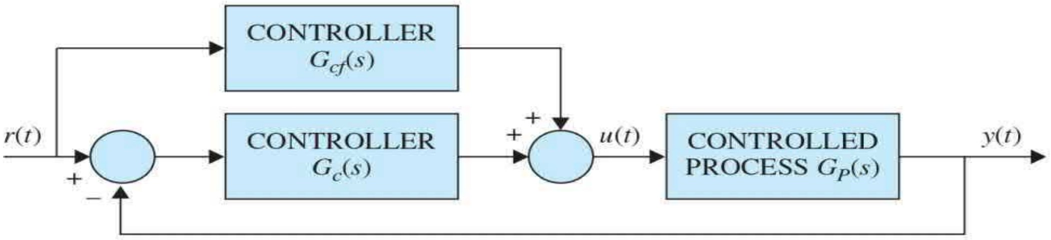
(c)


r(t)
CONTROLLER
G_c(s)
e(t)
u(t)
CONTROLLED
PROCESS G_P(s)
y(t)
+
−
+
−
CONTROLLER
G_H(s)

(d)


r(t)
CONTROLLER
G_cf(s)
e(t)
CONTROLLER
G_c(s)
u(t)
CONTROLLED
PROCESS G_P(s)
y(t)
+
−

(e)


CONTROLLER
G_cf(s)
r(t)
CONTROLLER
G_c(s)
+
+
u(t)
CONTROLLED
PROCESS G_P(s)
y(t)
+
−

(f)

Figure 11-2 Various controller configurations in control-system compensation. (*a*) Series or cascade compensation. (*b*) Feedback compensation. (*c*) State-feedback control. (*d*) Series-feedback compensation (two degrees of freedom). (*e*) Forward compensation with series compensation (two degrees of freedom). (*f*) Feedforward compensation (two degrees of freedom).

- ***Series (cascade) compensation.*** Figure 11-2*a* shows the most commonly used system configuration with the controller placed in series with the controlled process, and the configuration is referred to as **series** or **cascade compensation**.
- ***Feedback compensation.*** In Fig. 11-2*b*, the controller is placed in the minor feedback path, and the scheme is called **feedback compensation**.
- ***State-feedback compensation.*** Figure 11-2*c* shows a system that generates the control signal by feeding back the state variables through constant real gains, and the scheme is known as **state feedback**. The problem with state-feedback control is that, for high-order systems, the large number of state variables involved would require a large number of transducers to sense the state variables for feedback. Thus, the actual implementation of the state-feedback control scheme may be costly or impractical. Even for low-order systems, often not all the state variables are directly accessible, and an **observer** or **estimator** may be necessary to create the estimated state variables from measurements of the output variables.
- ***Series-feedback compensation.*** Figure 11-2*d* shows the series-feedback compensation for which a series controller and a feedback controller are used.
- ***Feedforward compensation.*** Figure 11-2*e* and *f* shows the so-called **feedforward compensation**. In Fig. 11-2*e*, the feedforward controller $G_{cf}(s)$ is placed in series with the closed-loop system, which has a controller $G_c(s)$ in the forward path.

In Fig. 11-2*f*, the feedforward controller $G_{cf}(s)$ is placed in parallel with the forward path. The key to the feedforward compensation is that the controller $G_{cf}(s)$ is not in the loop of the system, so it does not affect the roots of the characteristic equation of the original system. The poles and zeros of $G_{cf}(s)$ may be selected to add or cancel the poles and zeros of the closed-loop system transfer function.

The compensation schemes shown in Fig. 11-2*a* to *c* all have one degree of freedom in that there is only one controller in each system, even though the controller may have more than one parameter that can be varied. The disadvantage with a one-degree-of-freedom controller is that the performance criteria that can be realized are limited. For example, if a system is to be designed to achieve a certain amount of relative stability, it may have poor sensitivity to parameter variations. Or if the roots of the characteristic equation are selected to provide a certain amount of relative damping, the maximum overshoot of the step response may still be excessive because of the zeros of the closed-loop transfer function. The compensation schemes shown in Fig. 11-2*d* to *f* all have two degrees of freedom.

One of the commonly used controllers in the compensation schemes just described is a PID controller, which applies a signal to the process that is proportional to the actuating signal in addition to adding integral and derivative of the actuating signal. Because these signal components are easily realized and visualized in the time domain, PID controllers are commonly designed using time-domain methods. In addition to the PID-type controllers, lead, lag, lead-lag, and notch controllers are also frequently used. The names of these controllers come from properties of their respective frequency-domain characteristics. As a result, these controllers are often designed using frequency-domain concepts. Despite these design tendencies, however, all control system designs will benefit by viewing the resulting design from both time- and frequency-domain viewpoints. Thus, both methods will be used extensively in this chapter.

PID controllers are the most common controllers used in industrial applications.

It should be pointed out that these compensation schemes are by no means exhaustive. The details of these compensation schemes will be discussed in later sections of this chapter. Although the systems illustrated in Fig. 11-2 are all for continuous-data control, the same configurations can be applied to discrete-data control, in which case the controllers are all digital, with the necessary interfacings and signal converters.

11-1-3 Fundamental Principles of Design

After a controller configuration is chosen, the designer must choose a controller type that, with proper selection of its element values, will satisfy all the design specifications. The types of controllers available for control-system design are bounded only by one’s imagination. Engineering practice usually dictates that one choose the simplest controller that meets all the design specifications. In most cases, the more complex a controller is, the more it costs, the less reliable it is, and the more difficult it is to design. Choosing a specific controller for a specific application is often based on the designer’s past experience and sometimes intuition, and it entails as much *art* as it does *science*. As a novice, you may initially find it difficult to make intelligent choices of controllers with confidence. By understanding that confidence comes only through experience, this chapter provides guided experiences that illustrate the basic elements of control system designs.

After a controller is chosen, the next task is to choose controller parameter values. These parameter values are typically the coefficients of one or more transfer functions making up the controller. The basic design approach is to use the analysis tools discussed in the previous chapters to determine how individual parameter values influence the design specifications and, finally, system performance. Based on this information, controller parameters are selected so that all design specifications are met. While this process is sometimes straightforward, more often than not it involves many design iterations since controller parameters usually interact with each other and influence design specifications in conflicting ways. For example, a particular parameter value may be chosen so that the maximum overshoot is satisfied, but in the process of varying another parameter value in an attempt to meet the rise-time requirement, the maximum overshoot specification may no longer be met! Clearly, the more design specifications there are and the more controller parameters there are, the more complicated the design process becomes.

In carrying out the design either in the time domain or the frequency domain, it is important to establish some basic guidelines or design rules. Keep in mind that time-domain design usually relies heavily on the *s*-plane and the root loci. Frequency-domain design is based on manipulating the gain and phase of the loop transfer function so that the specifications are met.

In general, it is useful to summarize the time-domain and frequency-domain characteristics so that they can be used as guidelines for design purposes.

1. Complex-conjugate poles of the closed-loop transfer function lead to a step response that is underdamped. If all system poles are real, the step response is overdamped. However, zeros of the closed-loop transfer function may cause overshoot even if the system is overdamped.
2. The response of a system is dominated by those poles closest to the origin in the *s*-plane. Transients due to those poles farther to the left decay faster.
3. The farther to the left in the *s*-plane the system's dominant poles are, the faster the system will respond and the greater its bandwidth will be.
4. The farther to the left in the *s*-plane the system's dominant poles are, the more expensive it will be and the larger its internal signals will be. While this can be justified analytically, it is obvious that striking a nail harder with a hammer drives the nail in faster but requires more energy per strike. Similarly, a sports car can accelerate faster, but it uses more fuel than an average car.
5. When a pole and zero of a system transfer function nearly cancel each other, the portion of the system response associated with the pole will have a small magnitude.
6. Time-domain and frequency-domain specifications are loosely associated with each other. Rise time and bandwidth are inversely proportional. Larger phase margin, larger gain margin, and lower M_r will improve damping.

11-2 DESIGN WITH THE PD CONTROLLER

In most examples of control systems we have discussed thus far, the controller has been typically a simple amplifier with a constant gain *K*. This type of control action is formally known as **proportional control** because the control signal at the output of the controller is simply related to the input of the controller by a proportional constant.