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

K. J. Astrom, T. Hagglund - Advanced PID Control, 2006, ISA

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:

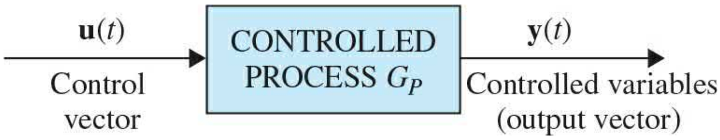


Figure 11-1 Controlled process.

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 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.

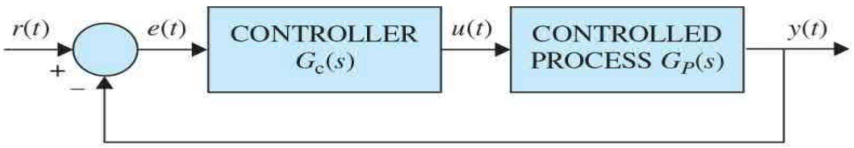
Finally, it is generally difficult (except for an experienced designer) to select a meaningful set of frequency-domain specifications that will correspond to the desired time-domain performance requirements. For example, specifying a phase margin of 60° would be meaningless unless we know that it corresponds 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.

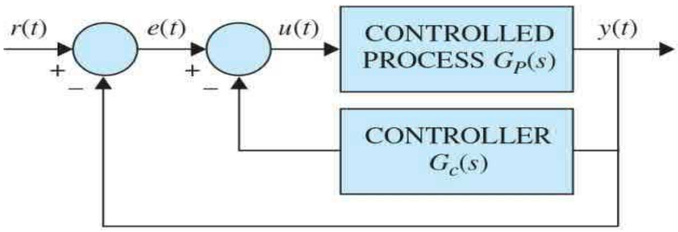
Most of the conventional design methods in control systems rely on the so-called **fixed-configuration design** in that the designer at the outset decides the basic configuration of the overall designed system and decides where the controller is to be positioned relative to the controlled process. The 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:



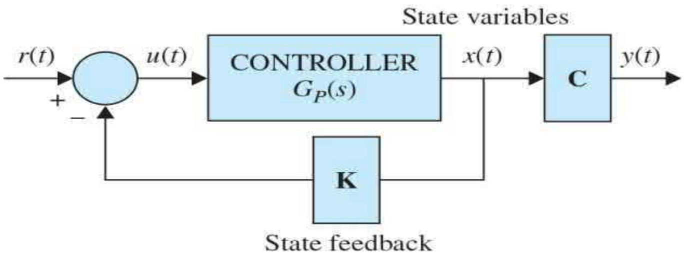
r(t)
e(t)
CONTROLLER $G_c(s)$
u(t)
CONTROLLED PROCESS $G_P(s)$
y(t)
+
−

(a)



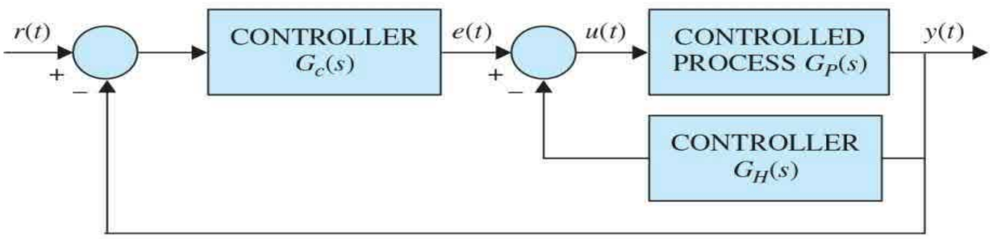
r(t)
e(t)
u(t)
CONTROLLED PROCESS $G_P(s)$
y(t)
CONTROLLER $G_c(s)$
+
−
+
−

(b)



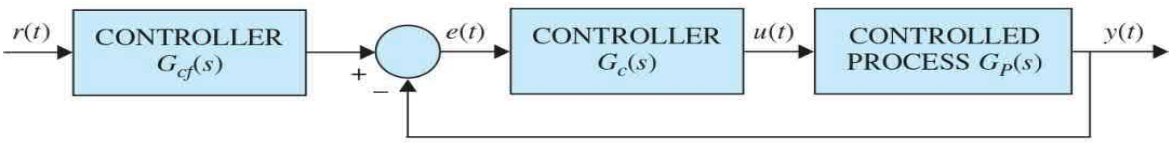
State variables
r(t)
u(t)
CONTROLLER $G_P(s)$
x(t)
C
y(t)
K
State feedback
+
−

(c)



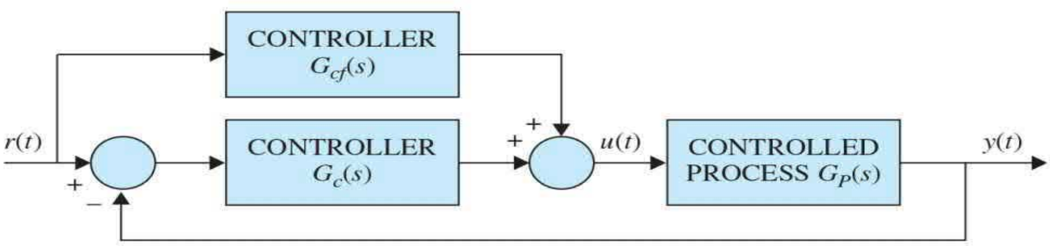
r(t)
CONTROLLER $G_c(s)$
e(t)
u(t)
CONTROLLED PROCESS $G_P(s)$
y(t)
CONTROLLER $G_H(s)$
+
−
+
−

(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.

4

Controller Design

4.1 Introduction

Control system design is a very rich field. There have been substantial advances over the past 50 years that have resulted in much insight and understanding as well as specific design methods. This development has been augmented by the advances in computing and the development of computer-based design tools. Broadly speaking, PID controllers have been designed using two different approaches; model-based control and direct tuning. The model based approaches start with a simple mathematical model of the process. Very simple models have been used, typically a first-order system with a time delay. In direct tuning a controller is applied to the process, and some simple experiments are performed to arrive at the controller parameters. Because of the simplicity of models and the controller special methods have been developed for PID control. From 1990 there has been a significant increase in the interest in design of PID controllers, partially motivated by the needs of automatic tuning devices for such controllers.

To develop design methods it is necessary to realize that there is a very wide range of different types of control problems even if the controller is restricted to PID. Some typical examples are:

- Design of a simple controller for a non-critical application.
- Design of a controller for a special process that minimizes fluctuations in important control variables.
- Development of a design technique that can be used in a universal autotuner for PID control.

There are also a number of important non-technical issues that should be considered: What is the time and effort required to apply the method? What is the knowledge level required of the user? A solution to the design problem should also give an understanding of when it is beneficial to add derivative action to a PI controller and when even more complex controllers should be considered.

This chapter gives an overview of ideas and concepts that are relevant for

PID control. It is attempted to bring design of PID controllers more into the mainstream of control design.

4.2 A Rich Variety of Control Problems

Before discussing specific tuning methods it is useful to realize that there is a wide range of control problems with very diverse goals. Some examples are: steady-state regulation, set-point tracking and path following, and control of buffers and surge tanks.

The goal of steady-state regulation is to keep process variables close to desired values. The key problems are caused by load disturbances, measurement noise, and process variations. Steady-state regulation is very common in process control.

In set-point tracking it is attempted to make process variables follow a given time function or a given curve. These problems typically occur in motion control and robotics. In some cases, for example, machine tool control or robotics, the demand on tracking precision is very severe. In other cases, for example, moving robots, the requirements are less demanding. There is a significant difference between tracking a given time curve and path following, which typically involves control of several variables.

Buffers are common in the industrial production. They are used to smooth variations between different production processes, both in process control and in discrete manufacturing. In process control they are often called *surge tanks*. Buffers are also common in computing systems. They are used in servers to smooth variations in demand of clients, and they are used in computer networks to smooth variations in the load. Buffers are also key elements in supply chains where effective buffer control has a major impact on profitability. The buffer levels should fluctuate; otherwise the buffer does not function. Ideally, no control should be applied unless there is a risk of over- or underflow. An integrating controller with low gain and a scheduling that gives higher gains at the buffer limits are commonly used.

The key issues in many of the control problems are attenuation of load disturbances, injection of measurement noise, robustness to process variations, and set-point following. The relative importance of these factors and the requirements vary from application to application, but all factors must be considered.

4.3 Feedback Fundamentals

A block diagram of a basic feedback loop with a controller having two degrees of freedom is shown in Figure 4.1. The process is represented by the block P. The controller is represented by the feedback block C and the feedforward

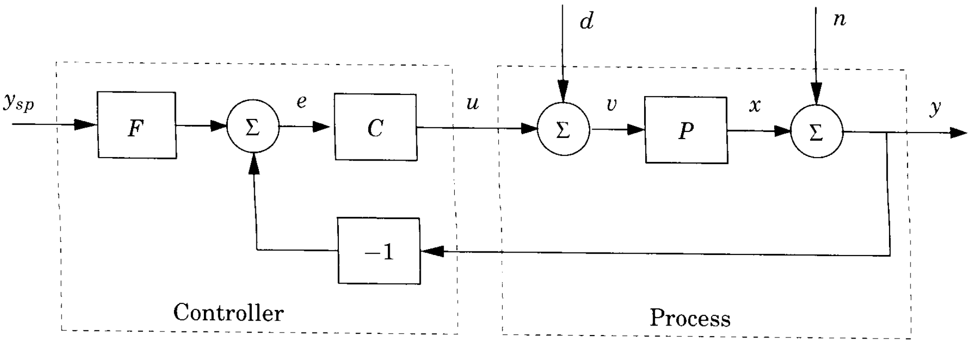


Figure 4.1 Block diagram of a basic feedback loop having two degrees of freedom.

part F. For an ideal PID controller with set-point weighting we have

$$
\begin{aligned}
C(s) &= K\Big(1 + \frac{1}{sT_i} + sT_d\Big) \\
F(s) &= \frac{b + \frac{1}{sT_i} + csT_d}{1 + \frac{1}{sT_i} + sT_d}.
\end{aligned}
\tag{4.1}
$$

Compare with (3.7) and (3.20). The signal u is the control signal, and the signal x is the real process variable. Information about x is obtained from the sensor signal y, which is corrupted by measurement noise n. The signal d represents load disturbances that drive the system away from its desired state. This signal can enter the process in different ways; in Figure 4.1 it is assumed that it acts on the process input.

The goal of control design is to determine the transfer functions C and F so that the process variable x is close to the set point y_{sp} in spite of load disturbances, measurement noise, and process uncertainties. The feedback can reduce the effect of load disturbances. Because of the feedback measurement noise is fed back into the system. It is essential to make sure that this does not cause large variations in the process variable. Since the process model is never accurate it is essential that the behavior of the closed-loop system is insensitive to variations in the process. The feedforward transfer function F is designed to give the desired response to set-point changes.

Fundamental Relations

The feedback loop is influenced by three external signals, the set point y_{sp}, the load disturbance d, and the measurement noise n. There are at least three signals x, y, and u that are of great interest for control. This means that there are nine relations between the input and the output signals. Since the system is linear these relations can be expressed in terms of the transfer functions. Let X, Y, U, D, N, and Y_{sp} be the Laplace transforms of x, y, u, d, n, and y_{sp}, respectively. The following relations are obtained from the block diagram in Figure 4.1:

$$
\begin{aligned}
X &= \frac{PCF}{1+PC}Y_{sp} + \frac{P}{1+PC}D - \frac{PC}{1+PC}N \\
Y &= \frac{PCF}{1+PC}Y_{sp} + \frac{P}{1+PC}D + \frac{1}{1+PC}N \\
U &= \frac{CF}{1+PC}Y_{sp} - \frac{PC}{1+PC}D - \frac{C}{1+PC}N.
\end{aligned}
\tag{4.2}
$$

There are several interesting conclusions we can draw from these equations. First, we can observe that several transfer functions are the same and that all relations are given by the following six transfer functions, called the *Gang of Six*.

$$
\begin{matrix}
\dfrac{PCF}{1+PC} & \dfrac{PC}{1+PC} & \dfrac{P}{1+PC} \\
\dfrac{CF}{1+PC} & \dfrac{C}{1+PC} & \dfrac{1}{1+PC}.
\end{matrix}
\tag{4.3}
$$

The transfer functions in the first column give the response of process variable and control signal to the set point. The second column gives the same signals in the case of pure error feedback when $F = 1$. The transfer function $P/(1+PC)$ in the third column tells how the process variable reacts to load disturbances, and the transfer function $C/(1+PC)$ gives the response of the control signal to measurement noise.

Notice that only four transfer functions,

$$
\begin{matrix}
\dfrac{PC}{1+PC} & \dfrac{P}{1+PC} \\
\dfrac{C}{1+PC} & \dfrac{1}{1+PC},
\end{matrix}
\tag{4.4}
$$

are required to describe how the system reacts to load disturbance and the measurement noise. These transfer functions are called the *Gang of Four*. They also capture robustness, as will be discussed in Section 4.6. Two additional transfer functions are required to describe how the system responds to set-point changes.

The special case when $F = 1$ is called a system with (pure) error feedback. In this case, all control actions are based on feedback from the error only. In this case, the system is completely characterized by the Gang of Four (4.4).

We are often interested in the magnitude of the transfer functions given by Equation 4.4. It is important to be aware that the transfer functions $PC/(1+PC)$ and $1/(1+PC)$ are dimension free, but the transfer functions $P/(1+PC)$ and $C/(1+PC)$ are not. For practical purposes it is therefore important to normalize the signals, for example, by scaling process inputs and outputs to the interval 0 to 1 or -1 to 1.

A Practical Consequence

The fact that six relations are required to capture properties of the basic feedback loop is often neglected in literature, particularly in the papers on PID control. To describe the system properly it is thus necessary to show the response of all six transfer functions. The transfer functions can be represented

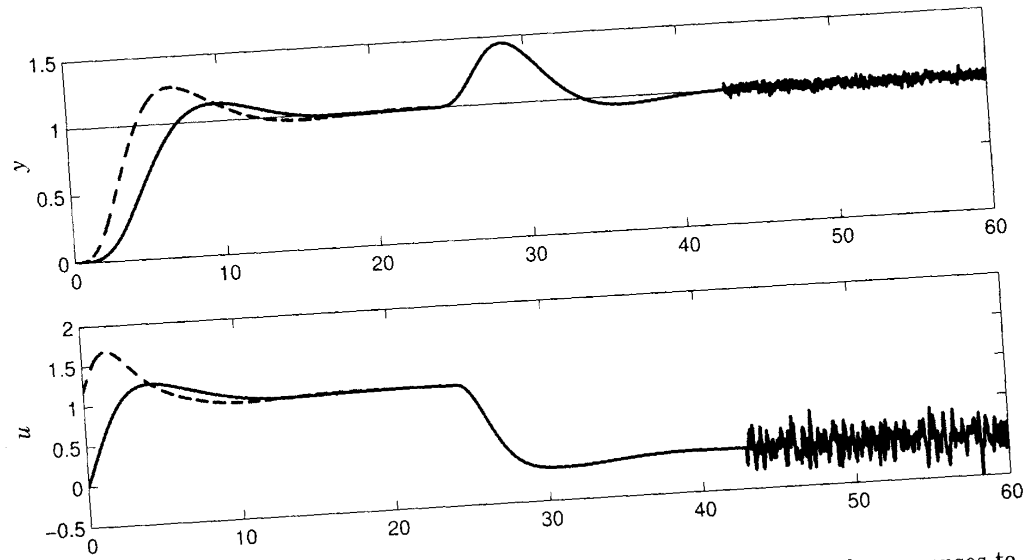


Figure 4.2 Representation of the properties of a basic feedback system by responses to a step in the reference, a step in the load disturbance, and measurement noise. The full lines are for set-point weight $b = 0$ and the dashed line is for set-point weight $b = 1$.

in different ways, by their step responses or by their frequency responses. Most papers on control only show the response of the process variable to set-point changes. Such a curve gives only partial information about the behavior of the system. To get a more complete representation of the system all six responses should be given, for example, as shown in Figure 4.2. This figure shows the responses in process variable and control signal to an experiment with a step change in set point followed by a step in the load disturbance, and measurement noise. The solid lines show the response when $F = 1$ and the dashed lines show the response when feedforward is used. Figure 4.2 thus gives a complete characterization of all six transfer functions in Equation 4.3.

Many Variations

The system shown in Figure 4.1 is a prototype problem. There are many variations of this problem. In Figure 4.1 the load disturbances act on the process input. In practice the disturbances can appear in many other places in the system. The measurement noise also acts at the process output. There may also be dynamics in the sensor, and the measured signal is often filtered. All these variations can be studied with minor modifications of the analysis based on Figure 4.1. As an illustration we will investigate the effects of a sensor filter. Figure 4.3 shows a block diagram of such a system. A typical example is a PID controller with set-point weighting and a second-order measurement filter. The transfer functions $F(s)$ and $C(s)$ in Figure 4.3 are given by (4.1) and the filter transfer function $G_f(s)$ is

$$G_f(s) = \frac{1}{1 + sT_f + s^2T_f^2/2}. \tag{4.5}$$

The relations between the input signals and output signals in Figure 4.3 are

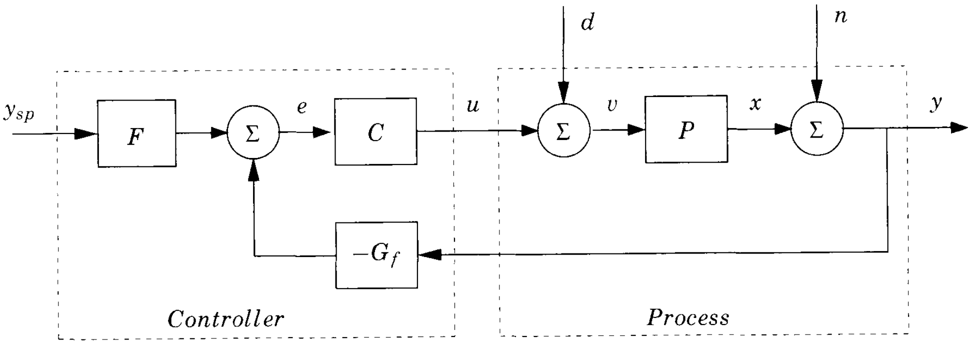


Figure 4.3 Block diagram of a basic feedback loop having two degrees of freedom and filtering of the measurement.

given by

$$
\begin{aligned}
X &= \frac{PCF}{1+PCG_f}Y_{sp} + \frac{P}{1+PCG_f}D - \frac{PCG_f}{1+PCG_f}N \\
Y &= \frac{PCF}{1+PCG_f}Y_{sp} + \frac{P}{1+PCG_f}D + \frac{1}{1+PCG_f}N \\
U &= \frac{CF}{1+PCG_f}Y_{sp} - \frac{PCG_f}{1+PCG_f}D - \frac{CG_f}{1+PCG_f}N.
\end{aligned}
\tag{4.6}
$$

Equation (4.6) is identical to (4.2) if the transfer function $C(s)$ and $F(s)$ are replaced by

$$
\bar{C}(s) = C(s)G_f(s), \quad \bar{F}(s) = \frac{F(s)}{G_f(s)}, \tag{4.7}
$$

The modifications required to deal with filtering are thus minor, and it suffices to develop the theory for the configuration in Figure 4.1.

Separation of Responses to Disturbances and Set Points

In early work on PID control it was a tradition to have two tuning rules, one for good set-point response and another for efficient attenuation of load disturbances. This practice still continues. A strong advantage of a controller with two degrees of freedom is that the responses to disturbances and set point can be designed separately. This follows from (4.2), which shows that the response to load disturbances and measurement noise is given by the $C(s)$, or from (4.6) by $\bar{C}(s) = C(s)G_f(s)$ when the measurement is filtered. A good design procedure is thus to determine $C(s)$ to account for robustness and disturbances. The feedforward transfer function $F(s)$ can then be chosen to give the desired set-point response. In general, this requires that the feedforward transfer function can be chosen freely. Simply choosing the set-point weights often give satisfactory results. Notice that there are some situations where only the error signal is available. The decoupling of the design problem then is not possible, and the design of the feedback then has to consider a trade-off between disturbances, robustness, and set-point response.