CHAPTER 1

Overview

This book discusses feedback systems and feedback control. A system is called a **feedback system** if it feeds back the knowledge of the system dynamical behavior for the control of the system. Feedback systems are everywhere; for example, a temperature control system such as an air conditioner is a feedback system since it measures the temperature to turn on or off the compressor motor. A human being is a very complex feedback system, who uses vision, touch, smell, etc., to coordinate many complex activities.

1.1 INTRODUCTION

Let us begin with an example. Figure 1.1 shows a car running along a straight road. Assume that the objective of the operation is to keep the car as close to the center line as possible. If there is no human or machine driving the car, it is likely to deviate further and further away from the center line, possibly because of imperfection of the car, imperfection of the road, wind gusts, or even earthquakes. Such a system is said to be unstable. One of the purposes of a driver is to make the system stable, i.e., to keep the car in the neighborhood of the center line. Here the car is the object of control, which is often called a **plant**. The driver plays the role of a **controller**. The task of making the combined system consisting of the plant and controller stable is called **stabilization**.

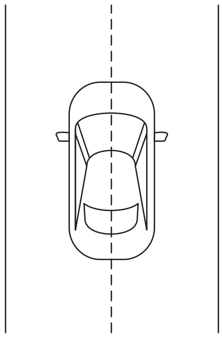

FIGURE 1.1: A car running in a straight road.

There are several conceivable schemes for achieving stabilization:

- **Open-loop control**: driving the car with eyes closed. This is clearly not going to work well, since the factors that affect the car position, such as wind, road conditions, etc., cannot be predicted. It is impossible to have a predetermined way to drive the car without observing the car position in real time. In general, one can never stabilize an unstable plant by open-loop control. It is possible to improve the performance of a stable plant by open-loop control, but even this is not very good.

- **Closed-loop control**: turning the wheel according to the deviation of the car position. Common sense indicates that as long as the wheels are turned in a proper way, the system can indeed be stabilized. Since the driver uses the information on the output to adjust the input, closed-loop control is also called **feedback control**. In general, feedback is essential to almost all control systems.

Automatic control is simply to use machines as controllers, to replace human beings.

Let us try to extract some essence of the plant from the car example we have just presented. First, a plant is a physical process that can be influenced from outside. The medium that can be used to influence the plant is called the **input**. In the example, the input is the wheel angle. The plant also produces some result, which is our concern. This result is called the **output**. In the example, the output is the deviation of the car from the center line. In addition to the input that can be manipulated, there are factors that influence the behavior of the plant but cannot be controlled, such as wind gust and road condition in the example. Such factors are called **disturbances**. In order to know and predict the behavior of a plant, and especially to know if the output is and will be satisfactory, we need to extract information from the plant. The information may or may not be the output itself, depending on the convenience and the technology available. Such information is called **measurement**. A plant can often be represented by a block diagram as shown in Figure 1.2. Here, u is the input, z is the output, d is the disturbance, and y is the measurement.

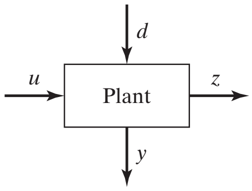

FIGURE 1.2: A plant.

If we take a closer look at a human feedback system, we can see that the human actually has to perform several subtasks. He/she has to observe the car position, then make some decisions regarding what actions to take, and finally exercise his/her decision and influence the motion of the car. Normally these three stages are called **sensing**, **control**, and **actuation**, respectively. If we are to replace a human driver by a machine, we need to build a sensor, a controller, and an actuator. A sensor measures a physical variable that can be used to deduce the behavior of the concerned output, such as the deviation of the car position, and turn it into a signal, usually an electrical signal, that the controller can read. The controller, often a computer or an electric circuit, takes the reading from the sensor, determines the action needed to correct the car position, and sends the decision to an actuator. The actuator then generates the quantity which influences the plant.

Therefore, a stabilizing closed-loop system can be represented by a block diagram as in Figure 1.3.

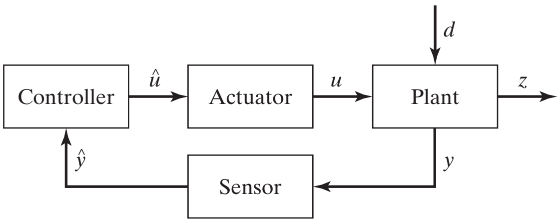

FIGURE 1.3: Structure of a feedback system for stabilization purpose.

Let us now look at another example in Figure 1.4. The purpose in this example is to have a car running along a hilly road against a persistent wind, so that its speed follows an external command, such as the maximum and minimum speed limits in certain segments of the road. There are two subtasks in this problem. The first is speed following even without the uphill or downhill slopes, or the head or tail wind. This problem is called **tracking**. The second is the reduction or complete elimination of the effect of the slopes and wind

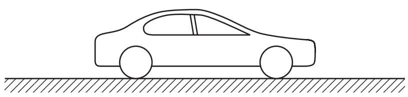

FIGURE 1.4: Speed control of a moving car.

on the car speed. This problem is called **disturbance rejection**. The overall problem, encompassing tracking and disturbance rejection, is called **regulation**. If the speed command is known to be a piecewise constant function, the tracking problem is called **set-point tracking**. Suppose that the acceleration of the car can be controlled by the gas pedal position. Again we have two possible schemes:

- **Open-loop control**: setting the gas pedal movement according to a computed position profile derived from the speed command, the road conditions in different segments of the road, and the wind speed obtained from an accurate weather forecast. One can imagine that the scheme will not work well since any error in the computed position profile, the road conditions, and the weather forecast will cause the speed to settle at the wrong value or not to settle at all because of the integration effect (the speed is proportional to the integral of the gas pedal position and hence small errors in the position may accumulate into large errors in speed).

- **Closed-loop control**: adjusting the gas pedal position according to the actual speed of the car. Since we can accelerate or decelerate the car in real time, according to the speed measurement, we should be able to control the speed of the car within a small neighborhood of the command as long as we take correct actions. Whether or not we know the road conditions or the wind speed is not important.

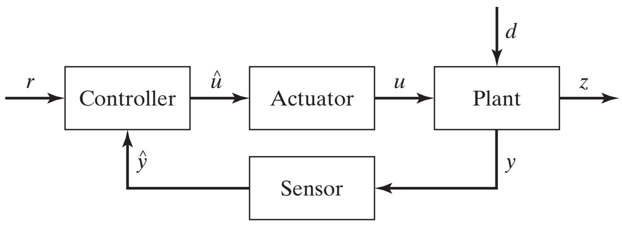

FIGURE 1.5: Structure of a feedback system for regulation purpose.

One can see that the major difference between regulation and stabilization is that there is a command signal, also called a **reference signal**, in the regulation situation. The controller needs to process this reference signal, in addition to other signals processed by the stabilizing controller. The structure of a feedback system for a regulation purpose is shown in Figure 1.5. Another difference between regulation and stabilization is that in the regulation problem, the disturbance is often assumed to be persistent and has some known features, such as being piecewise constant or piecewise sinusoids, whereas in the stabilization problem, the disturbance is assumed to be unknown and temporary in nature. The study of stabilization is important not only because there are genuine stabilization problems, such as suppressing vibration, balancing a pendulum, etc., but also because it is the key step in achieving regulation.

1.2 BASIC CONCEPTS

A **signal** $x(t)$ is a real-valued function of time. Conceptually, the time axis is the whole real axis $\mathbb{R}$ from $-\infty$ to ∞. Hence, a signal is a function from $\mathbb{R}$ to $\mathbb{R}$. However, we will mostly deal with unilateral or one-sided signals, i.e., signals $x(t)$ with $x(t) = 0$ for all $t < 0$. One such typical signal is the **unit step** signal, as shown in Figure 1.6. We use a special notation $\sigma(t)$ to denote the unit step signal

$$\sigma(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0. \end{cases}$$

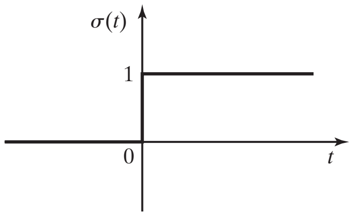

FIGURE 1.6: The unit step signal.

The unit step signal also comes handy in describing other unilateral signals, e.g., unilateral sinusoidal (function) signal $\sin \omega t\, \sigma(t)$ and unilateral exponential (function) signal $e^{\lambda t}\sigma(t)$. Another special signal that we often need to use is the so-called Dirac **unit impulse** signal $\delta(t)$, which is defined by the property:

$$\begin{cases} \delta(t) = 0, & t \neq 0 \\ \displaystyle\int_{-\infty}^{t} \delta(\tau) d\tau = \sigma(t), & \end{cases} \tag{1.1}$$

or by the property

$$\delta(t) = \frac{d\sigma(t)}{dt}. \tag{1.2}$$

The unit impulse function is not a function in the strict sense. It is a distribution and sometimes called a **singular function** or a **generalized function**. If we imagine a function as the distribution of mass in a long, straight string, then the unit impulse function means that there is a point mass of one unit at the middle of the string. This point mass occupies zero length in the string and hence has infinite density at the point, whereas the usual functions correspond to distributions with finite density. The unit impulse function $\delta(t)$ can be approximated by a narrow square pulse as shown in Figure 1.7(a). We also have a corresponding approximation of the unit step function as shown in Figure 1.7(b), which makes properties (1.1) and (1.2) consistent.

A **system** $\boldsymbol{S}$ is a map that transforms a signal vector

$$\boldsymbol{u}(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_m(t) \end{bmatrix},$$

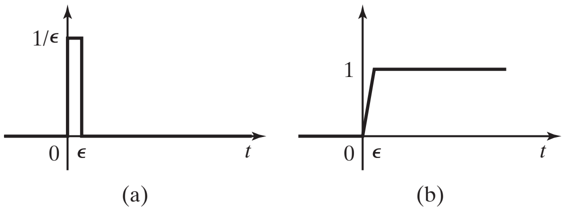

FIGURE 1.7: Approximate (a) unit impulse and (b) unit step.

called the **input**, to another signal vector

$$\boldsymbol{y}(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_p(t) \end{bmatrix},$$

called the **output**.

We write

$$\boldsymbol{y}(t) = \boldsymbol{S}[\boldsymbol{u}(t)] \quad \text{or} \quad \boldsymbol{y} = \boldsymbol{S}\boldsymbol{u}$$

to mean that $\boldsymbol{S}$ is a system with input $\boldsymbol{u}(t)$ and output $\boldsymbol{y}(t)$. We also use the block diagram as in Figure 1.8 to represent a system graphically. For example, plants, controllers, sensors, and actuators are all systems.

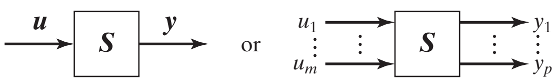

FIGURE 1.8: Graphic representation of a system.

A system is said to be a single-input–single-output (SISO) system if it has only one input and one output. If it has more than one input or more than one output, it is said to be a multi-input–single-output (MISO), single-input–multi-output (SIMO), or multi-input–multi-output (MIMO) system.

For technical reasons, we will mostly study SISO control systems in this book. Only occasionally will we deal with MISO and SIMO systems. The study of general MIMO systems requires more sophisticated tools and will be covered in more advanced books. For a clearer distinction, SISO systems will be denoted by normal upper case letters such as S, P, C, and MIMO (including MISO and SIMO) systems will be denoted by bold upper case letters such as $\boldsymbol{S}, \boldsymbol{P}, \boldsymbol{C}$. Similarly, scalar signals will be denoted by normal lower case letters such as x, u, y, and vector signals will be denoted by bold lower case letters such as $\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}$.

A system is said to be **static** or **memoryless** if the value of the output $y(t_0)$ at some time instance t_0 depends only on the value of the input $u(t_0)$ at t_0. Such systems are easy to deal with, but are very limited. We will be more interested in **dynamic systems** whose output $y(t_0)$ at time t_0 also depends on the value of the input $u(t)$ at other time instances $t \neq t_0$. One might feel awkward if the output $y(t_0)$ depends on a future input $u(t)$ with $t > t_0$. Indeed, this is not what we usually have. What we usually have are systems whose output at time t_0 depends only on the input $u(t)$ in the past or current time instances $t \leq t_0$. Such systems are called **causal systems**. All real-time physical systems are causal systems. In theoretical studies, we occasionally have to deal with noncausal systems. They are theoretical abstractions. From the applications point of view, they can only occur in non-real-time situations.

Finally in this section, let us pay attention to two very special MISO and SIMO systems that appear in almost all interconnected systems as will be seen throughout this book. They are too simple to be called systems. Instead, they are called a **summing point** and a **pickoff point**, respectively, as shown in Figure 1.9.

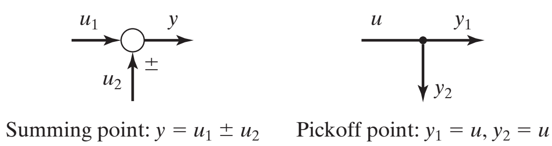

FIGURE 1.9: Summing point and pickoff point.

1.3 BASIC STRUCTURES OF FEEDBACK SYSTEMS

The purpose of this book is to study the design of a controller to satisfy given specifications in terms of stability and performance. As we have seen, there are two typical control tasks: stabilization and regulation. The control system structure for stabilization is given by Figure 1.3. However, the structure is not very convenient for a theory. First, there is no theory for the selection of actuators and sensors. Second, the effect of disturbance is very difficult to model. Third, in most of the applications, taking the measurement the same as the output has important advantages. For these reasons, we usually absorb the sensors and actuators into the plant, simplify the way the disturbance enters the system so that there is only an input disturbance and an output disturbance, and assume that the measurement and the output are the same. The general structure in Figure 1.3 then becomes a simpler yet more abstract structure as shown in Figure 1.10. It is this structure that we will use in our theory development. The stabilization problem then becomes the following mathematical problem: Given plant P, design controller C so that the system shown in Figure 1.10 has "good" stability.

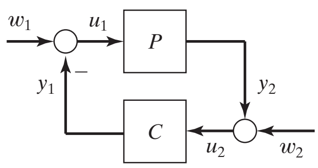

FIGURE 1.10: Feedback system for stabilization.

One may have noticed that in Figure 1.10 there is a minus sign attached to the signal y_1. This means that u_1 is w_1 minus y_1. This usage is mostly customary because the controller usually has a negative effect, in the sense that when y_2 is too big, the controller tries to reduce u_1, and vice versa. This type of feedback is called **negative feedback**. One may argue that if we replace C by $-C$, then the minus sign in Figure 1.10 can be removed. This is indeed a valid argument and is also an accepted practice. However, the tradition of keeping the negative sign there has strong reasons and we will follow this tradition throughout the book.

Similarly, in the regulation problem, we can also absorb the sensor and actuator to the plant and group the disturbances and noises into two groups: input disturbances and output noises. After doing these, Figure 1.5 becomes a simpler yet more abstract structure as shown in Figure 1.11. The regulation problem can be formulated into the following mathematical problem: Given plant P, design controller $\boldsymbol{C}$ so that good performance in tracking is achieved. Notice that the controller here is a MISO system. It has two inputs and one output. Such a controller is called a **two-degree-of-freedom (2DOF) controller**. Also, notice that there is a minus sign attached to the feedback signal y. This reflects the usual practice that the controller $\boldsymbol{C}$ generates the control signal u somehow from the difference between reference r and feedback y.

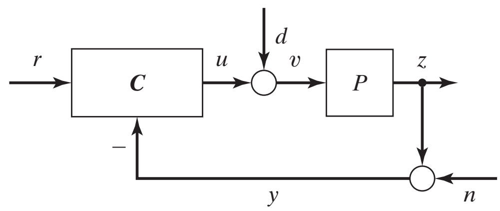

FIGURE 1.11: Feedback system for regulation.

In the early years of feedback control, the regulation problem was mostly solved by a more special feedback structure shown in Figure 1.12, which is called the **one-degree-of-freedom (1DOF) control** or **unity feedback**. In this structure, instead of taking the information of the reference and the measurement independently, the controller is driven by the difference between the reference and the measurement. Such a structure is simpler than the 2DOF structure since the controller is a SISO system and is actually a special case of the 2DOF structure by setting

$$\boldsymbol{C}\begin{bmatrix} r \\ -y \end{bmatrix} = C(r-y).$$

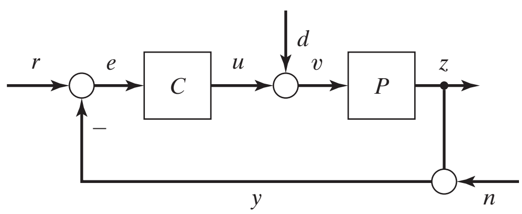

FIGURE 1.12: Unity feedback system.

This simple, or primitive, structure is still widely used today and is often considered as the default choice by many practitioners. However, as we will see in later parts of this book, the extra design freedom in 2DOF control over the 1DOF control provides significant advantages in achieving better performance and in trading off the often conflicting tracking and disturbance rejection requirements. One strong message of this book is that when we have a regulation problem, or even a pure tracking problem in the case when the disturbance is not present, the use of a 2DOF controller should be considered whenever possible.

1.4 ABOUT THIS BOOK

The main features of this book are as follows:

- It is a blend of classical (or even preclassical) and modern (or even postmodern) approaches.

  Most control textbooks treat classical control theory and modern control theory separately, with the implied message that the classical control theory is more elementary and more accessible to beginners, while the modern control theory is more advanced and more sophisticated. One common division of the classical and modern approaches is that the classical approach is based on transfer functions, whereas the modern approach is based on state space descriptions. Another division of the classical and modern approaches is based on the time line: everything developed before the 1950s is classical, everything developed after the 1950s is modern, and those developed during the 1950s are in the gray zone. We believe that all these separations, divisions, and segregations only do more harm than good.

  In this book, we try to break the fine line between the classical and modern approaches, and integrate control theory development in different stages into a unified theory for SISO system analysis and design. As in classical control, we mainly use the transfer function as a system model, and try to design simple controllers using intuitive techniques. As in modern control, we emphasize quantitative analysis and analytical design, and try to design optimal controllers and understand fundamental design limitations, i.e., what feedback control can or cannot do. We attempt not to sacrifice mathematical rigor and attempt to make connections to computer-aided analysis and design.

- The use of 2DOF controllers in regulation problems is emphasized.

  2DOF controllers are not new. They appeared in the earlier days when feedback control first became a widely used practice. Many *ad hoc* control schemes in animals and machines are 2DOF by nature.

  However, history took a sharp turn when many popular textbooks and other publications defined a feedback control system as "a system that maintains a prescribed relationship between the output and some reference input by comparing them and using the difference as a means of control" (Ogata, 2008), or a system in which "the controlled signal $c(t)$ should be fed back and compared with the reference input, and an actuating signal proportional to the

> difference of the input and the output must be sent through the system to correct the error" (Kuo and Golnaraghi, 2002).
>
> Such a definition, though covering a good number of situations when it is indeed the difference of the reference and the outcome that is used in the decision-making process of the controller, misses many other practices where the controllers process the reference and outcome independently.
>
> A more accurate definition was given in Dorf and Bishop (2008): "A feedback control system is a control system that tends to maintain a prescribed relationship of one system variable to another by comparing functions of these variables and using the difference as a means of control." The difference between this definition and the one in Ogata (2008) is the word "functions". In general, the function of the reference and that of the output are allowed to be completely different, giving rise to two complete degrees of freedom.

In this book, we take the unprecedented step in introducing the notation $\sigma(t)$ to denote the unit step function. The unit step function has wide applications in diverse disciplines. However, it has not had a standard notation as its sister, the unit impulse signal $\delta(t)$, does. It has sometimes been denoted by $u(t)$, $\mathbf{1}(t)$, or other variations. The unit step is the integral of the unit impulse and the unit impulse is the differentiation of the unit step; as such, considering the meanings of Greek letters Σ and Δ as sum and difference, respectively, we now simply have the Σ of $\delta(t)$ as $\sigma(t)$ and the Δ of $\sigma(t)$ as $\delta(t)$.

We attempt to make this book self-contained so that it not only tells us "how" but also "why". We do not share the argument that in undergraduate textbooks rigorous reasonings should give way to intuitions and design recipes. We strongly believe that learning a few "whys" is better than learning many "hows". In this book, we provide as much as possible, the reasons and justifications behind theorems, design procedures, algorithms, etc.

Such reasons and justifications might be difficult to digest for an average reader, but these insights definitely provide a source of information for instructors and students with an interest in in-depth exploration. We tint the parts of the text containing sophisticated mathematical reasoning to indicate that these parts may be skipped without affecting the basic understanding of the book.

While we pay attention to the theoretic soundness of the theory, we also pay attention to the illustrative examples, which are usually quite simple, yet informative, and to case studies, which are nontrivial but are commonly accessible in undergraduate laboratories. We strongly suggest that the use of this book be accompanied by control experiments of real physical systems, starting with system modeling, going through controller design, analysis, and redesign, and ending with controller implementation and hardware-in-the-loop simulations.

In this textbook, computer-aided analysis and design are integrated into the presentation of theory and examples. MATLAB, together with SIMULINK, is used as the programming platform. Analysis and design procedures are stated in the form of algorithms so that they can be programmed easily. Some of the exercise problems are specifically labeled as MATLAB problems to give ample opportunities for students to practice their MATLAB program skills and to strengthen their understanding of theory by translating the algorithms into programs. If all MATLAB problems in the book are completed, enough number of programs can be generated to form a MATLAB toolbox of SISO system analysis and design.

Finally, in this textbook, we attempt to create opportunities to motivate the student into further exploration of the deeper and wider space of feedback control theory. We do this by pointing out the insufficiencies of the content of this book, by referring to sources in the literature for materials beyond the coverage of this book, and by giving several extra credit problems which require a fair amount of extra reading and thinking.

PROBLEMS

1.1. Give several examples of stabilization control.

1.2. Give several examples of regulation control.

1.3. Suppose that you are a meticulously lawful driver who always follows the speed limit closely, never overspeed and seldom underspeed. Do you think that you are a unity feedback controller or a 2DOF controller?

1.4. Find the derivatives of unilateral functions $(\sin\ \omega t)\sigma(t)$, $(\cos\ \omega t)\sigma(t)$, and $e^{\lambda t}\sigma(t)$.

NOTES AND REFERENCES

A prerequisite of this course is a basic knowledge on signals and systems. An excellent textbook on this is

> A. V. Oppenheim and A. S. Willsky with S. H. Newab, *Signals and Systems*, 2nd Edition, Prentice Hall, Upper Saddle River, New Jersey, 1996.

Other good references are

> H. Kwakernaak and R. Sivan, *Modern Signals and Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.

> E. A. Lee and P. Varaiya, *Structure and Interpretation of Signals and Systems*, Addison Wesley, Boston, 2003.

There are several commonly used textbooks on control systems, some essentially written over 30 years ago, such as

> R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 11th Edition, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2008.

> B. C. Kuo and F. Golnaraghi, *Automatic Control Systems*, 8th Edition, Prentice Hall, Englewood Cliffs, New Jersey, 2002.

> K. Ogata, *Modern Control Engineering*, 5th Edition, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2008.

and some are more recent, such as

G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 5th Edition, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2006.

G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control Systems Design*, Prentice Hall, Upper Saddle River, New Jersey, 2001.

An early book introducing 2DOF controllers is

I. Horowitz, *Synthesis of Feedback Systems*, Academic Press, London, 1963.

A recent textbook advocating the use of 2DOF controllers is

W. A. Wolovich, *Automatic Control Systems*, Sauders College Publishing, Fort Worth, 1994.

Two textbooks with the aim of integrating classical and modern approaches are

J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*, Macmillan Publishing Company, New York, 1992.

J. W. Helton and O. Merino, *Classical Control Using H^∞ Methods: Theory, Optimization, and Design*, SIAM, Philadelphia, 1998.