1
Introduction

1.1 What is Control?

When we use the word *control* in everyday life, we are referring to the act of producing a desired result. By this broad definition, control is seen to cover all artificial processes. The temperature inside a refrigerator is controlled by a thermostat. The picture we see on the television is a result of a controlled beam of electrons made to scan the television screen in a selected pattern. A compact-disc player focuses a fine laser beam at the desired spot on the rotating compact-disc in order to produce the desired music. While driving a car, the driver is controlling the speed and direction of the car so as to reach the destination quickly, without hitting anything on the way. The list is endless. Whether the control is automatic (such as in the refrigerator, television or compact-disc player), or caused by a human being (such as the car driver), it is an integral part of our daily existence. However, control is not confined to artificial processes alone. Imagine living in a world where the temperature is unbearably hot (or cold), without the life-supporting oxygen, water or sunlight. We often do not realize how controlled the natural environment we live in is. The composition, temperature and pressure of the earth's atmosphere are kept stable in their livable state by an intricate set of natural processes. The daily variation of temperature caused by the sun controls the metabolism of all living organisms. Even the simplest life form is sustained by unimaginably complex chemical processes. The ultimate control system is the human body, where the controlling mechanism is so complex that even while sleeping, the brain regulates the heartbeat, body temperature and blood-pressure by countless chemical and electrical impulses per second, in a way not quite understood yet. (You have to wonder who designed *that* control system!) Hence, control is everywhere we look, and is crucial for the existence of life itself.

A study of control involves developing a mathematical model for each component of the control system. We have twice used the word *system* without defining it. A system is a set of self-contained processes under study. A *control system* by definition consists of the system to be controlled – called the *plant* – as well as the system which exercises control over the plant, called the *controller*. A controller could be either human, or an artificial device. The controller is said to supply a signal to the plant, called the *input to the plant* (or the *control input*), in order to produce a desired response from the plant, called the *output from the plant*. When referring to an isolated system, the terms *input* and *output* are used to describe the signal that goes into a system, and the signal that comes out of a system, respectively. Let us take the example of the control system consisting of a car and its driver. If we select the car to be the plant, then the driver becomes the controller, who applies an input to the plant in the form of pressing the gas pedal if it is desired to increase the speed of the car. The speed increase can then be the output from the plant. Note that in a control system, what control input can be applied to the plant is determined by the physical processes of the plant (in this case, the car's engine), but the output could be anything that can be directly measured (such as the car's speed or its position). In other words, many different choices of the output can be available at the same time, and the controller can use any number of them, depending upon the application. Say if the driver wants to make sure she is obeying the highway speed limit, she will be focusing on the speedometer. Hence, the speed becomes the plant output. If she wants to stop well before a stop sign, the car's position with respect to the stop sign becomes the plant output. If the driver is overtaking a truck on the highway, both the speed and the position of the car *vis-á-vis* the truck are the plant outputs. Since the plant output is the same as the output of the control system, it is simply called the *output* when referring to the control system as a whole. After understanding the basic terminology of the control system, let us now move on to see what different varieties of control systems there are.

1.2 Open-Loop and Closed-Loop Control Systems

Let us return to the example of the car driver control system. We have encountered the not so rare breed of drivers who generally boast of their driving skills with the following words: "Oh I am so good that I can drive this car with my eyes closed!" Let us imagine we give such a driver an opportunity to live up to that boast (without riding with her, of course) and apply a blindfold. Now ask the driver to accelerate to a particular speed (assuming that she continues driving in a straight line). While driving in this fashion, the driver has absolutely no idea about what her actual speed is. By pressing the gas pedal (control input) she hopes that the car's speed will come up to the desired value, but has no means of verifying the actual increase in speed. Such a control system, in which the control input is applied without the knowledge of the plant output, is called an *open-loop control system*. Figure 1.1 shows a *block-diagram* of an open-loop control system, where the sub-systems (controller and plant) are shown as rectangular blocks, with arrows indicating input and output to each block. By now it must be clear that an open-loop controller is like a rifle shooter who gets only one shot at the target. Hence, open-loop control will be successful only if the controller has a pretty good prior knowledge of the *behavior* of the plant, which can be defined as the relationship between the control input

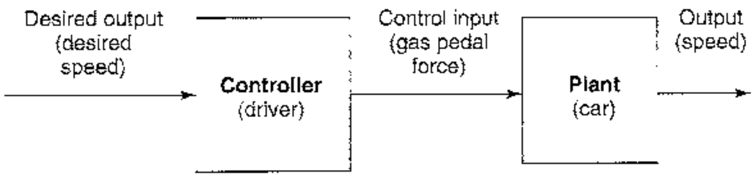

Figure 1.1 An open-loop control system: the controller applies the control input without knowing the plant output

and the plant output. If one knows what output a system will produce when a known input is applied to it, one is said to know the system's behavior.

Mathematically, the relationship between the output of a *linear* plant and the control input (the system's behavior) can be described by a *transfer function* (the concepts of linear systems and transfer functions are explained in Chapter 2). Suppose the driver knows from previous driving experience that, to maintain a speed of 50 kilometers per hour, she needs to apply one kilogram of force on the gas pedal. Then the car's transfer function is said to be 50 km/hr/kg. (This is a very simplified example. *The actual car is not going to have such a simple transfer function.*) Now, if the driver can accurately control the force exerted on the gas pedal, she can be quite confident of achieving her target speed, even though blindfolded. However, as anybody reasonably experienced with driving knows, there are many uncertainties – such as the condition of the road, tyre pressure, the condition of the engine, or even the uncertainty in gas pedal force actually being applied by the driver – which can cause a change in the car's behavior. If the transfer function in the driver's mind was determined on smooth roads, with properly inflated tyres and a well maintained engine, she is going to get a speed of less than 50 km/hr with 1 kg force on the gas pedal if, say, the road she is driving on happens to have rough patches. In addition, if a wind happens to be blowing opposite to the car's direction of motion, a further change in the car's behavior will be produced. Such an unknown and undesirable input to the plant, such as road roughness or the head-wind, is called a *noise*. In the presence of uncertainty about the plant's behavior, or due to a noise (or both), it is clear from the above example that an open-loop control system is unlikely to be successful.

Suppose the driver decides to drive the car like a sane person (i.e. with both eyes wide open). Now she can see her actual speed, as measured by the speedometer. In this situation, the driver can adjust the force she applies to the pedal so as to get the desired speed on the speedometer; it may not be a one shot approach, and some trial and error might be required, causing the speed to initially overshoot or undershoot the desired value. However, after some time (depending on the ability of the driver), the target speed can be achieved (if it is within the capability of the car), irrespective of the condition of the road or the presence of a wind. Note that now the driver – instead of applying a pre-determined control input as in the open-loop case – is adjusting the control input according to the actual observed output. Such a control system in which the control input is a function of the plant's output is called a *closed-loop system*. Since in a closed-loop system the controller is constantly in touch with the actual output, it is likely to succeed in achieving the desired output even in the presence of noise and/or uncertainty in the linear plant's behavior (transfer-function). The mechanism by which the information about the actual output is conveyed to the controller is called *feedback*. On a block-diagram, the path from the plant output to the controller input is called a *feedback-loop*. A block-diagram example of a possible closed-loop system is given in Figure 1.2.

Comparing Figures 1.1 and 1.2, we find a new element in Figure 1.2 denoted by a circle before the controller block, into which two arrows are leading and out of which one arrow is emerging and leading to the controller. This circle is called a *summing junction*, which adds the signals leading into it with the appropriate signs which are indicated adjacent to the respective arrowheads. If a sign is omitted, a positive sign is assumed. The output of

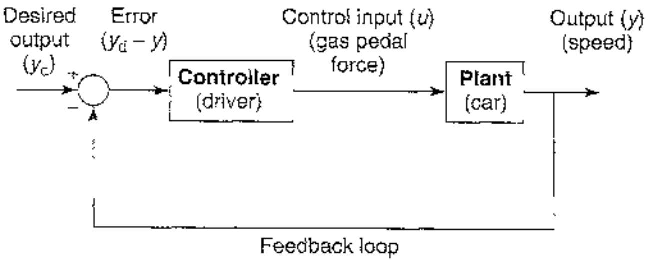

Figure 1.2 Example of a closed-loop control system with feedback; the controller applies a control input based on the plant output

the summing junction is the arithmetic sum of its two (or more) inputs. Using the symbols u (control input), y (output), and y_d (desired output), we can see in Figure 1.2 that the *input* to the *controller* is the error signal $(y_d - y)$. In Figure 1.2, the controller itself is a system which produces an *output* (control input), u, based upon the input it receives in the form of $(y_d - y)$. Hence, the behavior of a *linear* controller could be mathematically described by its transfer-function, which is the relationship between u and $(y_d - y)$. Note that Figure 1.2 shows only a popular kind of closed-loop system. In other closed-loop systems, the input to the controller could be different from the error signal $(y_d - y)$. The controller transfer-function is the main design parameter in the design of a control system and determines how rapidly – and with what *maximum overshoot* (i.e. maximum value of $|y_d - y|$) – the actual output, y, will become equal to the desired output, y_d. We will see later how the controller transfer-function can be obtained, given a set of design requirements. (However, deriving the transfer-function of a human controller is beyond the present science, as mentioned in the previous section.) When the desired output, y_d, is a constant, the resulting controller is called a *regulator*. If the desired output is changing with time, the corresponding control system is called a *tracking system*. In any case, the principal task of a closed-loop controller is to make $(y_d - y) = 0$ as quickly as possible. Figure 1.3 shows a possible plot of the actual output of a closed-loop control system.

Whereas the desired output y_d has been achieved after some time in Figure 1.3, there is a large maximum overshoot which could be unacceptable. A successful closed-loop controller design should achieve both a small maximum overshoot, and a small error magnitude $|y_d - y|$ as quickly as possible. In Chapter 4 we will see that the output of a linear system to an arbitrary input consists of a fluctuating sort of response (called the *transient response*), which begins as soon as the input is applied, and a settled kind of response (called the *steady-state response*) after a long time has elapsed since the input was initially applied. If the linear system is *stable*, the transient response would decay to zero after sometime (*stability* is an important property of a system, and is discussed in Section 2.8), and only the steady-state response would persist for a long time. The transient response of a linear system depends *largely* upon the characteristics and the initial *state* of the system, while the steady-state response depends both upon system's characteristics and the input as a function of time, i.e. $u(t)$. The maximum overshoot is a property of the transient response, but the error magnitude $|y_d - y|$ at large time (or in the limit $t \to \infty$) is a property of the steady-state response of the closed-loop system. In

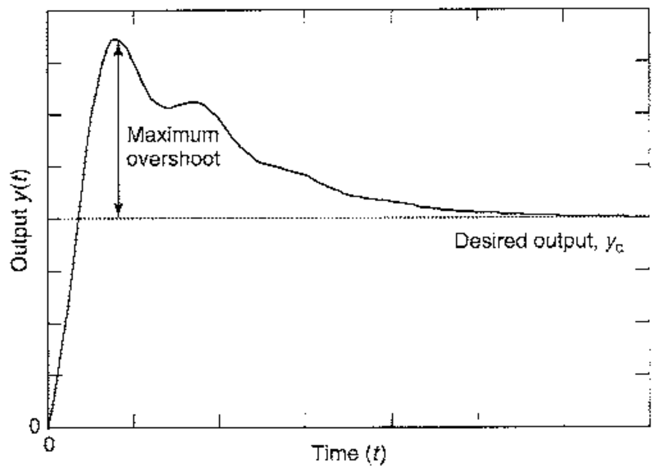

Figure 1.3 Example of a closed-loop control system's response; the desired output is achieved after some time, but there is a large maximum overshoot

Figure 1.3 the steady-state response asymptotically approaches a constant y_d in the limit $t \to \infty$.

Figure 1.3 shows the basic fact that it is impossible to get the desired output *immediately*. The reason why the output of a linear, stable system does not *instantaneously* settle to its steady-state has to do with the inherent physical characteristics of all practical systems that involve either *dissipation* or *storage* of *energy* supplied by the input. Examples of energy storage devices are a spring in a mechanical system, and a capacitor in an electrical system. Examples of energy dissipation processes are mechanical friction, heat transfer, and electrical resistance. Due to a transfer of energy from the applied input to the energy storage or dissipation elements, there is initially a fluctuation of the total energy of the system, which results in the transient response. As the time passes, the energy contribution of storage/dissipative processes in a stable system declines rapidly, and the total energy (hence, the output) of the system tends to the same function of time as that of the applied input. To better understand this behavior of linear, stable systems, consider a bucket with a small hole in its bottom as the system. The input is the flow rate of water supplied to the bucket, which could be a specific function of time, and the output is the *total* flow rate of water coming out of the bucket (from the hole, as well as from the overflowing top). Initially, the bucket takes some time to fill due to the hole (dissipative process) and its internal volume (storage device). However, after the bucket is full, the output largely follows the changing input.

While the most common closed-loop control system is the *feedback control system*, as shown in Figure 1.2, there are other possibilities such as the *feedforward control system*. In a feedforward control system – whose example is shown in Figure 1.4 – in addition to a feedback loop, a feedforward path from the desired output (y_d) to the control input is generally employed to counteract the effect of noise, or to reduce a known undesirable plant behavior. The feedforward controller incorporates some *a priori* knowledge of the plant's behavior, thereby reducing the burden on the feedback controller in controlling

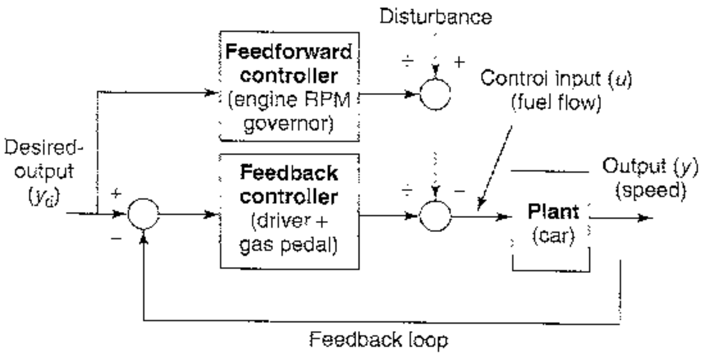

Figure 1.4 A closed-loop control system with a feedforward path; the engine RPM governor takes care of the fuel flow disturbance, leaving the driver free to concentrate on achieving desired speed with gas pedal force

the plant. Note that if the feedback controller is removed from Figure 1.4, the resulting control system becomes open-loop type. Hence, a feedforward control system can be regarded as a *hybrid* of open and closed-loop control systems. In the car driver example, the feedforward controller could be an engine rotational speed governor that keeps the engine's RPM constant in the presence of disturbance (noise) in the fuel flow rate caused by known imperfections in the fuel supply system. This reduces the burden on the driver, who would have been required to apply a rapidly changing gas pedal force to counteract the fuel supply disturbance if there was no feedforward controller. Now the feedback controller consists of the driver and the gas-pedal mechanism, and the control input is the fuel flow into the engine, which is influenced by not only the gas-pedal force, but also by the RPM governor output and the disturbance. It is clear from the present example that many practical control systems can benefit from the feedforward arrangement.

In this section, we have seen that a control system can be classified as either open- or closed-loop, depending upon the physical arrangement of its components. However, there are other ways of classifying control systems, as discussed in the next section.

1.3 Other Classifications of Control Systems

Apart from being open- or closed-loop, a control system can be classified according to the physical nature of the laws obeyed by the system, and the mathematical nature of the governing differential equations. To understand such classifications, we must define the *state* of a system, which is the fundamental concept in modern control. The *state* of a system is any set of physical quantities which need to be specified at a given time in order to completely determine the behavior of the system. This definition is a little confusing, because it introduces another word, *determine*, which needs further explanation given in the following paragraph. We will return to the concept of state in Chapter 3, but here let us only say that the state is all the information we need about a system to tell what the system is doing at any given time. For example, if one is given information about the speed of a car and the positions of other vehicles on the road relative to the car, then one has sufficient information to drive the car safely. Thus, the state of such a system consists of the car's speed and relative positions of other vehicles. However, for the same system one could choose another set of physical quantities to be the system's state, such as velocities of all other vehicles relative to the car, and the position of the car with respect to the road divider. Hence, by definition the state is not a unique set of physical quantities.

A control system is said to be *deterministic* when the set of physical laws governing the system are such that if the state of the system at some time (called the *initial conditions*) and the input are specified, then one can precisely predict the state at a later time. The laws governing a deterministic system are called *deterministic laws*. Since the characteristics of a deterministic system can be found merely by studying its response to initial conditions (transient response), we often study such systems by taking the applied input to be zero. A response to initial conditions when the applied input is zero depicts how the system's state *evolves* from some initial time to that at a later time. Obviously, the *evolution* of only a deterministic system can be determined. Going back to the definition of state, it is clear that the latter is arrived at keeping a deterministic system in mind, but the concept of state can also be used to describe systems that are *not* deterministic. A system that is not deterministic is either *stochastic*, or has *no* laws governing it. A *stochastic* (also called *probabilistic*) system has such governing laws that although the initial conditions (i.e. state of a system at some time) are known in every detail, it is impossible to determine the system's state at a later time. In other words, based upon the *stochastic* governing laws and the initial conditions, one could only determine the probability of a state, rather than the state itself. When we toss a perfect coin, we are dealing with a stochastic law that states that both the possible outcomes of the toss (head or tail) have an equal probability of 50 percent. We should, however, make a distinction between a physically stochastic system, and our *ability* (as humans) to *predict* the behavior of a deterministic system based upon our measurement of the initial conditions and our understanding of the governing laws. Due to an uncertainty in our knowledge of the governing deterministic laws, as well as errors in measuring the initial conditions, we will frequently be unable to predict the state of a deterministic system at a later time. Such a problem of unpredictability is highlighted by a special class of deterministic systems, namely *chaotic* systems. A system is called *chaotic* if even a small change in the initial conditions produces an arbitrarily large change in the system's state at a later time.

An example of chaotic control systems is a *double pendulum* (Figure 1.5). It consists of two masses, m_1 and m_2, joined together and suspended from point O by two rigid massless links of lengths L_1 and L_2 as shown. Here, the state of the system can be defined by the angular displacements of the two links, $\theta_1(t)$ and $\theta_2(t)$, as well as their respective angular velocities, $\theta_1^{(1)}(t)$ and $\theta_2^{(1)}(t)$. (In this book, the notation used for representing a kth order *time derivative* of $f(t)$ is $f^{(k)}(t)$, i.e. $d^k f(t)/dt^k = f^{(k)}(t)$. Thus, $\theta_1^{(1)}(t)$ denotes $d\theta_1(t)/dt$, etc.) Suppose we do not apply an input to the system, and begin observing the system at some time, $t = 0$, at which the initial conditions are, say, $\theta_1(0) = 40°$, $\theta_2(0) = 80°$, $\theta_1^{(1)}(0) = 0°/\text{s}$, and $\theta_2^{(1)}(0) = 0°/\text{s}$. Then at a later time, say after 100 s, the system's state will be very much different from what it would have been if the initial conditions were, say, $\theta_1(0) = 40.01°$, $\theta_2(0) = 80°$, $\theta_1^{(1)}(0) = 0°/\text{s}$, and $\theta_2^{(1)}(0) = 0°/\text{s}$. Figure 1.6 shows the time history of the angle $\theta_2(t)$ between 85 s and 100 s

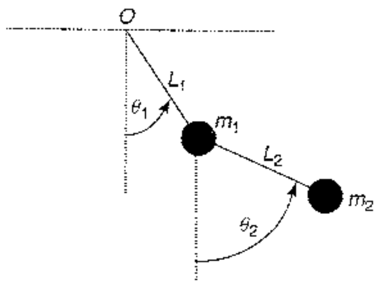

Figure 1.5 A double pendulum is a chaotic system because a small change in its initial conditions produces an arbitrarily large change in the system's state after some time

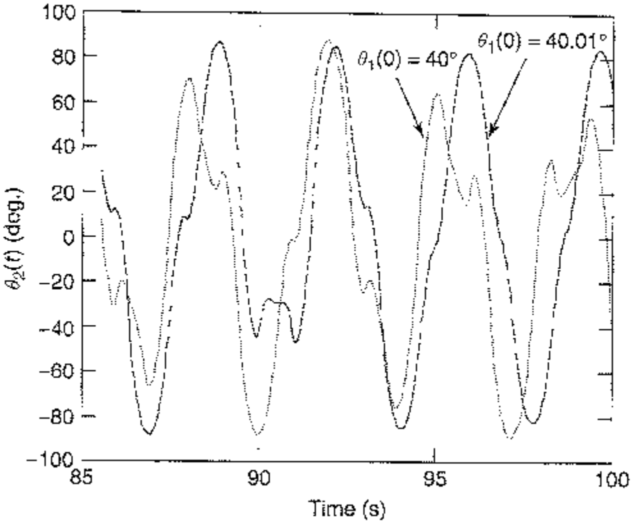

Figure 1.6 Time history between 85 s and 100 s of angle θ_2 of a double pendulum with $m_1 = 1$ kg, $m_2 = 2$ kg, $L_1 = 1$ m, and $L_2 = 2$ m for the two sets of initial conditions $\theta_1(0) = 40°$, $\theta_2(0) = 80°$, $\theta_1^{(1)}(0) = 0°/s$, $\theta_2^{(1)}(0) = 0°/s$ and $\theta_1(0) = 40.01°$, $\theta_2(0) = 80°$, $\theta_1^{(1)}(0) = 0°/s$, $\theta_2^{(1)}(0) = 0°/s$, respectively

for the two sets of initial conditions, for a double pendulum with $m_1 = 1$ kg, $m_2 = 2$ kg, $L_1 = 1$ m, and $L_2 = 2$ m. Note that we know the governing laws of this deterministic system, yet we cannot predict its state after a given time, because there will always be some error in measuring the initial conditions. Chaotic systems are so interesting that they have become the subject of specialization at many physics and engineering departments.

Any unpredictable system can be mistaken to be a stochastic system. Taking the car driver example of Section 1.2, there may exist deterministic laws that govern the road conditions, wind velocity, etc., but our ignorance about them causes us to treat such phenomena as random noise, i.e. stochastic processes. Another situation when a deterministic system may appear to be stochastic is exemplified by the toss of a coin deliberately loaded to fall every time on one particular side (either head or tail). An unwary spectator may believe such a system to be stochastic, when actually it is very much deterministic!

When we analyze and design control systems, we try to express their governing physical laws by differential equations. The mathematical nature of the governing differential equations provides another way of classifying control systems. Here we depart from the realm of physics, and delve into mathematics. Depending upon whether the differential equations used to describe a control system are linear or nonlinear in nature, we can call the system either *linear* or *nonlinear*. Furthermore, a control system whose description requires partial differential equations is called a *distributed parameter system*, whereas a system requiring only ordinary differential equations is called a *lumped parameter system*. A vibrating string, or a membrane is a distributed parameter system, because its properties (mass and stiffness) are distributed in space. A mass suspended by a spring is a lumped parameter system, because its mass and stiffness are concentrated at discrete points in space. (A more common nomenclature of distributed and lumped parameter systems is *continuous* and *discrete* systems, respectively, but we avoid this terminology in this book as it might be confused with *continuous time* and *discrete time* systems.) A particular system can be treated as linear, or nonlinear, distributed, or lumped parameter, depending upon what aspects of its behavior we are interested in. For example, if we want to study only small angular displacements of a simple pendulum, its differential equation of motion can be treated to be linear; but if large angular displacements are to be studied, the same pendulum is treated as a nonlinear system. Similarly, when we are interested in the motion of a car as a whole, its state can be described by only two quantities: the position and the velocity of the car. Hence, it can be treated as a lumped parameter system whose entire mass is concentrated at one point (the center of mass). However, if we want to take into account how the tyres of the car are deforming as it moves along an uneven road, the car becomes a distributed parameter system whose state is described exactly by an infinite set of quantities (such as deformations of all the points on the tyres, and their time derivatives, in addition to the speed and position of the car). Other classifications based upon the mathematical nature of governing differential equations will be discussed in Chapter 2.

Yet another way of classifying control systems is whether their *outputs* are *continuous* or *discontinuous* in time. If one can express the system's state (which is obtained by solving the system's differential equations) as a continuous function of time, the system is called *continuous in time* (or *analog system*). However, a majority of modern control systems produce outputs that 'jump' (or are discontinuous) in time. Such control systems are called *discrete in time* (or *digital systems*). Note that in the limit of very small time steps, a digital system can be approximated as an analog system. In this book, we will make this assumption quite often. If the time steps chosen to sample the discontinuous output are relatively large, then a digital system can have a significantly different behaviour from that of a corresponding analog system. In modern applications, even analog controllers are implemented on a digital processor, which can introduce digital characteristics to the control system. Chapter 8 is devoted to the study of digital systems.

There are other minor classifications of control systems based upon the systems' characteristics, such as *stability*, *controllability*, *observability*, etc., which we will take up in subsequent chapters. Frequently, control systems are also classified based upon the number of inputs and outputs of the system, such as *single-input, single-output* system, or *two-input, three-output* system, etc. In *classical* control (an object of Chapter 2) the distinction between *single-input, single-output* (SISO) and *multi-input, multi-output* (MIMO) systems is crucial.

1.4 On the Road to Control System Analysis and Design

When we find an unidentified object on the street, the first thing we may do is prod or poke it with a stick, pick it up and shake it, or even hit it with a hammer and hear the sound it makes, in order to find out something about it. We treat an unknown control system in a similar fashion, i.e. we apply some well known inputs to it and carefully observe how it responds to those inputs. This has been an age old method of analyzing a system. Some of the well known inputs applied to study a system are the *singularity functions*, thus called due to their peculiar nature of being *singular* in the mathematical sense (their time derivative tends to infinity at some time). Two prominent members of this zoo are the *unit step function* and the *unit impulse function*. In Chapter 2, useful computer programs are presented to enable you to find the response to *impulse* and *step* inputs – as well as the response to an *arbitrary* input – of a single-input, single-output control system. Chapter 2 also discusses important properties of a control system, namely, *performance*, *stability*, and *robustness*, and presents the analysis and design of linear control systems using the *classical approach* of *frequency response*, and *transform methods*. Chapter 3 introduces the *state-space* modeling for linear control systems, covering various applications from all walks of engineering. The solution of a linear system's governing equations using the state-space method is discussed in Chapter 4. In this chapter, many new computer programs are presented to help you solve the state-equations for linear or nonlinear systems.

The design of modern control systems using the state-space approach is introduced in Chapter 5, which also discusses two important properties of a plant, namely its *controllability* and *observability*. In this chapter, it is first assumed that all the quantities defining the state of a plant (called *state variables*) are available for exact measurement. However, this assumption is not always practical, since some of the state variables may not be measurable. Hence, we need a procedure for estimating the unmeasurable state variables from the information provided by those variables that we can measure. Later sections of Chapter 5 contains material about how this process of *state estimation* is carried out by an *observer*, and how such an estimation can be incorporated into the control system in the form of a compensator. Chapter 6 introduces the procedure of designing an *optimal control system*, which means a control system meeting all the design requirements in the most efficient manner. Chapter 6 also provides new computer programs for solving important optimal control problems. Chapter 7 introduces the treatment of random signals generated by stochastic systems, and extends the philosophy of state estimation to plants with noise, which is treated as a random signal. Here we also learn how an *optimal* state estimation can be carried out, and how a control system can be made *robust* with respect to *measurement and process noise*. Chapter 8 presents the design and analysis of *digital control systems* (also called *discrete time systems*), and covers many modern digital control applications. Finally, Chapter 9 introduces various advanced topics in modern control, such as *advanced robust control techniques, nonlinear control*, etc. Some of the topics contained in Chapter 9, such as *input shaping control* and *rate-weighted optimal control*, are representative of the latest control techniques.

At the end of each chapter (except Chapter 1), you will find exercises that help you grasp the essential concepts presented in the chapter. These exercises range from analytical to numerical, and are designed to make you think, rather than apply ready-made formulas for their solution. At the end of the book, answers to some numerical exercises are provided to let you check the accuracy of your solutions.

1.5 MATLAB, SIMULINK, and the Control System Toolbox

Modern control design and analysis requires a lot of *linear algebra* (matrix multiplication, inversion, calculation of *eigenvalues* and *eigenvectors*, etc.) which is not very easy to perform manually. Try to remember the last time you attempted to invert a 4×4 matrix by hand! It can be a tedious process for any matrix whose size is greater than 3×3. The repetitive linear algebraic operations required in modern control design and analysis are, however, easily implemented on a computer with the use of standard programming techniques. A useful high-level programming language available for such tasks is the MATLAB®, which not only provides the tools for carrying out the matrix operations, but also contains several other features, such as the time-step integration of linear or nonlinear governing differential equations, which are invaluable in modern control analysis and design. For example, in Figure 1.6 the time-history of a double-pendulum has been obtained by solving the coupled governing nonlinear differential equations using MATLAB. Many of the numerical examples contained in this book have been solved using MATLAB. Although not required for doing the exercises at the end of each chapter, it is recommended that you familiarize yourself with this useful language with the help of Appendix A, which contains information about the commonly used MATLAB operators in modern control applications. Many people, who shied away from modern control courses because of their dread of linear algebra, began taking interest in the subject when MATLAB became handy. Nowadays, personal computer versions of MATLAB are commonly applied to practical problems across the board, including control of aerospace vehicles, magnetically levitated trains, and even stock-market applications. You may find MATLAB available at your university's or organization's computer center. While Appendix A contains useful information about MATLAB which will help you in solving most of the modern control problems, it is recommended that you check with the MATLAB user's guide [1] at your computer center for further details that may be required for advanced applications.

SIMULINK® is a very useful Graphical Users Interface (GUI) tool for modeling control systems, and simulating their time response to specified inputs. It lets you work directly with the block-diagrams (rather than mathematical equations) for designing and analyzing

® MATLAB, SIMULINK and Control System Toolbox are registered trademarks of *MathWorks, Inc.*

control systems. For this purpose, numerous linear and nonlinear blocks, input sources, and output devices are available, so that you can easily put together almost any practical control system. Another advantage of using SIMULINK is that it works seamlessly with MATLAB, and can draw upon the vast programming features and function library of MATLAB. A SIMULINK block-diagram can be converted into a MATLAB program (called *M-file*). In other words, a SIMULINK block-diagram does all the programming for you, so that you are free to worry about other practical aspects of a control system's design and implementation. With advanced features (such as the *Real Time Workshop* for C-code generation, and specialized block-sets) one can also use SIMULINK for practical implementation of control systems [2]. We will be using SIMULINK as a design and analysis tool, especially in simulating the response of a control system designed with MATLAB.

For solving many problems in control, you will find the *Control System Toolbox®* [3] for MATLAB very useful. It contains a set of MATLAB M-files of numerical procedures that are commonly used to design and analyze modern control systems. The Control System Toolbox is available at a small extra cost when you purchase MATLAB, and is likely to be installed at your computer center if it has MATLAB. Many solved examples presented in this book require the Control System Toolbox. In the solved examples, effort has been made to ensure that the application of MATLAB is clear and direct. This is done by directly presenting the MATLAB line commands – and some MATLAB *M-files* – followed by the numerical values resulting after executing those commands. Since the commands are presented *exactly* as they would appear in a MATLAB *workspace*, the reader can easily reproduce all the computations presented in the book. Again, take some time to familiarize yourself with MATLAB, SIMULINK and the Control System Toolbox by reading Appendix A.

References

1. *MATLAB® 6.0 – User's Guide*, The Math Works Inc., Natick, MA, USA, 2000.
2. *SIMULINK® 4.0 – User's Guide*, The Math Works Inc., Natick, MA, USA, 2000.
3. *Control System Toolbox 5.0 for Use with MATLAB® – User's Guide*, The Math Works Inc., Natick, MA, USA, 2000.