Chapter 1

Introduction to Dynamic Systems and Control

1.1 INTRODUCTION

In solving engineering problems, there is a need to understand and determine the dynamic response of a physical system that may consist of several components. These efforts involve *modeling*, *analysis*, and *simulation* of physical systems. Typically, building a prototype system and conducting experimental tests are either infeasible or too expensive for a preliminary design. Therefore, mathematical modeling, analysis, and simulation of engineering systems aid the design process immensely.

Dynamic systems and control involves the analysis, design, and control of physical engineering systems that are often composed of interacting mechanical, electrical, and fluid subsystem components. One example is an electrically controlled hydraulic actuator that is used to change the position of an aerodynamic surface (e.g., rudder) on an airplane. This system consists of several interacting components: an electromagnet circuit is used to open a mechanical valve that allows high-pressure hydraulic fluid to flow into a cylinder chamber; the fluid pressure causes a mechanical piston to move; and mechanical linkages connecting the hydraulic piston to the aerodynamic surface (e.g., rudder) cause the rudder to change position. Finally, an onboard digital computer (an "autopilot") uses feedback from sensors to adjust the operation of the hydraulic actuator so that the rudder position (and the subsequent response of the airplane) matches the desired value. This example demonstrates why it is advantageous for the engineer to understand the dynamic response of this interconnected system without relying on experimentation with a physical prototype.

Here are definitions of the important terms that we use throughout the book:

System: A combination of components acting together to perform a specified objective. The components or interacting elements have cause-and-effect (or input-output) relationships. One example of a system is a direct-current (DC) motor where a voltage input causes angular velocity (the output) of the mechanical load attached to the motor's shaft.

Dynamic system: A system where the current output variables (or *dynamic variables*) depend on the initial conditions (or stored energy) of the system and/or the previous input variables. The dynamic variables of the system (e.g., displacement, velocity, voltage, pressure) vary with time. For the DC motor example, the angular velocity of the motor is the dynamic variable and the circuit voltage is the input.

Modeling: The process of applying the appropriate fundamental physical laws in order to derive mathematical equations that adequately describe the physics of the engineering system. Dynamic systems are represented by differential equations. For the DC motor example, the electrical circuit is modeled by using Kirchhoff's voltage law, and the mechanical motion is modeled by using Newton's second law.

Mathematical model: A mathematical description of a dynamic system's behavior, which is usually a set of linear or nonlinear ordinary differential equations (ODEs). For the DC motor example, the mathematical model consists of a differential equation for the electrical current and a differential equation for mechanical motion.

Simulation: The process of obtaining the system's dynamic response by numerically solving the governing modeling equations. Simulation involves numerical integration of the model's differential equations and is performed by digital computers and simulation software.

System analysis: The use of analytical calculations or numerical simulation tools to determine the system response in order to assess its performance. Repeated analysis aids the design process where the system's configuration or parameters are altered to improve performance or meet desired constraints. For the DC motor example, we might apply a constant voltage input and determine the characteristics of the angular velocity response by using analytical calculations ("by hand") or numerical simulations. If the angular velocity response is inadequate, we could alter the system's parameters in order to improve performance.

1.2 CLASSIFICATION OF DYNAMIC SYSTEMS

In general, we can classify dynamic systems according to the following four categories: (1) distributed versus "lumped" systems, (2) continuous-time versus discrete-time systems, (3) time-varying versus time-invariant systems, and (4) linear versus nonlinear systems.

Distributed versus Lumped Systems

A *distributed system* requires an infinite number of "internal" variables, and, therefore, the system is governed by partial differential equations (PDEs). A *lumped system* involves a finite number of "internal" variables, and, therefore, the system is governed by ODEs. For example, if we want to model a hydraulic piston, we would "lump" all pressure distributions in a cylinder chamber into one single pressure term. Therefore, we would have one ODE for the time derivative of pressure (dP/dt or $\dot{P}$) for each "lump" of fluid in a particular chamber. In this textbook, we work exclusively with lumped systems and ODEs.

Continuous-Time versus Discrete-Time Systems

A *continuous-time system* involves variables and functions that are defined for all time, whereas a *discrete-time system* involves variables that are defined only at discrete time points. We may think of continuous-time systems consisting of variables in the "analog" domain, such as position $x(t)$. Discrete-time systems consist of variables in the "digital" domain, such as the sampled (measured) position $x(kT_s)$ that exists only at the discrete-time points $t = T_s, t = 2T_s, \ldots, t = kT_s$ where T_s is the sampling interval. Continuous-time systems are described by differential equations while discrete-time systems are described by difference equations. In this volume, we work with continuous-time systems and differential equations. We introduce discrete-time systems in Chapter 10 when we examine the role of digital computers in automatic control systems and the need to convert analog signals to digital signals and vice versa.

Time-Varying versus Time-Invariant Systems

In a *time-varying system*, the system parameters change with time (e.g., the friction coefficient changes with time). In a *time-invariant system*, the parameters remain constant. The reader should not confuse the variation of the system parameters with the variation of the dynamic variables. For the DC motor example, the system parameters would be electrical resistance of the circuit, inductance of the coil windings around the rotor, friction coefficient for the rotor bearings, and moment of inertia of the rotor. If these system parameters do not change with time (i.e., they are constants for the system model), then the DC motor is a time-invariant system. Of course, the dynamic variables associated with the DC motor (electrical current of the circuit and angular velocity of the output shaft) can change with time. We focus primarily on time-invariant systems in this text.

Linear versus Nonlinear Systems

Suppose we have a system or input−output relationship that is described by the function $y = f(u)$ where u is the input and y is the output. Linear systems obey the *superposition property*:

1. If $y_1 = f(u_1)$, then $ay_1 = f(au_1)$, where $a =$ any constant.
2. If $y_1 = f(u_1)$ and $y_2 = f(u_2)$, then $y_1 + y_2 = f(u_1 + u_2)$.

Consider again the DC motor example: suppose we apply 12 volts (V) to a motor and through measurements determine the steady-state (constant) angular velocity to be 1600 revolutions per minute (rpm). Next, if we apply 6 V to the motor and the measured steady-state angular velocity is 800 rpm then the system obeys the first superposition property and the DC motor system is linear. Of course, a physical system that demonstrates linearity (such as the DC motor) has a limited linear range of operation; that is, we cannot increase the input voltage by a factor of 100 and expect the corresponding angular velocity to increase by a factor of 100. Increasing the system input beyond a threshold may cause the output to saturate (i.e., reach a limit) and, therefore, the system is no longer linear.

The second superposition property shows that the total dynamic response of a linear system can be obtained by adding or superimposing the responses (or solutions) to individual input functions. Nonlinear systems do not obey either superposition property.

The following equations are examples of *linear* ODEs:

$$\ddot{x} + 3\dot{x} - 40x = 6u \tag{1.1}$$

$$2\ddot{x} + 0.4\dot{x} + 0.6e^{-2t}x = -8u \tag{1.2}$$

Equation (1.1) is a second-order linear ODE because the dynamic variable x and its derivatives appear as linear combinations (we will use the over-dot notation to denote derivatives with respect to time; hence, $\dot{x} = dx/dt$, $\ddot{x} = d^2x/dt^2$, etc.). Equation (1.1) involves constant coefficients and hence it is a *linear time-invariant* (LTI) differential equation. Equation (1.2) is linear as x and its derivatives appear in linear combinations. Because the coefficient $0.6e^{-2t}$ changes with time, Eq. (1.2) is a *linear time-varying* ODE. The following equation

$$2\ddot{x} + 3\dot{x} + 16x^2 = 5u \tag{1.3}$$

is a *nonlinear* ODE because of the x^2 term.

All physical systems are nonlinear. However, if we confine the input−output variables to a restricted (nominal) range, then we can often replace a nonlinear system with a *linear* model comprising linear differential equations. This important process is called *linearization*. Obtaining a linear model is extremely important and advantageous in system analysis because it is possible to obtain the analytical (closed-form) solution to linear ODEs. Nonlinear systems must be solved by using numerical methods to integrate the ODEs.

1.3 MODELING DYNAMIC SYSTEMS

A major focus of this book is mathematical modeling of dynamic systems. Developing an appropriate model is always the first step in system analysis because it is impossible to determine the system's response without a mathematical representation of the system dynamics. Mathematical models are obtained by applying the appropriate laws of physics to each element of a system. Some system parameters (such as friction characteristics) may be unknown, and these parameters are often determined through experimentation and observation, which lead to empirical relations. Engineering judgment must be used to trade model complexity with the accuracy of the analysis. Nonlinearities (such as gear backlash) are often ignored in preliminary design studies in order to derive linear models. Sometimes, low-order approximate linear models can be developed to accurately represent the system dynamics. These low-order linear models can be solved analytically ("by hand"), which gives the engineer an intuitive feel for the nature of the dynamic system. Furthermore, simulations are easier to construct with low-order linear models and therefore the time required to perform system analysis is reduced. Nonlinear models, on the other hand, require numerical solutions using simulation software. Extremely complex nonlinear models typically require small integration time steps to accurately solve the ODEs thus increasing computer-run time. Consequently, there is usually a trade-off between model complexity and analysis time.

Engineers must remember that the results obtained for a particular mathematical model are only approximate and are valid only to the extent of the assumptions used to derive the model. The model must be sufficiently sophisticated to demonstrate the significant features of the dynamic response without becoming too cumbersome for the available analysis tools. The validity of a mathematical model can often be verified by comparing the model solution (such as simulation results) with experimental results. The Shuttle Avionics Integration Laboratory (SAIL) was a hardware-in-the-loop test facility at NASA Johnson Space Center [1]. SAIL consisted of actual Space Shuttle hardware (such as

the flight deck, cockpit displays, sensors, and electronic wiring) and mathematical models of the physical forces due to aerodynamics, gravity, and propulsion. Engineers and astronauts used SAIL to perform "real-time" simulations of Space Shuttle missions in order to test and validate the flight software. The simulation results from SAIL tests showed an excellent match with actual Shuttle flight data. SAIL tests, however, could occasionally be intermittent owing to their reliance on using very complex mathematical models (i.e., computer software) to interface with and drive all of the physical Shuttle hardware. The SAIL facility is an example of one extreme end of the mathematical modeling spectrum: a complex, "high-fidelity" simulation that was at times prone to sporadic testing. This trade-off was necessary in order to accurately model the Shuttle's flight dynamics.

Simulation Tools

Several commercial simulation tools have been developed to help engineers design and analyze dynamic systems. We briefly discuss a few of these software tools as examples of mathematical modeling and system analysis.

Simulink is a numerical simulation tool that is part of the MATLAB software package developed by MathWorks [2]. It uses a graphical user interface (GUI) to develop a block diagram representation of dynamic systems. Simulink is used by engineers in industry and academia. Constructing system models with Simulink is relatively easy and, therefore, it is often used to build simple models during the preliminary design stage. However, Simulink can be used to simulate complex, highly nonlinear systems. In this book, we use it extensively to simulate and analyze dynamic systems.

Caterpillar Inc. has developed the simulation tool Dynasty that allows engineers to construct complex models of large off-road vehicles [3]. The engineer can build software models of integrated machines by "dragging-and-dropping" subsystem models from a library. These subsystems include engines, linkages, drive trains, hydraulics, and controls. The underlying physics of each subsystem are contained within the mathematical model of the individual component. The Dynasty software simulates the dynamics of the integrated vehicle model and allows engineers to perform tests, analyze the dynamic response, and vary the subsystem components in order to improve overall system performance. Caterpillar engineers used Dynasty to analyze and design the 797B mining truck (its largest vehicle) and bring it to production in less than half the time it would take by building physical prototypes of the truck.

EASY5, originally developed by Boeing, is a graphics-based simulation tool for constructing virtual prototypes of engineering systems [4]. As in Dynasty, the user can select prebuilt components from libraries that include models of mechanical, electrical, hydraulic, pneumatic, and thermal subsystems. EASY5 can interface with Simulink and other computer-aided engineering software tools. Engineers have used EASY5 to analyze and design aerospace vehicles, for example.

In summary, it should be noted that all numerical simulation tools are constructed using the basic principles of mathematical modeling that are presented in this book. That is, the appropriate physical law (e.g., Newton's second law, Kirchhoff's voltage law) is applied to the particular system (mechanical, electrical, fluid, etc.) in order to develop the differential equations that describe the system dynamics. The differential equations are then solved using numerical integration methods. The solution to the differential equations is the system's dynamic response.

1.4 OBJECTIVES AND TEXTBOOK OUTLINE

The objective of this book is to present a comprehensive yet concise treatment of dynamic systems and control. In particular, on completing this volume, the reader should be able to accomplish the following tasks: (1) develop the mathematical models for mechanical, electrical, fluid, or thermal systems; (2) obtain the system's dynamic response (due to input functions and/or initial energy storage) by using numerical simulation tools and analytical techniques; and (3) analyze and design feedback control systems in order to achieve a desirable system response. This book primarily emphasizes lumped, continuous-time, LTI systems. Hence, all mathematical models involve ODEs and the majority have constant coefficients. Nonlinear systems are given considerable attention, and, therefore, we make frequent use of Simulink to obtain the dynamic response. Furthermore, we often utilize the linearization process in order to approximate the nonlinear dynamics with linear system dynamics. As is demonstrated, obtaining a linear mathematical model allows us to use a wealth of analytical tools for system analysis and graphical techniques for designing feedback control systems.

This book is organized according to its three major objectives. Chapters 2–4 deal with developing the mathematical models of physical engineering systems. In particular, Chapter 2 treats mechanical systems and the derivation

of the modeling equations by applying Newton's laws of motion. Chapter 3 deals with mathematical models for electrical and electromechanical systems. Here we apply the element laws that govern the interaction between electrical charge, current, magnetic flux, and voltage. Mathematical models for fluid and thermal systems are developed in Chapter 4 by utilizing the conservation of mass and the conservation of energy, respectively. In Chapters 2–4, we focus on developing mathematical models of "real-world" physical engineering systems such as vehicle suspension systems, energy-transmission devices, and systems with mixed disciplines such as electromechanical, hydromechanical, and pneumatic actuators. Chapter 5 deals with the standard formats for representing the various mathematical models derived in the previous three chapters. These standard formats facilitate the second major topic of the book—obtaining the system response— whether we use numerical or analytical techniques.

Chapter 6 begins the system-analysis section of the book. This chapter introduces Simulink as the numerical simulation tool of choice for obtaining the response of linear and nonlinear dynamic systems. Chapter 7 presents analytical techniques for solving the mathematical modeling equations "by hand." Here we analyze the total system response (comprising the transient and steady-state responses) to input functions. In Chapter 8, we introduce the Laplace transformation method for obtaining the response of dynamic systems that are modeled by LTI differential equations. Chapter 9 deals with obtaining the response of a dynamic system that is driven by an oscillating or harmonic input function. This frequency-response analysis is aided by graphical techniques such as the Bode diagram.

Chapter 10 introduces the reader to the third major objective of the textbook: the analysis and design of feedback control systems. Here we investigate the use of feedback (from measurement sensors) to shape the system input function in order to achieve a desirable output response. Although different control schemes are discussed, this chapter emphasizes the proportional-integral-derivative (PID) controller (and its variants) because it is the most widely used control scheme in industry. Control-system design is aided by two graphical techniques, the root-locus method and the Bode diagram, that are discussed in this chapter.

Chapter 11 presents five engineering case studies that demonstrate the three major objectives of this textbook: modeling, analysis, and control of dynamic systems. These examples are inspired by research from the engineering literature and involve physical systems such as vehicle suspensions and actuators. The final chapter serves as a "capstone" for this book.

Appendix A presents units and Appendix B gives a brief overview of MATLAB usage, its commands, and programming with MATLAB. Only the MATLAB commands that pertain to solving problems in dynamic systems and control are presented in Appendix B. Appendix C is a tutorial on using Simulink to simulate linear and nonlinear dynamic systems.

REFERENCES

1. Melone, K., "SAILing Through Space," *Boeing Frontiers*, Vol. 9, September 2010, pp. 24–25.
2. http://www.mathworks.com/products/simulink/ (accessed 10 March 2014).
3. Dvorak, P., "Software Simulates Many Disciplines in One Model," *MachineDesign.com*, http://machinedesign.com/article/software-simulates-many-disciplines-in-one-model-1106 (accessed 10 March 2014).
4. http://www.mscsoftware.com/Products/CAE-Tools/Easy5.aspx (accessed 10 March 2014).