HZ BOOKS

经典原版书库

机械电子系统设计

（英文版）

MECHATRONICS SYSTEM DESIGN

DEVDAS SHETTY • RICHARD A. KOLK

（美）Devdas Shetty 哈特福德大学
Richard A. Kolk 美国联合技术公司 著

机械工业出版社
China Machine Press

经典原版书库

机械电子系统设计

（英文版）

Mechatronics System Design

Devdas Shetty
哈特福德大学
（美） Richard A. Kolk 著
美国联合技术公司

机械工业出版社
China Machine Press

Devdas Shetty and Richard A. Kolk
Mechatronics System Design
ISBN: 0-534-95285-2

981-254-466-6

本书版权登记号：图字：01-2004-1647

图书在版编目（CIP）数据

机械电子系统设计（英文版）/（美）谢提（Shetty, D.）等著. –北京：机械工业出版社，2004.5
（经典原版书库）
书名原文：Mechatronics System Design
ISBN 7-111-14119-9

Ⅰ. 机… Ⅱ. 谢… Ⅲ. 机电系统－设计－英文 Ⅳ. TH-39

中国版本图书馆CIP数据核字（2004）第017991号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：迟振春
北京中加印刷有限公司印刷 · 新华书店北京发行所发行
2004 年 5 月第 1 版第 1 次印刷
787mm × 1092mm 1/16 · 27印张
印数：0 001-3 000册
定价：39.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

Preface

Mechatronics is the synergistic combination of mechanical and electrical engineering, computer science, and information technology, which includes control systems as well as numerical methods used to design products with built-in intelligence.

We intend this text to be useful for the upper-level undergraduate or graduate student in mechanical, industrial, manufacturing, civil, electrical, and of course, mechatronic engineering. The text blends the pertinent aspects of mechatronics—system modeling, simulation, sensors, actuation, real-time computer interfacing, and control—into a single unified result suitable for use in the college-level mechatronic curriculum. Students are introduced to all the topics needed to develop a good understanding of the basic principles used in mechatronics technology.

A unique feature of this text is its coverage of modeling and simulation of physical systems. Historically this topic has required students to be proficient in at least one computer language such as FORTRAN, C, or BASIC. Recognizing that this is seldom true, we use block diagram-based visual programming environments in this text for all modeling and simulation tasks. Students will find this environment intuitive, flexible, and much easier to understand.

Key Features

This textbook is written for the student and practicing engineer. You will find it useful as a reference as well as a resource for explanation. The key features of this text include

- Overview and explanation of mechatronics from a model-based perspective.
- Modified Analogy Approach for creating dynamical models of physical systems.
- Sensor and actuator *cookbook* sections describing the operation and selection of many devices.
- Thorough discussion of classical control incorporating *real-world* constraints.
- A discussion of analog and digital hardware components for real-time computer interfacing.
- A collection of case studies complete with parts list suitable for laboratory exercises.
- A summary of recent advances in the mechatronics field and future trends.

Due to the extensive use of microelectronics, new materials, and control, many of the processing circuits, which were historically part of the external configuration, are now routinely built, or integrated, into the product. For example, smart sensors with integrated microelectronic circuits for linearization and signal conditioning make sensors modular and easy to use in a wide variety of applications. Microsensors and microelectromechanical systems have countless applications in consumer products, health care, process control, military/aerospace and environmental engineering. Their small size, exceptional performance, and broad applicability offer engineers new potential for enhanced system designs.

Organization of the Book

A mechatronics course based on this textbook will introduce the student to the mechatronic design process through examples, problems, and case studies which can be quickly and affordably assembled and investigated in laboratory settings.

- Following an introduction to the fundamentals of mechatronics, *Chapter 1* provides an in-depth discussion of the mechatronic design process.
- *Chapter 2* is devoted entirely to system modeling and simulation. Students will learn to create accurate computer-based dynamic models from illustrations and other information using the Modified Analogy Approach. This unique method combines the standard analogy approach to modeling, found in many texts, with block diagrams, the major difference being the ability to incorporate nonlinearities directly without linearization. Chapter 2 addresses a variety of physical systems often found in mechatronics. Such systems include mechanical, electrical, thermal, fluid, and hydraulic components. Models and techniques developed in this chapter are used in subsequent chapters in the chronology of the mechatronic design process.
- *Chapter 3* presents the basic theoretical concepts and operating principles of sensors and transducers. Other topics discussed include instrumentation principles, analog and digital sensing, and sensors for motion, force, and vibration.
- *Chapter 4* discusses actuating devices, including DC motors, stepper motors, fluid power devices, and piezoelectric actuators. A complete stepper motor/drive model suitable for use in mechatronic analysis is presented in detail.
- *Chapter 5* discusses the hardware and software requirements of mechatronics. This includes Boolean logic, analog and digital electronics, and programmable logic controllers.
- *Chapter 6* presents controls and their design for use in mechatronic systems. Special attention is paid to real-world constraints, including time delays and nonlinearities. The Root Locus and Bode Plot design methods are discussed in detail, along with several design procedures for common control structures, including PI, PD, lag, lead, rate feedback, and pure gain.
- *Chapter 7* discusses the theoretical and practical aspects of computer interfacing and real-time data acquisition and control. Signal processing and data interpretation are also handled using the visual programming approach.
- *Chapter 8* presents advanced concepts, future trends, and the state-of-the-art developments in mechatronics.
- *Chapter 9* presents a collection of case studies suitable for laboratory investigations. Component integration and design considerations are addressed for applications including robotics, manipulators, machine tools, aircraft, thermal systems, and fluid process plants. All case studies are implemented using a general purpose I/O board, low-cost equipment, a visual simulation environment, and application software. the authors will provide additional information for those interested in duplicating the case studies.

The real challenge in writing this book has been to connect complex and seemingly independent topics in a clear and concise manner, which is absolutely essential for an understanding of mechatronics. Readers of this text will be equipped with all the tools necessary to plan, test, and implement a well-designed mechatronic system.

Acknowledgments

Material presented in this book is a collection of many years of research and teaching by the authors at the University of Hartford, Cooper Union, United Technologies Research Center, McDonnell Douglas Aircraft, and The Carrier Corporation.

We are indebted to a number of colleagues and friends for their help in the preparation of this work. For reviewing the manuscript we would like to acknowledge Professors Chester Dudzik, Bob Loftas, and Paul Botosoni. For useful suggestions and encouragement, we thank Professors Ernest Gardow, Donald Leone, Louis Godbout, Frank Lahey, and Dr. John Cagnetta, all of the University of Hartford.

Special thanks to Dr. Fred Cogswell, Mr. Tom Hardy, Mr. Carl Sgamboti, and Dr. Tim Obee, all of the United Technologies Research Center, for numerous discussions on system modeling. We are grateful to Mr. Steve Defoe and Mr. Paul James, both of Carrier Electronics, for their reviews and suggestions on the controls chapter. And we thank Visual Solutions, Inc. for assistance in the real-time interfacing chapter.

Contents

Chapter 4 Actuating Devices 211

Chapter 5 Hardware Components for Mechatronics 239

Chapter 6 Signals, Systems, and Controls 277

Chapter 7 Real-Time Interfacing 325

Chapter 1

Mechatronics System Design

- 1.1 What Is Mechatronics?
- 1.2 Integrated Design Issues in Mechatronics
- 1.3 Mechatronics Key Elements
  - Information Systems
  - Mechanical Systems
  - Electrical Systems
  - Computer Systems
  - Sensors and Actuators
  - Real-Time Interfacing
- 1.4 The Mechatronics Design Process
- 1.5 Advanced Approaches in Mechatronics
- 1.6 Summary
- References
- Problems

This chapter provides the student with an overview of the mechatronic design process and a general description of the technologies employed in the mechatronic approach. This chapter begins by introducing the key elements, techniques, and the design process used for mechatronics system design. Following a definition of mechatronics and a discussion of several important design issues; the six key elements of mechatronics—information systems, electrical systems, mechanical systems, computer systems, sensors, actuators, and real-time interfacing—are introduced. Characteristics pertinent to mechatronics are developed from first principles. Although experience in any of the supporting technologies is helpful, it is not necessary. The chapter closes with a description of the mechatronics design process and a discussion of some interesting current applications.

1.1 What Is Mechatronics?

► ***Mechatronics is a methodology used for the optimal design of electromechanical products.***

A *methodology* is a collection of practices, procedures, and rules used by those who work in a particular branch of knowledge, or *discipline.* The familiar technological disciplines include thermodynamics, electrical engineering, computer science, and mechanical engineering, to name several. The mechatronic system is *multi-disciplinary,* embodying four fundamental disciplines: *electrical, mechanical, computer science, and information technology.*

The F-22 fighter, under development at Lockheed-Martin and Boeing, is an example of mechatronic technology in action. The design metric emphasizes reliability, maintainability, performance, and cost. Multi-disciplinary functionality, including the integrated flight-propulsion controls and two-dimensional, thrust-vectoring engine nozzles, is being designed into the aircraft starting at the preliminary design stage.

Multi-disciplinary systems are not new; they have been successfully designed and used for many years. One of the most common is the *electromechanical* system, which often uses a computer algorithm to modify the behavior of a mechanical system. Electronics are used to transduce information between the computer science and mechanical disciplines.

The difference between a mechatronic system and a multi-disciplinary system is not the constituents, but rather *the order in which they are designed*. Historically, multi-disciplinary system design has employed a sequential *design-by-discipline* approach. For example, the design of an electromechanical system is often accomplished in three steps beginning with the mechanical design. When the mechanical design is complete, the power and microelectronics are designed followed by the control algorithm design and its implementation. The major drawback of the design-by-discipline approach is that fixing the design at various points in the sequence causes new constraints, resulting from the design to that point; to be created and passed on to the next discipline. Many control system engineers are familiar with the following quip:

- ► *"Design and build the mechanical system. Then bring in the painters to paint it and the control system engineers to install the controls."*

Control designs often fail because of these additional constraints. For example, cost reduction is a major factor in most systems. Tradeoffs made during the mechanical and electrical design stages often involve sensors and actuators. Lowering the sensor-actuator count and using less accurate sensors or less powerful actuators are some of the standard methods for achieving cost savings. Their effects on the control system are additional, and sometimes conflicting, constraints.

- ► ***The mechatronic design methodology is based on a concurrent, instead of sequential, approach to discipline design, resulting in products with more synergy.***

The branch of engineering called *systems engineering* uses a concurrent approach for *preliminary design*. In a way, mechatronics is an extension of the system engineering approach, but it is *supplemented* with information systems to *guide* the design and is applied at *all* stages of design, not just the preliminary design step, making it more *comprehensive*. There is a synergy in the integration of mechanical, electrical, and computer systems with information systems for the design and manufacture of products and processes. The synergy is generated by the right combination of parameters; that is, the final product can be better than just the sum of its parts. Mechatronic products exhibit performance characteristics that were previously difficult to achieve without the synergistic combination. The key elements of the mechatronics approach are presented in Figure 1-1.

Even though the literature often adopts this concise representation, a clearer but more complex representation is shown in Figure 1-2.

Mechatronics is the result of applying information systems to physical systems. The physical system, the rightmost dotted block, consists of mechanical, electrical, and computer systems as well as actuators, sensors, and real-time interfacing. In some of the literature, this block is called an electromechanical system.

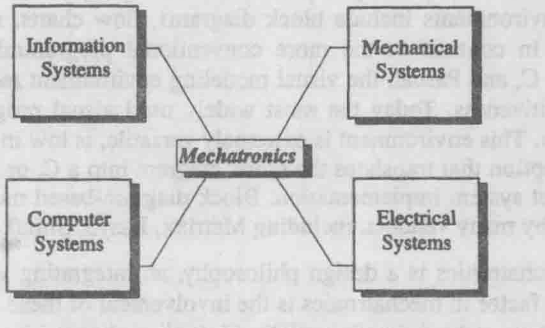

Figure 1-1 Mechatronics Constituents

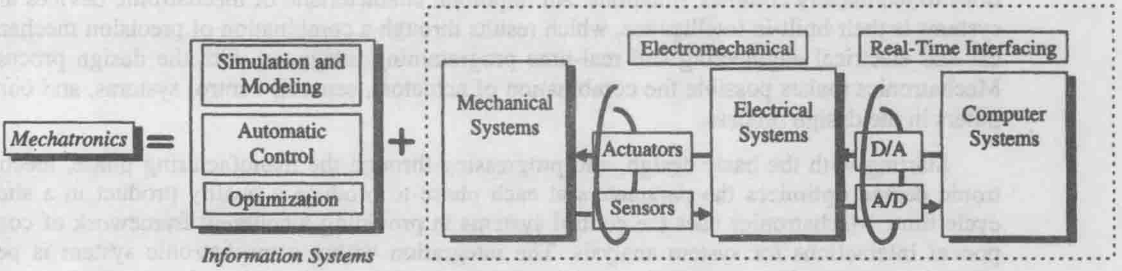

Figure 1-2 Mechatronics Key Elements

▶ ***A mechatronic system is not an electromechanical system and is more than a control system.***

Sensors and actuators are used to transduce energy from high power, usually the mechanical side, to low power, the electrical and computer side. The block labeled mechanical systems frequently consists of more than just mechanical components and may include fluid, pneumatic, thermal, acoustic, chemical, and other disciplines as well. New developments in sensing technologies have emerged in response to the ever-increasing demand for solutions of specific monitoring applications. They have produced faster, more sensitive; and more precise measuring devices. Sensors are being miniaturized and implemented in solid state form so that several sensors can be integrated and their functions combined.

Control is a general term and can occur in living beings as well as machines. The term *automatic control* describes the situation in which a machine is controlled by another machine.

Irrespective of the application, such as industrial control, manufacturing, testing, or military, new developments in sensing technology are constantly emerging.

1.2 Integrated Design Issues in Mechatronics

The inherent concurrency, or simultaneous engineering, of the mechatronics approach relies heavily on the use of system modeling and simulation throughout the design and prototyping stages. Because the model will be used and altered by engineers from multiple disciplines, it is especially important that it be programmed in a visually intuitive environment. Such environments include block diagrams, flow charts, state transition diagrams, and Bond graphs. In contrast to the more conventional programming languages such as FORTRAN, BASIC, C, and Pascal, the visual modeling environment requires little training due to its inherent intuitiveness. Today the most widely used visual programming environment is the *block diagram*. This environment is extremely versatile, is low in cost, and often includes a *code generator* option that translates the block diagram into a C, or similar, high-level language suitable for target system implementation. Block diagram-based modeling and simulation packages are offered by many vendors, including Matrixx, Easy5, SimuLink, VisSim, and LabView.

Mechatronics is a design philosophy, an integrating approach to engineering design. The primary factor in mechatronics is the involvement of these areas throughout the design process. Through a mechanism of simulating interdisciplinary ideas and techniques, mechatronics provides ideal conditions to raise the synergy, thereby providing a catalytic effect for the new solutions to technically complex situations. An important characteristic of mechatronic devices and systems is their built-in intelligence, which results through a combination of precision mechanical and electrical engineering and real-time programming integrated with the design process. Mechatronics makes possible the combination of actuators, sensors, control systems, and computers in the design process.

Starting with the basic design, and progressing through the manufacturing phase, mechatronic design optimizes the parameters at each phase to produce a quality product in a short cycle time. Mechatronics uses the control systems in providing a coherent framework of component interactions for system analysis. The integration within a mechatronic system is performed through the combination of hardware (components) and software (information processing). Hardware integration results from designing the mechatronic system as an overall system and bringing together the sensors, actuators, and microcomputers into the mechanical system. Software integration is primarily based on advanced control functions. Figure 1-3 illustrates how the hardware and software integration take place. It also shows how process knowledge and information processing are involved in addition to the feedback process.

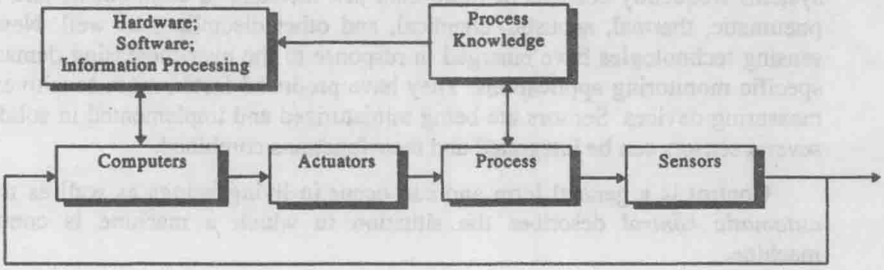

Figure 1-3 General Scheme of Hardware and Software Integration

The first step in the focused development of mechatronic systems is to analyze the customer needs and the technical environment in which the system is integrated. Complex technical systems designed to solve problems tend to be a combination of mechanical, electric, fluid power, and thermodynamic parts with hardware in digital and analog form coordinated by complex software. Typical mechatronic systems gather data and information from their technical environment using sensors. The next step is to use elaborate ways of modeling and description methods to cover all subtasks of this system in an integrated manner. This includes an effective description of the necessary interfaces between subsystems at an early stage. The data are processed and interpreted, leading to actions carried out by actuators. Mechatronic systems result in shorter developmental cycles, lower costs, and higher quality. They also provide additional influence through the acquisition of information from the process.

Mechatronic design supports the concepts of concurrent engineering. In the designing of a mechatronic product, it is necessary that the knowledge and required information be coordinated among different expert groups. Concurrent engineering is a design approach in which the design of a product and manufacture of a product are merged in a special way. Traditional barriers between design and manufacturing are removed. It has been influenced partly by the recognition that many of the high costs in manufacturing are decided at the product design stage itself. Even during the design stage, it is involved with customer perception, market analysis, optimized performance, life cycle performance, quality, reliability, and sales. Product design and process planning take place concurrently. The total philosophy of concurrent engineering in the organization is well suited for team-oriented project management, with emphasis on collective decision making. Successful implementation of concurrent engineering is possible by coordinating adequate exchange of information and dealing with organizational barriers to cross-functional cooperation. Due to the influence of concurrent engineering, traditional barriers between design and manufacturing have decreased; however, the lack of a common interface language has made the information exchange in concurrent engineering difficult.

A mechatronic product can achieve impressive results if it is effectively integrated with the concurrent engineering management strategy. The benefits that accrue are greater productivity, higher quality, and production reliability by the incorporation of intelligent, self-correcting sensory and feedback systems. The integration of sensors and control systems in a complex system reduces capital expenses, maintains a high degree of flexibility, and results in a higher percentage of machine utilization. In order to implement a mechatronic concurrent engineering system that can achieve these objectives, the organization must start with a long-range plan that can take into account tomorrow's changing needs in processes, data functions, control, and integration tools.

1.3 Mechatronics Key Elements

Information Systems

Information systems include all aspects of information transmission, from signal processing to control systems to analysis techniques. An information system is a combination of four disciplines: communication systems, signal processing, control systems, and numerical methods. In mechatronics applications we are most concerned with modeling, simulation, automatic control, and numerical methods for optimization.

Modeling and Simulation

Modeling is the process of representing the behavior of a real system by a collection of mathematical equations and logic. The term *real system* is synonymous with *physical system*, that is, a system whose behavior is based on matter and energy. Models can be broadly categorized as either *static* or *dynamic*. In a static model there is no energy transfer. Systems that are static produce no motion, heat transfer, fluid flow, traveling waves, or anything that changes. On the other hand, a dynamic model has energy transfer that results in power flow. Power, or rate of change of energy, causes motion, heat transfer, and other phenomena that change in time. Phenomena are observed as *signals,* and because time is often the independent variable, most signals are indexed with respect to time.

Models are cause-effect structures; they accept external information and process it with their logic and equations to produce one or more outputs. Exogenous, or externally produced, information supplied to the model can be either fixed in value or changing. An external fixed-value unit of information is called a *parameter,* and an external changing unit of information is called an *input signal*. Traditionally, all model output information is assumed to be changing and is therefore referred to as *output signals*.

Because models are collections of mathematical and logic expressions, they are straightforward to represent in text-based programming languages such as C, BASIC, or FORTRAN. Unfortunately, once the model is in the programming language, one must be familiar with the specific language to understand the model. Because most engineers are not familiar with most programming languages, text-based modeling proved to be a poor candidate for mechatronics. The ideal candidate would be picture-based, or *visual*, instead of text-based and intuitive. *Visual modeling environments* are not without problems. One of the most serious is ambiguity. The same qualities that make a picture "worth a thousand words" can make pictures more difficult to interpret. Ambiguity can be reduced by making the visual modeling environment more intuitive.

As early as 1978, *visual programming languages* suitable for physical system modeling began appearing. Languages such as K3LA, Digicon, Easy5, and Protoblock were developed primarily for use in the aeronautical industry and were not readily available elsewhere. In the early 1980's Integrated Systems introduced their block diagram-based Matrixx programming environment. Initially the software needed the power of a minicomputer to operate; however, as PC's became more powerful, the block diagram technology migrated to the microcomputer. Today there are many good block diagram-based modeling applications available for the PC, including Matrixx, Easy5, SimuLink, VisSim, and LabView.

Simulation is the process of *solving* the model and is performed on a computer. Although simulations can be performed on analog computers, it is far more common to perform them on digital computers. The process of simulation can be divided into three sections: initialization, iteration, and termination. If the starting point is a block diagram-based model description, then, in the initialization section, the equations for each of the blocks must be sorted according to the pattern in which the blocks have been connected. For example, a model consisting of three blocks, A, B, and C, connected in series (input to A is exogenous, output of A to input of B, output of B to input of C) would have its equations sorted with the Block A equations first, followed by those in Block B and then by those in Block C.

The iteration section solves any differential equations present in the model using numerical integration and/or differentiation.

An ordinary differential equation is, in general, a nonlinear equation that contains one or more derivative terms as a function of a single independent variable. For most simulations, this independent variable is time. The order (or degree) of an ordinary differential equation equals the highest derivative term present. Most methods employed for the numerical solution of ordinary differential equations are based on the use of *approximating polynomials* that fit a truncated Taylor Series expansion of the ordinary differential equation. Three steps are required:

1. Write a Taylor Series expansion of the functional form of the ordinary differential equation solution about its initial condition(s). Because the independent variable considered is time, all derivative terms in the series will be taken with respect to time.

2. Truncate the Taylor Series at one of the derivative terms, and the resulting truncated series becomes the approximating polynomial.
3. Compute all constant terms and each derivative term based on the initial-condition values to complete the approximating polynomial.

The display section of a simulation is used to present and post process the output. Output may be saved to a file, displayed as a digital reading, or graphically displayed as a chart, strip chart, meter readout, or even animation. Except for some flow charting environments, such as Visio, Rflow, and ABC Flowcharter, all visual modeling environments include the simulation function. Some of the most commonly used environments are Matrixx/System Build (Integrated Systems), Easy5 (Boeing), Matlab/Simulink (Mathworks), LabView (National Instruments), and VisSim (Visual Solutions).

Automatic Controls

Control system engineering is a discipline introduced in the late nineteenth century after Maxwell's discovery that stability of low-order systems (third and under) depends on the roots of their *characteristic equation* and *Routh's Array* method, which provided the engineer with an analytical "tool" for assessing the stability of systems with higher-order characteristic equations. Until the mid-twentieth century, most control applications were entirely motion based (mechanical), motion-temperature based (thermo-mechanical), or motion-flow based (fluid-mechanical). It was not until the invention of the electronic feedback amplifier by H. S. Black at Bell Labs in 1927 that it became possible to combine motion with electronics, producing electromechanical systems.

During the period between 1927 and 1975, the use of electronics to modify the behavior of mechanical systems grew rapidly, especially after the introduction of the transistor and microprocessor. There was no formal procedure for designing these electromechanical systems, but one based on experience was established.

➤ *Design and build the mechanical system, then paint it and install the controls.*

There is more truth in this than you may believe. The fact is that in most applications of controls to systems, the "other system" is designed and usually built first, and the resulting "constraints" are passed, along with the "desired behavior" constraints, to the control system discipline.

➤ *Mechatronics appears to be nothing more than control system engineering. What's the difference?*

The difference is the sequence of design steps. In electromechanical control applications the mechanical system is designed and built first, followed by the electrical (control) system. This approach places additional, and unnecessary, constraints on the control system design, specifically those constraints that result from the design of the mechanical system. In a mechatronic system a concurrent approach is used instead of a series of design steps.

To appreciate controls fully, one must become familiar with some of the basic terminology and concepts used. Because control involves modification of a system's behavior, it is critical to have a concise way to represent systems and their behavior.

An important notational aid used in dynamic systems is called *operator notation.* This *shortcut* notation simplifies how differential and integral equations are represented. The D operator, the operator we will use throughout this text, is defined as follows; $\frac{1}{D}(\cdot) \equiv \int (\cdot) d\tau$ and $D(\cdot) = \frac{d}{dt}(\cdot)$. Several examples using the D operator are presented below.

$$\text{Example 1: } \dot{x}(t) = -4x(t) + r(t) \quad \Rightarrow \quad Dx(t) = -4x(t) + r(t)$$
$$\text{Example 2: } \ddot{x}(t) + \dot{x}(t) - x(t) = r(t) \quad \Rightarrow \quad D^2 x(t) + Dx(t) - x(t) = r(t)$$
$$\text{Example 3: } \dot{x}(t) = 2x(t) + \int r(\tau) d\tau \quad \Rightarrow \quad Dx(t) = 2x(t) + \frac{1}{D} r(t)$$

The cause-effect relationship for many systems can be approximated by a collection of linear ordinary differential equations. To simplify the discussion, consider a system with one input and one output. This configuration is called a single input-single output, or *SISO,* system. Further assume that the input is represented by the variable $r(t)$, the output by the variable $y(t)$, and the system equation by the differential equation

$$\ddot{y}(t) - 2\dot{y}(t) + 7y(t) = \dot{r}(t) - 6r(t)$$

The transfer function provides an equivalent but more meaningful representation of a system than a differential equation. The transfer function is ***the ratio of the output variable to the input variable represented as the ratio of two polynomials in the D operator***. The differential equation presented above can be converted to a transfer function in three steps.

1. Rewrite the equation using operator notation.

$$D^2 y(t) - 2Dy(t) + 7y(t) = Dr(t) - 6r(t)$$

2. Collect and factor all output terms on the left side and input terms on the right side.

$$y(t) \cdot (D^2 - 2D + 7) = r(t) \cdot (D - 6)$$

3. Obtain the transfer function by solving for the ratio of the output to the input signal.

$$\frac{y(t)}{r(t)} = \frac{(D-6)}{(D^2 - 2D + 7)} = \text{Transfer Function}$$

Transfer functions may be represented using any operator, including *s, w,* or *z,* and regardless of the operator, the procedure for computing the transfer function remains unchanged.

A *monic* polynomial has its highest D-power coefficient set to 1. To minimize the number of coefficients in a transfer function, the numerator and denominator polynomials are typically written in monic form with any resulting gain factored out in front of both polynomials. The following example illustrates the monic form factorization.

$$\frac{16D - 4}{5D^2 + 3D + 1} \xrightarrow{\textit{Monic Form}} \frac{16}{5} \cdot \left(\frac{D - \frac{4}{16}}{D^2 + \frac{3}{5}D + \frac{1}{5}} \right)$$